Mobile Web Services for Collaborative Learning

Master Thesis

Iliyana Ivanova
Matr. No. 262369

September 14, 2007

Supervisors: Prof. Dr. Matthias Jarke
Prof. Dr. Wolfgang Prinz

Advisors: Dipl.-Inform. Mohamed Amine Chatti
M. Sc. Satish Narayana Srirama
Declaration

I herewith declare with my signature, that I have written this master thesis on my own, that all reference or assistance received during the writing of the thesis is stated completely, and that any citation is referenced to its source truly.

Aachen, September 14, 2007
Iliyana Ivanova
Abstract

Throughout the years there has been a clear evolution from traditional education through distance and electronic learning to mobile learning. At the same time the view of learning as an isolated individual activity changes to the view of learning as a collaborative process, in which learners at various performance levels work together towards achieving a common goal.

Searching for particularities within the continuously increasing information overflow can take a great deal of efforts and time. Moreover, the most valuable and innovative information is hard to find, as it lies within communities of practice. Locating the right person, who can provide us with exactly the information that we need and who can help us solve exactly the problems that we come upon, is the best way to learn forward.

This master thesis presents a novel approach to expert finding within a truly collaborative mobile learning environment, targeted not only within the framework of the user’s social network, but also within the social networks of his acquaintances, and the social networks of the acquaintances of his acquaintances, and so forth. Such an expert finder flow usually leads to the discovery of more than one potential expert, and the user’s subjective decision who of them is the most knowledgeable one can be based either on the rating for the expert’s level of expertise in the field, or on the path that the expert finder request has travelled before reaching the respective expert. After having found an expert, the user is provided with all the necessary information in order to contact him for further assistance and to ask him particular questions regarding specific issues.

Alongside the valuable knowledge that flows within the system from the experts to the non-experienced users, the system supports the retrieval of a variety of literature resources, such as articles, inproceedings, proceedings, books, URLs, master and PhD theses, and unpublished resources, which have been tagged by the users. As tagging is something subjective, a three-level scale of relevance of a tag to a resource has been introduced.

Except retrieving specific resources at the time when they are needed, the system maintains image and audio resources within photocasting and podcasting channels, through which they are automatically distributed to all subscribers, as soon as they become available. This is one of the most suitable ways for sharing up-to-date knowledge, as the amount of information that can be conveyed through such broadcasting channels is enormous.

Regarding the technical aspects of the thesis, the developed MobileHost CoLearn system is based on the concept of web service provisioning from mobile phones. It takes full advantage of the latest developments in the telecommunication domain in terms of transmission rates and resource capabilities. The seamless interoperable way of expert and knowledge discovery has been achieved through truly interoperable web services and other widely accepted standards. The MobileHost CoLearn system is the first of its kind that adapts mobile web services for collaborative learning, bringing the benefits of the latest technological developments to the average learner.
## Contents

### 1 Introduction

1.1 Motivation .................................................. 18  
1.2 Thesis Objectives ........................................... 20  
1.3 Thesis Outline .............................................. 21

### 2 State of the Art

2.1 Collaborative Learning ...................................... 24  
2.2 E-learning .................................................... 25  
   2.2.1 Overview ............................................... 25  
   2.2.2 Computer Supported Collaborative Learning .......... 25  
2.3 Web 2.0 and Social Software ................................ 26  
   2.3.1 Web 2.0 ................................................ 26  
   2.3.2 Social Software ........................................ 27  
   2.3.3 Potential Use of Social Software for Community Building and Collaborative Learning ..... 28  
      2.3.3.1 Social Software for Building Virtual Communities ........ 28  
      2.3.3.2 Social Software for Supporting Collaborative Learning ... 29  
2.4 M-Learning .................................................. 31  
   2.4.1 Overview ............................................... 32  
      2.4.1.1 Mobile Devices .................................... 32  
      2.4.1.2 Communication Technologies ......................... 33  
   2.4.2 Critical Success Factors for the Development of Mobile Applications ............ 35  
      2.4.2.1 Learner Needs and Requirements ..................... 35  
      2.4.2.2 Mobile Device Constraints ........................... 36  
      2.4.2.3 Network Constraints ................................ 36  
      2.4.2.4 Quality Factors .................................... 37  
2.5 Web Services ................................................. 39  
   2.5.1 Overview ............................................... 40  
      2.5.1.1 Web Services Architecture ......................... 40
2.5.1.2 Web Services Roles ........................................ 40
2.5.1.3 Web Services Operations ................................. 41
2.5.2 Web Services Protocols and Standards ..................... 41
  2.5.2.1 Simple Object Access Protocol (SOAP) ................. 41
  2.5.2.2 Web Services Description Language (WSDL) ........... 42
  2.5.2.3 Universal Description, Discovery and Integration (UDDI) .. 43
2.5.3 Web Services Based Projects for M-Learning ................. 44
2.6 Mobile Web Services Provisioning ............................. 45
  2.6.1 Overview ................................................. 45
  2.6.2 Mobile Web Services Provisioning Based Projects .......... 45
    2.6.2.1 Mobile Host ........................................ 45
    2.6.2.2 UDP Based Communication ............................ 47
  2.6.3 Mobile Web Services Provisioning Based Projects for M-Learning .. 47
2.7 Java 2 Platform Micro Edition .................................. 47
2.8 Summary ........................................................ 49

3 MobileHost CoLearn System Design 50
  3.1 Scenarios ...................................................... 50
  3.2 System Requirements .......................................... 52
    3.2.1 Functional Requirements .................................. 52
    3.2.2 Non-Functional Requirements .............................. 53
  3.3 System Architecture .......................................... 55
  3.4 Modules Hierarchy .......................................... 56
  3.5 Expertise Management Module Design ......................... 58
    3.5.1 Expert Management Module Design ....................... 58
      3.5.1.1 Overview ....................................... 58
      3.5.1.2 Use Case Diagram ................................ 59
    3.5.2 Resource Management Module Design ...................... 60
      3.5.2.1 Overview ....................................... 60
      3.5.2.2 Use Case Diagram ................................ 60
    3.5.3 Broadcast (Photocast and Podcast) Management Module Design .. 61
      3.5.3.1 Overview ....................................... 62
      3.5.3.2 Use Case Diagram ................................ 62
3.6 Expertise Finder Module Design ............................................. 64
3.6.1 Expert Finder Module Design ........................................... 64
3.6.1.1 Use Case Diagram ................................................... 66
3.6.1.2 Expert Search Web Service ........................................ 66
3.6.1.3 Expert Rating Web Service ......................................... 68
3.6.2 Resource Finder Module Design ........................................ 70
3.6.2.1 Use Case Diagram ................................................... 70
3.6.2.2 Resource Finder Web Service ...................................... 71
3.6.3 Expert Answer Module Design .......................................... 73
3.6.3.1 Use Case Diagram ................................................... 73
3.6.3.2 Problem Web Service ............................................... 74
3.6.3.3 Expert Answer Web Service ....................................... 74
3.7 Expertise Broadcasting (Photocasting and Podcasting) Module Design . 75
3.7.1 Use Case Diagram ......................................................... 76
3.7.2 Available Photocasting/Podcasting Channels Web Services ............ 77
3.7.3 Subscribe to Photocasting/Podcasting Channel Web Services ....... 77
3.7.4 Receive Photocast/Podcast Web Services ................................ 78
3.7.5 Comment Photocast/Podcast Web Services ........................... 79
3.7.6 Retrieve Photocast/Podcast Comments Web Services ............... 81
3.7.7 Unsubscribe from Photocasting/Podcasting Channel Web Services .. 82
3.8 Summary ................................................................. 83

4 MobileHost CoLearn System Implementation 84
4.1 Technologies and Tools Used ............................................. 84
4.1.1 J2ME ................................................................. 84
4.1.1.1 CLDC 1.1 / MIDP 2.0 ............................................. 84
4.1.1.2 Required Optional Packages ..................................... 85
4.1.2 Eclipse SDK and EclipseME Plug-In ................................ 86
4.1.3 NetBeans IDE and NetBeans Mobility Pack ........................ 87
4.1.4 Sun Java Wireless Toolkit for CLDC and Sony Ericsson SDK .... 87
4.1.5 kSOAP2 and kXML2 .................................................. 88
4.2 Overall MobileHost CoLearn Implementation ........................ 88
4.2.1 Overall Class Diagram ................................................ 88
4.2.2 Overall Screen Flow .............................................. 89
4.3 Expert Management Module Implementation .................. 91
  4.3.1 Expert Management Class Diagram .......................... 91
  4.3.2 Expert Management Screen Flow ............................. 92
  4.3.3 Storage of Experts' Data ..................................... 94
4.4 Resource Management Module Implementation ................ 95
  4.4.1 Resource Management Class Diagram ......................... 95
  4.4.2 Resource Management Screen Flow ........................... 95
  4.4.3 Storage of Resource Data .................................... 98
4.5 Photocast and Podcast Management Modules Implementation .... 98
  4.5.1 Photocast and Podcast Management Class Diagram .......... 99
  4.5.2 Photocast and Podcast Management Screen Flow ............ 101
4.6 Expert Finder Module Implementation ........................... 103
  4.6.1 Expert Finder Class Diagram ................................ 103
  4.6.2 Expert Finder Screen Flow ................................... 105
4.7 Resource Finder Module Implementation .......................... 107
  4.7.1 Resource Finder Class Diagram ............................... 107
  4.7.2 Resource Finder Screen Flow ................................ 108
4.8 Expert Answer Module Implementation ............................ 109
  4.8.1 Expert Answer Class Diagram ................................ 109
  4.8.2 Expert Answer Screen Flow .................................. 110
4.9 Photocasting and Podcasting Modules Implementation .......... 112
  4.9.1 Photocasting and Podcasting Module Class Diagram .......... 112
  4.9.2 Photocasting and Podcasting Module Screen Flow .......... 113
4.10 Summary .......................................................... 115

5 Evaluation ............................................................ 116
  5.1 User Evaluation .................................................. 117
    5.1.1 Methodology for User Evaluation ......................... 117
    5.1.2 User and Test Profiles ................................. 120
    5.1.3 Overall User Evaluation of the MobileHost CoLearn System . 120
    5.1.4 User Evaluation of the Expertise Management Module .... 121
    5.1.5 User Evaluation of the Expertise Finder Module .......... 122
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Thesis domains</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>Time bar of Web 2.0 buzz words [Garc06]</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>NetExpert’s Internal Architecture [SaPu01]</td>
<td>31</td>
</tr>
<tr>
<td>4</td>
<td>M-learning as a new stage of e-learning [adopted from [GGSm04]]</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Combined blogging and podcasting architecture, used at the University of</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>Washington [ABBa06]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Web Services Architecture [GGKS02]</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>SOAP message structure [adopted from [Cera02]]</td>
<td>42</td>
</tr>
<tr>
<td>8</td>
<td>WSDL document structure [adopted from [Cera02]]</td>
<td>43</td>
</tr>
<tr>
<td>9</td>
<td>Web Services based architecture for m-learning [TrRo03, TrRo04]</td>
<td>44</td>
</tr>
<tr>
<td>10</td>
<td>Mobile Web Services Architecture [SJPr06a]</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>Core architecture of the Mobile Host [Srir04, SJPr06a]</td>
<td>46</td>
</tr>
<tr>
<td>12</td>
<td>Java 2 Editions and Their Target Markets [Sun02b]</td>
<td>48</td>
</tr>
<tr>
<td>13</td>
<td>J2ME Configurations and Profiles [adopted from [Keogh03]]</td>
<td>49</td>
</tr>
<tr>
<td>14</td>
<td>Expert Search Scenario</td>
<td>51</td>
</tr>
<tr>
<td>15</td>
<td>Overall Use Case Diagram</td>
<td>54</td>
</tr>
<tr>
<td>16</td>
<td>Overall Architecture of the MobileHost CoLearn System</td>
<td>55</td>
</tr>
<tr>
<td>17</td>
<td>Hierarchy of the MobileHost CoLearn System Modules</td>
<td>57</td>
</tr>
<tr>
<td>18</td>
<td>Use Case Diagram of the Expert Management Module</td>
<td>59</td>
</tr>
<tr>
<td>19</td>
<td>Use Case Diagram of the Resource Management Module</td>
<td>61</td>
</tr>
<tr>
<td>20</td>
<td>Use Case Diagram of the Photocast and Podcast Management Modules</td>
<td>63</td>
</tr>
<tr>
<td>21</td>
<td>Single Path Flow of the Expert Finder Request</td>
<td>65</td>
</tr>
<tr>
<td>22</td>
<td>Complex Flow of the Expert Finder Request</td>
<td>65</td>
</tr>
<tr>
<td>23</td>
<td>Use Case Diagram of the Expert Finder Module</td>
<td>67</td>
</tr>
<tr>
<td>24</td>
<td>Expert Rating SOAP Request Message (Level 3 to Level 0)</td>
<td>69</td>
</tr>
<tr>
<td>25</td>
<td>Use Case Diagram of the Resource Finder Module</td>
<td>70</td>
</tr>
<tr>
<td>26</td>
<td>Resource Finder SOAP Request Message</td>
<td>71</td>
</tr>
<tr>
<td>27</td>
<td>Resource Finder SOAP Response Message</td>
<td>72</td>
</tr>
<tr>
<td>28</td>
<td>Use Case Diagram of the Expert Answer Module</td>
<td>73</td>
</tr>
<tr>
<td>29</td>
<td>Expert Answer SOAP Request Message</td>
<td>75</td>
</tr>
<tr>
<td>30</td>
<td>Use Case Diagram of the Photocasting Module</td>
<td>76</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

31 Available Photocasting Channels SOAP Response Message ................. 78
32 Subscribe to Photocasting Channel SOAP Request Message ............... 79
33 Receive Photocast SOAP Request Message .................................. 80
34 Comment Photocast SOAP Request Message .................................. 81
35 Retrieve Photocast Comments SOAP Response Message ................. 82
36 Target J2ME Configuration and Profile ...................................... 85
37 Overall MobileHost CoLearn Class Diagram ................................ 89
38 Overall Screen Flow of the MobileHost CoLearn Application ............ 90
39 Class Diagram of the Expert Management Module ........................... 92
40 Screen Flow of the Expert Management Module .............................. 93
41 Storage of Experts’ Data .......................................................... 94
42 Class Diagram of the Resource Management Module ....................... 96
43 Screen Flow of the Resource Management Module ............................ 97
44 Storage of Resource Data: Books Sample .................................... 99
45 Class Diagram of the Photocast and Podcast Management Modules ........ 100
46 Screen Flow of the Photocast Management Module ............................ 102
47 Class Diagram of the Expert Finder Module ................................... 104
48 Screen Flow for My Requests of the Expert Finder Module ................ 106
49 Class Diagram of the Resource Finder Module ................................ 108
50 Class Diagram of the Expert Answer Module ................................... 110
51 Screen Flow of the Expert Answer Module ..................................... 111
52 Class Diagram of the Photocasting Module .................................... 113
53 Screen Flow of the Photocasting Module ...................................... 114
54 ISO 9126 software quality measures [Cons06] ................................ 116
55 Timestamps Taken for Performance Analysis .................................. 125
56 Overall Scalability Performance Analysis of the Expert Search Web Service ... 127
57 Granular Performance Analysis of the Expert Search Web Service .......... 127
58 Performance Analysis of the Resource Finder Web Service Server ........ 128
59 Performance Analysis of the Resource Finder Web Service Client ........ 129
60 Performance Analysis of the Problem and Expert Answer Web Services ... 130
61 Expert Search SOAP Request Message (Level 2 to Level 3) ............... 137
62 Expert Search SOAP Response Message (Level $k+1$ to Level $k$) ........ 138
63 Problem SOAP Request Message .............................................................. 138
64 Screen Flow for Others’ Requests of the Expert Finder Module ................. 140
65 Screen Flow of the Resource Finder Module ............................................ 141
List of Tables

1. Mobile device constraints and their impact on the design process of m-learning applications [PaMa06] .................................................. 37
2. Network constraints, impacting the design process of m-learning applications [NLVS04, PaMa06] .................................................. 38
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AWT</td>
<td>Abstract Window Toolkit</td>
</tr>
<tr>
<td>CBS</td>
<td>Cell Broadcast Service</td>
</tr>
<tr>
<td>CDC</td>
<td>Connected Device Configuration</td>
</tr>
<tr>
<td>CLDC</td>
<td>Connected Limited Device Configuration</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CVM</td>
<td>Sun’s C Virtual Machine</td>
</tr>
<tr>
<td>D-Learning</td>
<td>Distance Learning</td>
</tr>
<tr>
<td>E-Learning</td>
<td>Electronic Learning</td>
</tr>
<tr>
<td>FCOP</td>
<td>File Connection Optional Package</td>
</tr>
<tr>
<td>GCF</td>
<td>Generic Connection Framework</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphics Interchange Format</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communications</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GUID</td>
<td>Globally Unique Identifier</td>
</tr>
<tr>
<td>HREF</td>
<td>Hypertext Reference</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMC</td>
<td>Internet Mail Consortium</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>J2ME</td>
<td>Java 2 Platform, Micro Edition</td>
</tr>
<tr>
<td>J2SE</td>
<td>Java 2 Platform, Standard Edition</td>
</tr>
<tr>
<td>JAD</td>
<td>Java Application Descriptor</td>
</tr>
<tr>
<td>JAR</td>
<td>Java ARchive</td>
</tr>
<tr>
<td>JCP</td>
<td>Java Community Process</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>JSR</td>
<td>Java Specification Request</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>KB</td>
<td>Kilobyte</td>
</tr>
<tr>
<td>KVM</td>
<td>Sun’s K Virtual Machine</td>
</tr>
<tr>
<td>MB</td>
<td>Megabyte</td>
</tr>
<tr>
<td>M-Learning</td>
<td>Mobile Learning</td>
</tr>
<tr>
<td>MIDI</td>
<td>Musical Instrument Digital Interface</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MIDP</td>
<td>Mobile Information Device Profile</td>
</tr>
<tr>
<td>MIDlet</td>
<td>Mobile Informational Device Application</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MMPI</td>
<td>Mobile Media Application Programming Interface</td>
</tr>
<tr>
<td>MP3</td>
<td>MPEG-1 Audio Layer 3</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>OBEX</td>
<td>OBject EXchange</td>
</tr>
<tr>
<td>OTA</td>
<td>Over-The-Air</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDAP</td>
<td>Personal Digital Assistant Profile</td>
</tr>
<tr>
<td>PhD</td>
<td>Doctor of Philosophy</td>
</tr>
<tr>
<td>PIM</td>
<td>Personal Information Management</td>
</tr>
<tr>
<td>PIMOP</td>
<td>Personal Information Management Optional Package</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>ROM</td>
<td>Read Only Memory</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SUMI</td>
<td>Software Usability Measurement Inventory</td>
</tr>
<tr>
<td>SUS</td>
<td>System Usability Scale</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol / Internet Protocol</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WAV</td>
<td>Waveform Audio</td>
</tr>
<tr>
<td>WMA</td>
<td>Wireless Messaging Application Programming Interface</td>
</tr>
<tr>
<td>WS</td>
<td>Web Service</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>WTK</td>
<td>Wireless Toolkit</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1 Introduction

This section presents the motivation for the master thesis. It discusses the currently existing problems and challenges in the domain of mobile collaborative learning, and presents a sketch of a potential solution. It gives a brief overview of the research fields, involved in the proposed solution, and presents the outline of the thesis documentation.

1.1 Motivation

Throughout the years there has been a clear evolution from traditional education through distance learning and electronic learning to mobile learning. Traditional education is performed in classrooms, where the teacher presents the learning material to a group of students. The biggest advantages are the possibility for direct contact between teacher and students and the ability for immediate feedback. However, traditional learning also has disadvantages: the content and presentation of the learning material depend highly on the abilities of the teacher; the physical presence of the students in the classroom is required, or otherwise they miss the training material. These disadvantages lead to search for new and more effective educational methods.

Distance learning, or d-learning, appears as a new teaching methodology, which effectively delivers education to students, without the need for them to be physically present at a certain time and on a certain place. It provides flexibility and convenience, especially if the learner has other commitments. The first stage of the evolution of distance learning is characterized with the asynchronous exchange of printed media between teachers and students, sent by the post.

With the advancements in electronic devices and network technologies, another stage of distance learning appears, i.e. electronic learning, or e-learning. E-learning can be used either as an enhancement to traditional learning or as a completely electronic-mediated learning environment, which reinforces the learner experience through the use of network-connected electronic devices, such as personal computers. It provides greater adaptability to the learner needs by utilizing multimedia, which can be paused at any time, reversed and replayed as needed.

The rapid development of wireless communication technologies and the advent of mobile devices in everyday life lead to a follow-up of e-learning, called mobile learning, or m-learning. M-learning provides learners the flexibility for not keeping a fixed position during the learning process through the use of portable devices, such as mobile phones, smart phones and personal digital assistants, communicating through versatile network technologies, such as IEEE 802.11, GSM, WAP, GPRS, UMTS, Bluetooth, IrDA.

At the same time, with the evolution from traditional learning through d-learning and e-learning to m-learning, the view of learning as an isolated individual activity changes to the view of learning as a collaborative process. Learning is not viewed any more as being instructor-centered, but learner-centered; it is not viewed any more as a process
of individual assimilation of knowledge, but as a process of participation in communities of practice, in which learners at various performance levels **work together towards achieving a common goal**. Collaborative learning targets not only collaborative improvement of knowledge within a field of practice, but also attainment of higher-order thinking skills and acquisition of problem solving abilities.

Talking about knowledge sharing, special emphasis has to be placed upon Web 2.0 and social software. **Web 2.0** refers to a perceived second generation of Web based services, which emphasize online media sharing and collaboration. The various genres of **social software**, such as photccasting and podcasting, social networking and social tagging, wikis, blogs, etc. have tremendously changed the Internet over the last few years, bringing along new opportunities for community building and collaborative learning.

Fueled by the expanding adoption of the Internet nowadays, the **overflow of information** on the Web is continuously increasing. Coping with such large quantities of data can be pretty hard. Searching for a particular issue can take a great deal of efforts and time. Moreover, some of the most **innovative and valuable information is not made available online, but lies within groups of practice**. Locating the right person who can provide us with the right information or the right resources is the best way to find the data that we need.

As collaborative learning is a learner-centered process, in which people learn from each other by exchanging resources, it brings along the need for getting out of a centralized system and having seamless **interoperable way for resource provisioning among many learners**. Needless to say, **web services** - specifically distributed services that process XML-encoded SOAP messages, sent over HTTP, and described using WSDL - are by far the most broadly accepted interoperable way for fulfilling this need. HTTP and SOAP provide communication and message interoperability. **WSDL** provides the description of the service to support interoperability between development tools, complementing communication interoperability with the ability to exchange interface definitions.

With the improvement of the characteristics of mobile devices and the development of communication technologies, a natural direction in the domain of web services is exploring the potential use of **mobile devices as web service providers**. There is already quite a lot of research on web service provisioning from standard desktop systems; however, there is still no extensive research on the web service provisioning from mobile devices with limited resources, such as mobile phones.

There are currently no projects, or at least none, known to the author, which utilize mobile web service provisioning for fulfilling the learner’s needs. The developed MobileHost CoLearn system is the first one of its kind, which cares for the demands of large collaborative learning groups in mobile networks, by enabling learners to find experts in particular fields, retrieve their tagged literature resources, inquiry them particular questions regarding problems encountered, and automatically receive their photocasting and podcasting content as soon as it becomes available.

The thesis combines the latest developments in the related research domains: collaborative
learning, mobile learning, mobile social software and mobile web service provisioning, as shown in Figure 1. Since this thesis is all about supporting collaborative learning activities, the base of the thesis is the domain of collaborative learning. The aim of the thesis is to provide ways of supporting mobile collaborative learning, and therefore, the domain of mobile learning is on top of the collaborative learning domain. Since the proposed way of supporting mobile collaborative learning is by using mobile social software, it is the next research field in the stack. As the overall goal of the thesis is to provide innovative ways for supporting mobile collaborative learning processes, on top of these two domains is the domain of mobile web service provisioning.

![Figure 1: Thesis domains](image)

1.2 Thesis Objectives

Searching for particularities within the continuously increasing information overflow can take a great deal of efforts and time. Moreover, the most valuable and innovative information is hard to find, as it lies within communities of practice. Locating the right person, who can provide us with exactly the information that we need and who can help us solve exactly the problems that we come upon, is the best way to learn forward.

This master thesis presents a novel approach to expertise finding within a truly collaborative mobile learning environment. It enables the users to find experts in particular fields, who can provide them with exactly the information that they need and who can solve the specific problems they encounter. The discovery of experts is not targeted only within the framework of the user's social network, but also within the social networks of his acquaintances, and the social networks of the acquaintances of his acquaintances, and so forth. Such an expert finder flow will usually lead to the discovery of more than one potential expert, and the user's subjective decision who of them is the most knowledgeable one can be based either on
the rating for the expert’s level of expertise in the field, or on the path that the expert finder request has travelled before reaching the respective expert. After having found an expert, the user is provided with all the necessary information in order to be able to contact him for further assistance and to ask him questions regarding particular issues.

Alongside the valuable knowledge that flows within the system from the experts to the non-experienced users, the system supports the retrieval of a variety of resources. Among the most valuable literature types for each learner are the articles, inproceedings, proceedings, books, URLs, master and PhD theses, and unpublished resources. For the easy retrieval of specific resources, users should be able to tag them with particular keywords, and, as tagging is subjective and relevant, the user should be provided the opportunity to set the relevance of a tag to a resource. For this purpose, a three-level scale of relevance of a tag to a resource is employed.

Except retrieving specific resources at the time when they are needed, the system maintains image and audio resources within photocasting and podcasting channels, through which they are automatically distributed to all subscribers, as soon as they become available. This is one of the most suitable ways for sharing up-to-date knowledge, as the amount of information that can be conveyed through such broadcasting channels is enormous. In addition to employing the tagging mechanism, the leaving of feedback to broadcast resources should be also available, as well as the opportunity to retrieve the comments, that other learners have left for a resource.

Regarding the technical aspects of the thesis, the proposed MobileHost CoLearn system is based on the concept of web service provisioning from mobile phones. It takes full advantage of the latest developments in the telecommunication domain in terms of transmission rates and resource capabilities. The networking technology, which is used for communication between the web service provider and web service requesters, is GPRS, as it provides excellent speed, four times faster than GSM, and allows learners to be always connected, so that services are quick and easy to access.

The seamless interoperable way of expert and knowledge discovery has been achieved through truly interoperable web services and other widely accepted standards. The MobileHost CoLearn system is the first of its kind that adapts mobile web services for collaborative learning, bringing the benefits of the latest technological developments to the average learner.

1.3 Thesis Outline

The motivation for the master thesis is presented in Section 1. The thesis objectives are clearly defined. Current problems in the concerned research fields and suggested ways to overcome these problems are presented.

Section 2 discusses the current achievements in the research fields, related to this master thesis. Basic concepts of collaborative learning are introduced, followed by an overview of the e-learning and m-learning concepts. Web 2.0 and social software applications are examined and their potential use for supporting community building and collaborative learning is
argued. Existing expertise finder and knowledge management systems are reviewed. Web services are explained, including their protocols and standards, with an emphasis on mobile web service provisioning. The Mobile Host project, which serves as a base for this master thesis, is presented.

Section 3 works out in detail the conceptual approaches behind the developed MobileHost CoLearn system. It starts with a presentation of the scenarios for potential use of the system, followed by the overall system architecture and the hierarchy of the developed modules for collaborative mobile learning. The functional and non-functional requirements, which have to be fulfilled by the system as a whole, by its modules and web services, are systematically discussed.

Section 4 elaborates on the implementation of the MobileHost CoLearn system. It reviews the technologies and tools, which have been used. The overall implementation of the main collaborative learning modules is given, followed by a discussion of the implementation particularities of each of the modules, their sub-modules and their web services.

Section 5 discusses the evaluation of the developed MobileHost CoLearn system, including user evaluation and performance analysis. The methodologies employed for evaluating the user satisfaction and the system performance are presented first, followed by the results from the conducted evaluation.

Section 6 summarizes the results of the master thesis and outlines various perspectives, opening further research potential.
1 INTRODUCTION
2 State of the Art

As discussed in section 1.2, the thesis combines technologies from four research fields: collaborative learning, m-learning, mobile social software and mobile web service provisioning. This section reviews the state-of-the-art technologies in each of these domains, as well as current solutions of the above discussed problems, existing in the market and in research projects.

2.1 Collaborative Learning

As this master thesis is all about supporting collaborative learning through mobile provisioning of web services, the start point that should be discussed is the concept of collaborative learning. Going through an overview of this concept, the learning theories, which are related to it, are discussed.

Passive approaches to learning view it as a process, where learners receive and assimilate knowledge individually, while active approaches view learning as a social process, which utilizes communication with other learners [Hiltz98]. Collaborative or group learning refers to an instruction method, in which students at various performance levels work together in small groups towards a common goal [Gokh95]. It moves away from the view of learning as an isolated activity and proves that learning is a rich, collaborative and conversational experience, whether in classrooms, homes or in the streets [NLVS04]. The shared environment gives the students the opportunity to engage in discussions and take responsibility for their opinion.

There are different theories to learning, such as behaviourism, cognitivism, constructivism, connectivism, situated learning, etc. Three of these approaches, regarding collaborative learning, namely social constructivism, connectivism and situated learning, are discussed briefly in the next paragraphs.

According to the constructivism theory, learning is an active process, in which learners actively construct or build new ideas or concepts, based on both their previous and current knowledge [NLVS04]. In particular, the social constructivism theory, which was developed by post-revolutionary soviet psychologist Lev Vygotsky, emphasizes the collaborative nature of learning. Vygotsky argues that all cognitive functions originate through social interactions, and must therefore be explained as products of such interactions [BTRC06]. Learning is not simply a process of assimilation of new knowledge by learners; it is the process by which learners are integrated into a knowledge community [BTRC06].

The connectivism theory of learning is quite new, introduced by George Siemens in the last couple of years [Siem04, Siem05, Siem06]. It states that learning is a network forming process, based on the learner’s ability to create a network of ideas and to form connections between people and content, which leads to the notion that the capacity to know more is more critical than what is currently known [Siem04]. Knowledge may rest not only internally inside the learner, but also within his network of connections, as well as in non-human appliances. It
is formed by aggregating diverse, often opposing, viewpoints. The ability to see connections between ideas and concepts is a core skill. Connections, not content, are the starting point of the learning process; content is often the by-product of the learning process [Siem05].

According to the situated learning theory, first proposed by Jean Lave and Etienne Wenger, the acquisition of knowledge takes place in the context of social relationships, i.e. in a community of practice [LaWe91]. Learning is viewed as a social participation in such communities: groups of individuals, participating in communal activity, and experiencing/continuously creating their shared identity through engaging in and contributing to the practices of their communities [Weng98]. At the beginning the participation is peripheral and then increases gradually in engagement and complexity [Smith03]

2.2 E-learning

E-learning, which enhances the learner experience through the use of mobile devices, is growing fast both in corporate and educational learning environments. This section starts with an overview of the e-learning domain, followed by a discussion of the functionalities of e-learning platforms, grouped into 4 categories: access to resources, specific e-learning services, common services, presentation. Last, but not least, the new trend of using e-learning platforms to support collaborative learning activities, i.e. computer supported collaborative learning (CSCL), is presented.

2.2.1 Overview

The rapid development of communication technologies and the rising computer knowledge of the learners lead to the emergence of electronic learning, also called e-learning. E-learning is an all-encompassing term, generally used to refer to computer-enhanced learning. Online courses, web-based education, computer supported training and virtual university are widely used terms, all of which represent e-learning [TrRo04]. E-learning can be viewed as another stage of the evolution of d-learning.

The field of e-learning has two main facets. The first facet examines the use of technology to support distance learning, where the whole learning process is technology mediated, as the learners and instructors are physically separated and they never or rarely meet face to face for discussion. The second aspect is concerned with enhancing the learning experience with the help of information technology, where the traditional learning approaches can be supported with complementary services, such as online delivery of the learning materials, support for collaborative work, virtual communities, etc. In many cases, both aspects are simultaneously present [TrRo04].

2.2.2 Computer Supported Collaborative Learning

Computer supported collaborative learning (CSCL), as the name itself suggests, is a research field on supporting collaborative learning with the assistance of computer artifacts. It is
based on the idea that computer applications can enhance advanced socio-cognitive processes for knowledge building and knowledge sharing [PLHa02]. CSCL communities can be defined as scientific communities of practice, i.e. communities of practice with members working in a common field of research but being distributed across disciplines, organizations, cultures and geographical regions [KiWe05]. CSCL facilitates group processes in ways, which are not achievable by face to face communication; however, it does not aim at replacing this kind of communication. CSCL systems support communicating ideas and information, accessing information and documents, and providing feedback on problem-solving activities.

2.3 Web 2.0 and Social Software

Talking about computer-based collaboration, an emphasis has to be placed on Web 2.0 and the different types of social software, which can be used to support it. There has been a fundamental shift in the paradigm of how users think about and use the web. Instead of reading static web pages, users are now cataloging their personal libraries, organizing their favorite bookmarks, writing online documents, and sharing information with others through new generation social software. What began with blogs and wikis, has blossomed into an all-embracing phenomenon of sharing, collaboration, and user involvement [Kros06]. This section gives an overview of the main principles of Web 2.0, which encompasses the new generation of internet based services. Next, the types of social software applications and services, such as blogs, wikis, social tagging, social bookmarking, media sharing, etc. are discussed, followed by the potential use of each of these types of social software for building virtual communities and for supporting collaborative learning.

2.3.1 Web 2.0

The term Web 2.0 was conceived by the tech guru Tim O’Reilly from O’Reilly Media and Dale Dougherty from MediaLive in 2004 [Kros06] to refer to a perceived or proposed second generation of internet based services, and was introduced during the first Web 2.0 conference in October 2004 [BaOr04]. This new generation includes internet-based services, such as social networking sites, wikis, folksonomies and communication tools, which emphasize online collaboration and sharing among users.

The main principles of Web 2.0, as illustrated visually in the meme map of Web 2.0, developed at a brainstorming session during FOO Camp [O’Re05], include: The web as a platform, An architecture of participation, Harnessing collective intelligence, Data as the driving force, End of the software release cycle, Software above the level of a single device, Leveraging the power of the long tail, Design for reusability, Rich user experiences.

The time bar of Web 2.0 buzz words, as proposed by Jrgen Garca [Garc06], is presented in Figure 2. The figure shows the predecessors of Web 2.0, some of which are now its main constituents. As can be seen, Web 2.0 is based on concepts, evolving for a period of more than 10 years. The proposed buzz words suggest that Web 2.0 is all about sharing, collaboration and community building. How do these processes exactly happen? Through the use of web
applications and services, which can be cumulatively called "social software".

2.3.2 Social Software

Social software, or social media, refers to various, loosely connected types of applications that allow individuals to communicate with one another and to track discussions across the web as they happen [Tepp03]. Social software has seen a tremendous jump in usage over the past few years, with blogs, wikis, social networking and tagging systems proliferating alongside continued use of newsgroups, bulletin boards, instant messaging, collaborative filtering systems and other existing socially-oriented software systems [CHSm06].

**Instant messaging** (IM) supports near synchronous communication among parties, allowing them to share ideas across long distances almost instantly [VMEK04]. IM applications, such as ICQ, allow people to communicate in relative privacy. One can add people to his contact list, know when they are online, send them messages and respond to their messages. Whereas IM applications tend to primarily support messaging between two participants, text chat applications, such as the Internet Relay Chat (IRC), tend to support multiple participants [ONMa03]. Users can join chat rooms and communicate publicly with many people at once. They can also create their own rooms on any topic and invite others to join them. Often there is a steady stream of people entering and leaving rooms.

**Discussion forums** usually contain many different categories, organized in a hierarchy, according to topics and subtopics. They contain topic posts from different users, which are then discussed in threaded replies from other users [Gord06]. One can post a message for others to review, comment on others’ posts, or even comment them with special formatting in his own post. Many forums are public, a few are private.

**Blogs**, or weblogs, consist of a set of web pages with dated entries in a reverse chronological order, usually with sidebars of profile information, maintained and published with the help of a popular blog authoring tool [KNRT04], such as Blogger, LiverJournal, WordPress, TypePad, MySpace, etc. They contain primarily textual information, but many include also photos and other multimedia content [SNGS04]. New variants of blogs are gaining popularity every day, such as **phlogs** (short for photoblog) and **vlogs** (short for videoblog), where the
primarily content are photos and videos respectively, and **moblogs** (mobile blogs), which allow users to post content directly from their mobile device to the blog [CSKC06].

A **wiki** system, such as Wikipedia, is a network of web pages, containing information, which can potentially be created, modified or deleted by any user, using relatively easy wiki syntax [Rein06]. Unlike blogs, wikis let their users create new categories of information on the fly, and allow for editing of other people’s comments. A blog is better for recording a process, while wikis have the final result recorded, not the work process that went into reaching this result [Tepp03].

**Podcasting** is stemming from emerging trends in portable information technology, such as iPods and smartphones, but it can be also used with any other portable MP3 device or even desktop computers [RaZh06]. After the user subscribes to a desired RSS feed through a podcatcher- media aggregator software, such as iTunes- the podcatcher software automatically downloads the new content to the user’s device, whenever there are updates. The content can be versatile, such as text, photos, audio or video. Podcasting usually refers to the automatic distribution of audio content, while **vodcasting** refers to video content.

**Media sharing** has been used for a long time for exchanging data between users, who are sharing almost everything: ideas, goals, wish lists, hobbies, photos, videos, bookmarks [CSKC06].

**Social tagging** systems enable users to add short free form labels or keywords, called tags, to versatile internet resources, without relying on a controlled vocabulary [MNBD06]. They help people remember and organize their versatile information: e-mail, such as gmail; websites, such as del.icio.us; photos, such as Flickr; blogs, such as Technorati; research papers, such as CiteULike [SLR*06].

**Social bookmarking** websites, such as del.icio.us and dig, allow users to post their list of favourite bookmarks or websites, so that others can search for them and view them. Much like social bookmarking, **social citations**, such as CiteULike, aim towards allowing academics to post citations from articles they find interesting. These citations can be organized into predefined or user defined categories.

### 2.3.3 Potential Use of Social Software for Community Building and Collaborative Learning

Largely, Web 2.0 is all about community and collaboration through social software. As discussed in section 2.3.2, social software refers to various types of applications, which allow individuals to collaborate through computer-mediated communication. It is also argued that social software can be used for building virtual communities and for supporting collaborative learning [CSKC06, CJFW07, Leis03, Leis04].

### 2.3.3.1 Social Software for Building Virtual Communities

A community is a social grouping that exhibits to varying degrees shared spatial relations, social conventions, a sense of membership and boundaries, and an ongoing rhythm of social
interaction. A virtual community is a community, which emerges from the webs of personal relationships in computer-mediated networks [ZhWe03]. It is a group of individuals or organizations that share a set of common interests and use electronic media to communicate with each other. The communication is not restricted to time or location.

Social software forms people into groups with a bottom-up approach, enabling them to organize themselves into a network, based on their preferences [Leis03, Leis04]. In a collaborative virtual environment, community members pursue both their own goals and the common social goals. To achieve these goals, the members need to collaborate and to coordinate their activities.

The social aspects of blogs, phlogs, vlogs and moblogs is an active research area, trying to give an explanation for the reasons bloggers blog and how relationships with readers are managed [SNGS04, NSGu04], and how blogspace evolves over time [KNRT03, KNRT04]. Blogs support the building of virtual communities around people or interests through the mechanisms of comments, blogrolls and trackbacks/pingbacks. Comments allow creation of discussion forums. Blogrolls indicate the social relationships between bloggers, using the XHTML Friends Network standard. Trackbacks and pingbacks support notification of interblog conversations.

Wikis are an effective tool for mass collaborative authoring, as they allow everyone to add, modify and delete content on the fly, usually even without the need for registration, which makes wikis very easy to use. Podcasting and vodcasting allow community members to seamlessly exchange audio or video resources.

Social tagging, social bookmarking and social citation systems enable users to organize and share their tagged resources, bookmarks and citations with users with similar interests. They are a powerful tool for social navigation, not only letting people share information, but also helping them easily discover information, contributed by other community members [SLR*06]. Instead of imposing controlled vocabularies or categories, tagging systems’ vocabularies emerge organically from the tags chosen by the community members.

The different types of social software support the bottom-up building of virtual communities. The formation of a virtual community usually starts from a single person, who has a problem and seeks advice from the others, or who would like to share his knowledge and experience with the others. Let us get a blog as an example. The user posts an entry in his blog, followed by a couple of comments by other users, maybe people he knows or complete strangers, who reply to him by adding a comment in his blog, or trackbacking his entry and discussing the matter in their own blogs. As time passes by, more and more people get involved in the discussion. Thus, a bottom-up virtual community is build.

2.3.3.2 Social Software for Supporting Collaborative Learning

As discussed in section 2.1, collaborative learning refers to educational approaches, involving joint intellectual effort by a couple of learners, working together in searching for understanding, meaning or solutions. Social software supports the process of collaborative learning by providing means of personal knowledge and information management, distributed learning
object repositories, instant messaging, resource sharing, etc., thus enabling the creation of collective intelligence [Leis03].

*Instant messaging* and *text chat* enable learners to communicate efficiently, almost instantly, without restrictions to physical location. *Discussion forums* are a very suitable medium for problem solving, as they allow users to post their questions or problems and allow other learners to explain their point of view and give suggestions for solution or improvement.

*Blogs* and blog variations, such as *phlogs, vlogs* and *moblogs*, can be used by learners as a personal knowledge and information management tool. Blogs let learners express their knowledge or opinion about a problem, and let others comment on their thoughts, or create an interblog conversation by using the mechanisms of trackbacking / pingbacking.

As *wikis* are an effective and easy to use tool for mass collaborative authoring, they are very suitable for collaborative learning. Wikis allow learners, who work together on a project, to post their own thoughts, review and modify the thoughts of the others on the fly, thus achieving collaboratively the final result.

*Discussion forums* are another tool for expertise sharing, as it allows users to search for help on a variety of topics by posting questions in different threads, and wait for answers from the others.

*Podcasting* and *vodcasting* are additional channels for learners to exchange resources and their knowledge and provide audio or video feedback. These channels can assist non-native speakers to learn a new language, enable instructors to review training or lectures, replace full classroom or online sessions when content simply requires delivery, provide supplementary content or be part of a blended solution, provide feedback, etc. [Leis05]. Perhaps the biggest advantage of podcasting and vodcasting is the support for syndication of distributed learning object resources.

*Social tagging* systems have the potential to improve the personal organization of content, to facilitate the search and enhance reputation systems, while introducing new modalities of social communication and opportunities for data mining [MNBD06]. *Social bookmarking* and *social citations* allow academics, who are working in similar areas, to connect and share resources, thus extending services, such as Citeseer, which rely mostly on the knowledge of experts.

### 2.3.4 Expertise Finding in Knowledge and Social Networks

The goal of many researchers and designers is to help people cope with large quantities of information [ViLi00]. The interest is of course fueled by the increasing adoption of the Internet. The Web as viewed through a regular browser appears to most as a very crowded information sewer. It is fairly obvious that there is room for improvement.

Expert finding systems are a type of recommendation systems, which are designed in order to facilitate the finding of people with specific knowledge in a certain problem domain [McDon03, ZhAc05, ZAAd07]. Such systems include Expertise Recommender [MDAc00], NetExpert [SaPu01], ReferralWeb [KSSh97a, KSSh97b], etc.
McDonald [McDo07] explores how information systems can be augmented to assist users in finding other individuals who are likely to have specialized, expert information that they need. It describes a field study that considers the social and cognitive mechanisms that people use to find candidate sources of expertise. These mechanisms are the basis for a recommender system that can help users find expertise.

NetExpert [SaPu01] is a multiagent expertise location system that allows localizing experts within a real or virtual community by exploiting both knowledge and social networks. It replicates the process of social and knowledge network building at a community or organization level. In doing so, it is able to connect several networks and put into contact expertise that otherwise would remain hidden. NetExpert is integrated and used with I2CAT Collaboratory [SVBP01], an expertise recommendation system which works on Knowledge Pump metaphor [Glan98]. Each member of I2CAT Collaboratory contributes documents to the Common Repository and consults documents from it. This member’s activity is used in conjunction with analysis of member’s personal webpages to build a Knowledge profile that contains a description the expertise areas of any member of the community. Knowledge profile is presented using Vector Space Model with Tfidf [Salt86] technique and expertise is described using the most important terms appearing in the documents contributed by the user to the Collaboratory repository. From the knowledge profiles the social and knowledge networks are built and analysed. The internal architecture of NetExpert is shown in Figure 3.

![Figure 3: NetExpert’s Internal Architecture [SaPu01]](image)

### 2.4 M-Learning

As the whole world is going mobile with devices, which fit in our pockets and can connect us to a variety of information sources, there is considerable interest in exploiting the mobile technologies for educational use [NLVS04]. Going through an overview of m-learning, which is
performed while the learners are using mobile devices, such as notebooks, tablet PCs, PDAs, cellular phones and smart phones, communicating through versatile network technologies, such as IEEE 802.11, GSM, WAP, GPRS, UMTS, Bluetooth, IrDA, specific attention is paid to the critical success factors for the development of mobile learning applications, such as learner needs and requirements, mobile device and network constraints, quality components. Next, combining the m-learning domain with the collaborative learning domain, existing mobile social software projects for m-learning are presented.

2.4.1 Overview

In the last couple of years there has been a rapid increase in the development and usage of e-learning systems, both for academic and vocational training, addressing a broad variety of audiences. At the same time, the rapid development of wireless infrastructure and the advent of mobile devices in everyday life of people push the research to combine those two domains, which results in the emergence of m-learning [TrRo03].

Papanikolaou et al. define m-learning is a new paradigm, which creates a learning environment, in which ”the delivery of educational and/or training programs and/or materials by a mobile device and the dissemination of the material can be done in a synchronous or an asynchronous manner through the use of computer networking technology” [PaMa06]. In general, m-learning can be viewed as any form of studying, performed while the user is interacting with a mobile device. It can include anything from course materials, downloaded on your personal digital assistant, to online instructor facilitated training via laptop. It is a new stage of the progress of d-learning and e-learning, as shown in the learning evolution stages depiction in Figure 4.

![Figure 4: M-learning as a new stage of e-learning [adopted from [GGSm04]]](image)

2.4.1.1 Mobile Devices

M-learning is impossible without the use of mobile devices, such as cellular phones, smart phones, personal digital assistants, tablet PCs and notebooks, which vary significantly in terms of their functionality, size, price, etc. However, there are two things which unite all these devices: their mobility and wireless networking ability [GGSm04].
Cellular phones, also called mobile phones, are low class portable mobile devices, which connect to base transmitter and receiver stations via microwave radio frequencies, which in turn connect the user to the conventional telephone network. Cellular phones are mainly used for voice communication and short text messaging (SMS), and the newer models can also be used for multimedia messaging (MMS) and Internet access via the Wireless Application Protocol (WAP) or General Packet Radio Service (GPRS). The main disadvantages are the low memory capacity and low data transfer rate.

Personal Digital Assistants (PDAs) are lightweight handheld computers with significant processor power and communication capabilities. They are primarily used as personal organizers, as they usually provide a calendar with appointment scheduling functionalities and an address book. PDA devices usually have no keyboard, but an LCD display, on which handwritten text can be written with a digital pen and recognized by special software. The main operating systems for PDA devices are Palm and Microsoft Pocket PC, and typically used applications are word processor, spreadsheet, notepad, e-mail, multimedia player, etc.

Smart phones are hybrid devices, which combine the capabilities of cellular phones and PDAs, mainly using the Symbian and Windows Mobile operating systems. Smart phones are usually smaller than PDAs and bigger than cellular phones, do not have a full-sized keyboard and can recognize handwritten text. Besides being used for voice communication, smart phones are also famous with features, such as Internet and e-mail access, build-in camera, GPS navigation, sometimes support for PDF and Microsoft Office documents, etc.

Notebook computers, or laptops, are small portable computers with LCD displays. The newest models have the same computational power and capabilities as desktop computers, with the main differences being their small size and weight and support for wireless networking.

Tablet PCs are slate-shaped notebooks, considered to be one of the latest mobile devices. The main differences from notebook computers are the input modalities. A tablet PC has a touch screen and an on-screen virtual keyboard, which thanks to the digitizing tablet technology allow the user to use a stylus or digital pen, or a fingertip, to operate the computer, instead of a keyboard or a mouse. Other alternatives for data input include speech recognition and a standard keyboard. In terms of capabilities, the newest models are equal to notebook and desktop computers.

2.4.1.2 Communication Technologies

M-learning is also impossible without the use of communication technologies, such as GSM, WAP, IEEE 802.11, GPRS, UMTS, Bluetooth and IrDA, which vary significantly in terms of their data transmission range, ability, etc. Each of these technologies is represented in the next paragraphs [Eric06].

Global System for Mobile Communications (GSM) was first introduced in 1991 and is currently the leading digital cellular system, as it provides the best voice quality of the current digital wireless standards. GSM networks operate in more than 100 countries all over the world- on the 900 MHz and 1800 MHz wavebands in Europe, Asia and Australia, and
on the 1900 MHz waveband in North America, parts of Latin America and parts of Africa. GSM is a cellular network, which means that mobile phones connect to it by searching for cells (transmitter and receiver stations) in the immediate vicinity. GSM uses narrowband Time Division Multiple Access (TDMA) and eight simultaneous calls can occupy the same radio frequency. It simplifies data transmission to allow laptop and palmtop computers to connect to GSM phones and provides integrated voice mail, high-speed data, fax, paging and SMS capabilities, as well as secure communications.

**Wireless Application Protocol (WAP)** is a free unlicensed protocol for wireless communications, a de-facto industry standard, which primary application is to enable Internet access from a mobile phone or a PDA. It supports most wireless network standards, such as GSM, TDMA, CDMA, W-CDMA, etc. A WAP browser provides all the functionalities of a standard web browser, but it is simplified in order to operate under the hardware limitations of mobile devices. WAP devices understand the Wireless Markup Language (WML), which is a version of HTML optimized for small screens and navigation without a keyboard, and WMLScript, which is a client-side JavaScript-like scripting language.

**General Packet Radio Service (GPRS)** is a mobile data service, available to users of GSM and IS-136 mobile phones. GPRS provides more than four times faster speeds than conventional GSM systems by using unused TDMA channels. GPRS can be utilized for various services, such as SMS and MMS, WAP access, e-mail and web access. Using a packet-linked technology, subscribers are always connected and always online, so services are quick and easy to access. 2G cellular systems, combined with GPRS, are often referred to as 2.5G, that is, a technology between the second (2G) and third (3G) generations of mobile telephony.

**Universal Mobile Telecommunications System (UMTS)** is a global third generation (3G) mobile system, which uses Wideband Code Division Multiple Access (W-CDMA) as the underlying air interface. UMTS is sometimes referred to as 3GSM, in order to differentiate it from other network technologies, emphasizing the combination of a 3G technology and the GSM standard.

**Bluetooth** is an industrial specification for wireless Personal Area Networks (PANs). It is a short-range radio technology, which facilitates both voice and data communication, and offers a possibility for exchange and synchronization of information between versatile devices, such as desktop computers, mobile phones, notebooks, digital cameras, etc.

**Infrared Data Association (IrDA)** defines a suite of protocols for short range exchange of data over infrared light, for uses such as PANs. It facilitates the exchange of data between devices, such as notebooks, printers, smart phones and PDAs, which are 1 to 2 meters apart for high-power devices and 20 to 30 cm apart for low-power devices. IrDA devices typically have throughput of up to 115.2Kbps or 4Mbps. The biggest disadvantage of IrDA is the requirement for direct line of sight between the communicating devices.

**IEEE 802.11** refers to a set of Wireless Local Area Network (WLAN) standards, developed by working group 11 of the IEEE LAN/MAN Standards Committee (IEEE 802). It is also known by the brand **Wi-Fi**, as the Wi-Fi Alliance certifies the interoperability between
802.11 devices. These WLAN technologies enable short-range packet data communication with very high data rates, up to 54 Mbps, between base stations and user terminals, or directly between terminals. There are several standards for the WLAN air interface, operating in different radio frequencies. Two of them are currently dominating standards, namely IEEE 802.11b, operating in the 2.4 GHz spectrum with a bandwidth of 11 Mbps, and IEEE 802.11g, operating in the 5 GHz frequency range with a bandwidth of 54 Mbps. Another standard, IEEE 802.11n, is under development for WLANs operating in the 2.4 GHz frequency and supporting up to 540 Mbps, and is planned for release in 2008.

2.4.2 Critical Success Factors for the Development of Mobile Applications

Since learning and teaching have been practiced for thousands of years, the transfer of these processes into the electronic environment requires a holistic, interdisciplinary approach. Papanikolaou et al. identify the following major areas of critical success factors for the development of mobile learning applications, which have to be incorporated into the engineering process: learner needs and requirements, mobile device and network constraints, quality components [PaMa06].

2.4.2.1 Learner Needs and Requirements

The first area of critical success factors, leading to a successful design and development of mobile learning applications, examines the satisfaction of the users’ needs and the adherence to the users’ requirements. In order to satisfy these requirements, the learning environment should:

- Adapt to the learner’s needs - The m-learning environment should be shaped according to the learner’s needs, which depend on his socio-cultural background, education level, skills and competences, acquired from previous education and training.

- Facilitate knowledge transfer - The sharing of information, experiences and views forms an indispensable part of the learning process and is very effective in disseminating knowledge and establishing a community of learners.

- Motivate participation - The non-intuitive nature of the environment, the reduced interactivity, etc. can result in dissatisfaction and frustration, making the student reluctant to use the virtual environment. Thus it should be constructed in a way that increases and retains the student interest.

- Encourage collaboration - In a real classroom, collaboration can be supported by face-to-face communication. However, in a virtual environment, the student can easily become isolated. This should be prevented by supporting and encouraging collaborative work, and promoting a sense of community.
• Offer basic problem solving mechanisms - Students should be able to receive support and help by simple mechanisms, such as contact with the instructor, online tutorials, reference to useful resources, access to a technical helpdesk.

• Combine learning processes - A combination of learning processes, such as analysis, synthesis, reasoning, judging, problem solving, collaboration, simulation, evaluation, presentation and relation, should be considered while developing an m-learning environment.

• Deliver target group specific tools - Vocational training requires different tools than academic training, and undergraduate training has different pedagogical targets than postgraduate training. For creating an efficient environment, target group specific tools should be utilized.

• Adhere to the basic mechanisms and functions of a real environment - The m-learning environment can be used either complimentary or in parallel to the e-learning environment or to the real classroom. In either case, the m-learning environment should adhere to the basic mechanisms and functions of the real environment.

2.4.2.2 Mobile Device Constraints

The second area of critical success factors for m-learning applications includes the characteristics and peculiarities of the different types of mobile devices. The most significant constraints, such as computational power, input modalities, screen and memory size, etc. and their impact on the design process of m-learning platforms for the different types of mobile devices, such as mobile phones, personal digital assistants and laptops, are represented in Table 1. As it can be seen, the mobile device constraints have a much higher impact on the design process of m-learning applications if the learner is using a mobile phone than if he is using a laptop.

2.4.2.3 Network Constraints

The network capabilities, resources and constraints should also be considered when designing learning platforms for mobile learners. Among the most significant factors, which influence the development and quality of m-learning applications and services, are the available network bandwidth, the coverage area, security, authentication, billing schemes, the type of service or application to be developed, etc. Table 2 examines these factors for the various telecommunication networks: cellular networks, such as GSM, GPRS and UMTS, and other wireless networks, such as IP-based WLANs, using the IEEE standard 802.11x technologies, Bluetooth and IrDA.

In order to prevent the degradation of the performance of mobile learning applications and services in terms of delay and loss of data, blocking and disconnection of service, what is required is the use of reliable mobile and wireless technologies. As the availability of mobile learning applications from anywhere depends on the roaming capabilities across multiple
Table 1: Mobile device constraints and their impact on the design process of m-learning applications [PaMa06]

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Mobile phone</th>
<th>PDA</th>
<th>Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limited computational power</td>
<td>(High)</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Limited memory</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Small screen</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Limited display resolution</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Graphical limitations</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Complicated text input mechanisms</td>
<td>High</td>
<td>Medium/High</td>
<td>Low</td>
</tr>
<tr>
<td>Small multifunction keypads</td>
<td>High</td>
<td>Medium/High</td>
<td>Low</td>
</tr>
<tr>
<td>Unfriendly user interface</td>
<td>High</td>
<td>Medium/High</td>
<td>Low</td>
</tr>
<tr>
<td>Limited battery life</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Non-volatile capacity</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Limited security</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Limited bandwidth</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Limited connection stability</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

heterogeneous networks, much research work is done in the field of interoperability of different wireless technologies.

2.4.2.4 Quality Factors

The fourth area examines the quality components, such as usability, functionality, reliability, efficiency, maintainability and portability, and their incorporation in the design and development process of mobile learning applications. These quality measures are defined in the ISO 9126 international standard for the evaluation of software, and divided into four parts, which address the quality model, external metrics, internal metrics and quality of use metrics [EAGL96].

2.4.3 Mobile Social Software for M-Learning

There are various projects for m-learning, carried out at universities all over the world, which pay a great amount of attention to the transformation of the classical pedagogical methods for the new environment. With the introduction of Web 2.0 and social software, new types of m-learning projects, utilizing social media for supporting collaborative learning, evolve, such as the ones at the University of Washington [ABBa06], Claremont Graduate University [RaZh06], University of Southern California [Wolf06], etc.

The most advanced of them seems to be the one, developed at the University of Washington [ABBa06]. It provides a scalable solution, combining blogging and podcasting for enhancing
## 2 STATE OF THE ART

<table>
<thead>
<tr>
<th>Factors</th>
<th>Wi-Fi</th>
<th>Bluetooth</th>
<th>IrDA</th>
<th>GSM</th>
<th>GPRS</th>
<th>UMTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transmission speed</strong></td>
<td>802.11a - 54 Mbps, 802.11b - 11 Mbps, 802.11g - 54 Mbps</td>
<td>1 Mbps</td>
<td>16 Mbps</td>
<td>9.6 Kbps</td>
<td>114 Kbps (effectively 30 Kbps)</td>
<td>2 Mbps</td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td>Local: 802.11a - 100 feet, 802.11b - 300 feet, 802.11g - 300 feet. Range can be extended by providing multiple access points.</td>
<td>Local: 30 feet. Range can be extended by piggybacking multiple devices.</td>
<td>Local: 3 feet</td>
<td>Wide</td>
<td>Wide</td>
<td>Wide</td>
</tr>
<tr>
<td><strong>Application</strong></td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data, voice</td>
<td>Data, voice</td>
<td>Multimedia</td>
</tr>
</tbody>
</table>

Table 2: Network constraints, impacting the design process of m-learning applications [NLVS04, PaMa06]
the learners’ experience, which automatically captures MP3 recordings of class lectures, uploads them to the provided blog spaces, and delivers them to the students’ mobile devices upon subscription to the podcasts. The architecture of the system is presented in Figure 5.

![Combined blogging and podcasting architecture](image)

Figure 5: Combined blogging and podcasting architecture, used at the University of Washington [ABBa06]

At the beginning of the quarter, staff at the University of Washington establishes online class blog spaces with integrated RSS feeds for the participating classes. After class locations, dates, times and durations are scheduled via a web interface, the entire recording process happens seamlessly in the background: a central podcast server automatically captures data from over 50 networked Instreamers (MP3 streaming devices), installed in the participating classrooms, which start and stop recording according to the defined schedule. Each captured MP3 file is added to the corresponding class blog via an XML-RPC request, just one or two minutes after the end of the class. Students can either listen to the recordings online or subscribe to the podcast by clicking on the provided link. Upon subscription, subsequent lectures are automatically delivered to the students’ mobile devices, be they computers, notebooks, MP3 players, or some cell phones.

### 2.5 Web Services

This section starts with an overview of web services, including the web services architecture, roles and operations, and benefits, followed by an elaboration of the web services protocols and standards, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Combining the web services domain with the learning domain, existing web services based projects for m-learning are discussed.
2.5.1 Overview

Web services are self-contained modular applications, which can be described, published, located and invoked over a network, generally, the World Wide Web [IBM00]. They are changing the Internet from program-to-user business-to-consumer interactions to program-to-program business-to-business interactions, through the usage of existing and emerging standards, such as HTTP, XML, SOAP, WSDL and UDDI [Hans03, GGKS02].

A web service performs a specific task or a set of tasks. It is described by the Web Services Description Language (WSDL), a formal XML notation, which provides all the necessary details, in order to be able to interact with the service, such as message formats that detail the operations, transport protocols, location, etc. [GGKS02]. A thorough discussion of WSDL is deferred to section 2.5.2.2. In order for a web service to be accessible to the public, its WSDL description should be published to a well-known repository, such as a Universal Description, Discovery and Integration (UDDI) registry. Users can search through these UDDI registries by using an API, find needed information about a service and then invoke it. A detailed presentation of UDDI is deferred to section 2.5.2.3.

Connecting the bits of the whole picture of any service-oriented environment, several essential activities need to happen [IBM00]:

1. The web service needs to be created and its interfaces and invocation methods need to be defined by using a service description language, such as WSDL.

2. The web service needs to be published by registering its descriptions and access policies to a well known registry like UDDI, so that potential users can locate it.

3. The web service needs to be located by the users by querying the registry and receiving its binding details.

4. The web service needs to be invoked by the users by using the information in the service description in order to be of any benefit.

5. The web service needs to be unpublished, if it is no longer available or needed.

2.5.1.1 Web Services Architecture

The basic architecture for web services consists of three main components or roles: service provider, service requester (client) and service registry. The architecture, which consists of these components and the pattern of communication between them, is presented in Figure 6.

2.5.1.2 Web Services Roles

As it can be seen from Figure 6, the web services architecture describes three roles: service provider, service requester and service broker [IBM00, GGKS02, Srir04]. The service provider is the owner of the service, who deploys it on the Web, describes it by using
Figure 6: Web Services Architecture [GGKS02]

WSDL and publishes its details to a publicly accessible registry. The service registry is a searchable repository of service descriptions, where service providers publish their services and service requestors find services and obtain binding information for these services, i.e. it serves as a broker between providers and requesters, helping them to find each other and set up a business relationship. The service requester, or client, is the one that requires certain functions to be fulfilled, uses an API to ask the service broker about services it needs and uses the returned results to bind to a particular service and invoke it.

2.5.1.3 Web Services Operations

As shown in the web services architecture from Figure 6, the communication between the basic web service components is performed by the operations publish / unpublish, find and bind [Srir04, GGKS02, IBM00]. Service providers advertise (publish) the availability of their service to a service broker. Service requesters interact with a service registry to discover (find) a set of needed services and negotiate with service providers to access and invoke their services (bind). When the service is no longer available, the service provider removes the service from the service registry (unpublish).

2.5.2 Web Services Protocols and Standards

This section discusses the web services protocols and standards, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI).

2.5.2.1 Simple Object Access Protocol (SOAP)
SOAP is a lightweight protocol for exchanging structured information between peers in a decentralized, distributed environment [W3C03b]. It is a simple XML based protocol that lets applications exchange information over different protocols, such as HTTP, FTP, BEEP, etc. [Srir04]. A SOAP message is an XML document that consists of a mandatory SOAP Envelope, which contains an optional SOAP Header and a mandatory SOAP Body. The SOAP message structure is shown in Figure 7.

![Figure 7: SOAP message structure](adopted from [Cera02])

The header entries contain information, such as authentication information, digital signatures, transaction management details, payment details, etc. SOAP Body encompasses the message payload, intended for the recipient of the SOAP message. The optional Fault sub-element of SOAP body specifies error information. An error could be generated at any SOAP intermediary along the message path while processing the SOAP message.

### 2.5.2.2 Web Services Description Language (WSDL)

WSDL is an XML based specification, defining how to describe web services. WSDL describes interface information describing all available public functions, data type information for all message requests and message responses, binding information about the transport protocol to be used, and address information for locating the specified web service [Cera02]. Using WSDL, a client can locate a web service and invoke any of its publicly available functions [W3C01]. The process can also be automated, enabling applications to easily integrate with new services with little or no manual code and interaction [Srir04].

WSDL documents define web services as collections of network points, or ports. The WSDL specification uses the six major elements, as shown in Figure 8. As it can be seen, WSDL
documents use the following main elements in the definition of network services: definitions, types, message, portType, binding and service [Cera02]:

- **definitions** - the root element of the WSDL file, embracing the rest of the elements; defines the name of the web service and declares the used namespaces;

- **types** - describes the data types transmitted between the client and the server; uses the World Wide Web Consortim (W3C) XML Schema specification by default, and if the application uses only XML Schema build-in types, such as strings and integers, this element is not required;

- **message** - describes a one-way message, whether it is a single message request or a single message response; defines the name of the message and contains zero or more part elements, which can refer to message parameters or message return values;

- **portType** - shows the operations (functions) which are supported; combines multiple message elements to form a complete one-way or round-trip operation; portType can define multiple operations;

- **binding** - defines the full specifics of how the messages are transmitted on the wire; WSDL includes built-in extensions for defining SOAP services and SOAP-specific information therefore goes here;

- **service** - defines the address, where the services is located and can be invoked.

![Diagram of WSDL document structure](image)

**Figure 8: WSDL document structure [adopted from [Cera02]]**

### 2.5.2.3 Universal Description, Discovery and Integration (UDDI)

Universal Description, Discovery and Integration (UDDI) is a cross-industry effort, driven by all major platform and software providers, such as Dell, Fujitsu, HP, IBM, Intel, Microsoft, SAP, Oracle, Sun and Hitachi. UDDI is the name of a group of web-based registries, which expose information about an entity, be it a business entity or another entity, and its technical
interfaces, or APIs. These registries are run by multiple Operator Sites, which basic services can be accessed by anyone free of charge [UDCo02].

From a business developer’s point of view, UDDI is similar to an Internet search engine, as he can browse UDDI registries to view different businesses that expose Web Services, and the specifications of those services. Software developers can use the UDDI Programmers API to query the registry to discover services matching different criteria. Both business developers and software developers can also publish new business entities and services at the UDDI registry [Srir04]. The information that makes up a registration consists of five data structure types. These data structures, passed as input and output parameters of major API messages to and from the UDDI registries, are Business entity, Publisher assertion, Business service, Binding template, TModel.

2.5.3 Web Services Based Projects for M-Learning

Utilizing the web services architecture, discussed in section 2.5.1, an architecture for m-learning is proposed by Trifonova et al., where a ”mobile adapter” sits on top of an e-learning system and adapts the e-learning services for the specific mobile device [TrRo03, TrRo04]. The project aims at delivering an architecture, which is both general (i.e. able to provide all services, offered by an e-LMS, as well as services, which are new in the mobile domain) and generic (i.e. able to support mobile devices with different characteristics and be easily extensible for the new generation devices).

![Figure 9: Web Services based architecture for m-learning [TrRo03, TrRo04]](image)

The mobile adapter has three main modules. The ”Context Discovery” module handles the information about the user, the capabilities and limitations of his mobile device. Based on this information, the system checks the access privileges of the user and decides which services can be offered to him. When the user sends a request to the system, the ”Mobile Content Management and Presentation Adaptation” module is triggered, redesigning the data for the particular user and mobile device. The ”Packaging and Synchronization” module is
also triggered, selecting the most important resources and packages them, so that they can permanently fit into the mobile device’s memory for offline usage.

2.6 Mobile Web Services Provisioning

With the development of communication technologies and improvement of the characteristics of mobile devices, a natural direction in the domain of web services is exploring of the potential use of mobile devices as web service providers. This chapter reviews the mobile web services provisioning concept and the existing projects in the domain, one of which, the Mobile Host [Srir04], is used as a base for this master thesis.

2.6.1 Overview

In this thesis, the web services, which are provided by mobile devices, will be referred to as "mobile web services". Naturally, the mobile web services architecture contains the same components as the general web services architecture: service provider, service broker and service requester, and has the same pattern of communication, using SOAP, WSDL and UDDI, with the only difference that the web service provider is a mobile device. The basic architecture of mobile web service provisioning is shown in Figure 10.

![Figure 10: Mobile Web Services Architecture [SJPr06a]](image)

2.6.2 Mobile Web Services Provisioning Based Projects

Currently the mobile web service domain is quite a research, and there are not many projects available. Two of these projects: the Mobile Host [Srir04], which analyzes the feasibility of mobile web service provisioning, with SOAP communication over HTTP, and the mobile web service communication over UDP [GAWa06], are discussed in this section.

2.6.2.1 Mobile Host
The Mobile Host project [Srir04, SJPr06a, SJPr06b] takes on the challenge to implement the web service provider on the smart phone, thus to design the mobile terminal with the same general architecture as on any standard desktop system, even under the low-resource considerations of the Smart Phone. As with any web services based project, the standard WSDL is used to describe the services, and the standard UDDI registry is used for publishing and unpublishing the services.

The communication between the Mobile Host and the web service requester is done by using SOAP over HTTP. Alternate architectures for mobile web service provisioning are also possible with SOAP complaint proxy or gateway in between the Mobile Host and the web service requester: the communication between the client and the proxy is using SOAP and the communication between the proxy and the Mobile Host would be using a protocol, efficient for the data transport across the mobile networks. Many such proprietary protocols and implementations have evolved, such as WSOAP, gSOAP, eSOAP, etc. [SJPr06a]

The Mobile Host is developed as a web service handler, built on top of a normal web server. The web service requests, sent by HTTP tunneling, are diverted and handled by the Web Service handler. The core architecture of the Mobile Host is represented in Figure 11.

![Figure 11: Core architecture of the Mobile Host [Srir04, SJPr06a]](image)

At the HTTP interface, the Mobile Host listens for incoming HTTP GET/POST requests on a server socket. When a request is received, a socket for communication is created, and a new thread of execution is initiated by creating an instance of the Request Handler. The Request Handler extracts the incoming message from the input stream of the socket, and checks if the message is a web service request or not. If the message is a normal HTTP
request, the Request Handler processes the HTTP request just as a standard web server, and returns the response by writing to the output stream of the socket.

If the message is a web service request, sent over HTTP tunneling, it is diverted to be handled by the Web Service Handler. After the HTTP message body is read, the SOAP request is de-serialized to Java objects, using the SOAP processor. These Java objects are then passed to the Service Handler, which extracts the service details and invokes the respective service.

The web services, deployed on the Mobile Host, can access the local file system or any external devices, such as a GPS receiver, using Infrared, Bluetooth, etc. After the business logic of the service method is executed, the response is returned to the Request Handler. The Request Handler serializes the response and prepares the HTTP response message, which is then returned to the client by writing to the output stream of the socket.

Considering the low-resource constraints of Smart Phones, no deployment environment can be easily provided. Hence, all services have to be deployed at the installation of the Mobile Host. Alternatively, the Mobile Host can be configured to look for services at other locations apart from the main JAR location, where the services could then be deployed at runtime.

2.6.2.2 UDP Based Communication

As already discussed, apart from HTTP, SOAP messages can also be carried on top of any underlying protocols, such as TCP, UDP, BEEP, SMTP, etc. Gehlen et al. argue that the mobile web service communication over the User Datagram Protocol (UDP) performs faster than HTTP tunneling [GAWa06]. They suggest that UDP can be used to reliably transport messages by applying a simple Automatic Repeat Request (ARQ) protocol over UDP, and distinguishes the cases of reliable and unreliable binding. A system for an unreliable and reliable UDP SOAP binding for a mobile web service based middleware is also presented. The performance evaluation and comparison with HTTP tunneling is quite optimistic; however, it has been performed by two mobile phone emulators, running on desktop computers connected by Ethernet, and no measurements by using GPRS or UMTS networks, where the results would naturally turn out to be much worse, have been carried out.

2.6.3 Mobile Web Services Provisioning Based Projects for M-Learning

There are currently no mobile web service provisioning projects for m-learning, or at least none, which are known to the author. This is one of the reasons for exploring the potential use of mobile web service provisioning for the learning domain in this master thesis.

2.7 Java 2 Platform Micro Edition

Java 2 Platform Micro Edition (J2ME) is a collection of Java Application Programming Interfaces (APIs), providing a robust and flexible environment for applications, running on a broad range of resource-constrained devices, such as mobile phones, personal digital assistants, TV set-top boxes, etc. [Sun06a]. Unlike the desktop and server worlds, targeted
by the Java 2 Platform Standard Edition (J2SE) and the Java 2 Platform Enterprise Edition (J2EE), the micro-world includes a wide range of devices with different capabilities, and it is not possible to create a single software product which suits all of them [Topl02]. Therefore J2ME is not a single entity, but is divided into configurations, profiles and optional APIs. The Java 2 editions and their target markets are shown in Figure 12.

A configuration defines the minimum Java runtime environment suitable for a family of devices: the combination of a Java Virtual Machine (JVM) and a strict subset of the J2SE APIs. At present there are two main J2ME configurations: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). CLDC adresses the needs of low-level consumer electronic devices, such as cell phones and low-end PDAs with around 512KB of available memory. CDC is richer than CLDC, aimed at devices such as high-end PDAs with minimum 1MB RAM, 32-bit CPU and 2.5MB ROM. CLDC runs on top of Sun’s K Virtual Machine (KVM), and CDC runs on top of Sun’s C Virtual Machine (CVM).

A profile is based on a configuration and provides APIs related to the user interface, persistent storage, etc. The Mobile Information Device Profile (MIDP) is based on CLDC and provides in addition networking, local storage and user interface elements. The PDA Profile (PDAP) is similar to MIDP, but is aimed at PDAs with better screen capabilities and more memory than cell phones. The Foundation Profile is based on CDC and is a headless version of J2SE. The Personal Basis Profile is an extension of the Foundation Profile, including lightweight Graphical User Interface (GUI) support, which is a subset of the Abstract Window Toolkit (AWT). The Personal Profile is an extension of the Personal Basis Profile, having full AWT support. The J2ME configurations and profiles are presented in Figure 13.
When J2ME was first introduced, only one configuration, CLDC, and one profile, MIDP, had been defined as formal specifications through the Java Community Process (JCP). Today, there are nearly forty J2ME-related specifications at various stages in the JCP, and many of these specifications define optional packages instead of configurations or profiles. An optional package is a set of APIs, but unlike a profile, it does not define a complete application environment. Therefore it is always used in conjunction with a configuration or a profile. It extends the runtime environment to support device capabilities that are not universal enough to be defined as part of a profile or that need to be shared by different profiles [Sun02a]. Such optional packages are the PDA Optional Packages, defined by Java Specification Request (JSR) 75 and including the File Connection Optional Package and the Personal Information Management Optional Package, JSR-120 and JSR-205 Wireless Messaging API 1.1 and 2.0 respectively, JSR-135 Mobile Media API, JSR-82 Bluetooth API, JSR-172 Java ME Web Services 1.0, JSR-238 Mobile Internationalization API, JSR-179 Location API, etc.

The configuration, profile and optional APIs, ported to a device, is called a stack, and is the responsibility of the device manufacturer [LiKn05].

### 2.8 Summary

The latest trends and achievements in the domains of collaborative learning, mobile learning and mobile web service provisioning, which serve as a basis for this master thesis, are presented in section 2. First, collaborative learning is discussed, followed by an overview of e-learning and computer supported collaborative learning. Next, the concepts of Web 2.0 and the various types of social software are reviewed, and their potential use for supporting community building and collaborative learning is argued. Following is an overview of m-learning, and the various kinds of mobile devices and network technologies, which make m-learning possible. The requirements for any kind of mobile learning system are emphasized. Last, but not least, web services are explained, with a stress on mobile web service provisioning. Special attention is paid to the Mobile Host, a mobile web services provisioning project, which will be the foundation of the proposed architecture.
3 MobileHost CoLearn System Design

As discussed in Section 1, the MobileHost CoLearn system aims at supporting collaborative learning activities by means of web service provisioning from mobile phones communicating over GPRS. This section presents the design of the proposed architecture for collaborative mobile learning. The scenarios for potential use of the system are presented in Section 3.1, followed by the systematic discussion of the overall functional and non-functional requirements in Section 3.2. The system architecture is presented in Section 3.3, and the hierarchy of the developed modules for collaborative learning is discussed in Section 3.4. An elaboration of the requirements, which should be fulfilled by each of the individual modules of the MobileHost CoLearn system, is given in Section 3.5 - Section 3.7.

3.1 Scenarios

The next paragraphs exhibit some possible scenarios for collaborative learning by using the MobileHost CoLearn system.

**Expert finder scenario:** Anna has to create a UML model of a system she is going to implement. She has worked with UML before; however, as it is a very complex structure, she faces problems and needs some expert help. She knows a couple of people, such as Bob and Brandon, who are either likely to know something about her problem or are acquainted with other people with more extensive knowledge than themselves. She asks them to help her find the most knowledgeable person in the field by sending them the following request: if they do not know other people with a more extensive knowledge in UML than themselves, they have to return themselves as a result, stating their subjective assessment for their level of expertise in the field; if they know one or more people with better knowledge, they have to forward Anna’s request to them. As a result, Anna’s request is spread through a network of people, having a higher level of knowledge at each stage of the request path. In this way, a network of people, some of them not knowing each other at all, help Anna find the most suitable expert for her problem. In case Anna receives more than one answer, for example if she receives two responses, this means that there are two people, each of them evaluating himself as the biggest expert in the field. In the sample scenario, depicted in Figure 14, Anna receives responses from Susan and Melanie, containing their subjective rating for expertise in the field, 9 and 7 out of 10 respectively, and the path that the request has travelled before reaching them, namely Anna-Bob-Susan and Anna-Brandon-Ben-Melanie, complemented with the comments that each person on the path has made towards the next one. Finding more than one expert can be either due to the fact that these two people do not know each other, or because each of them evaluates himself to be a bigger expert than the other one. Facing the problem of choosing between the two potential experts, Anna can make a subjective decision, based either on their levels of expertise in the field or on the paths that the requests have passed before the responses are returned to her.

**Combined literature resource retrieval, photocasing, tagging and commenting scenario:** Melanie and George are architecture students, whose current task is to make a
report about the architecture of the town of Aachen. They decide to split the tasks in two parts, search for related literature resources, go around town individually and capture photos with their mobile phones. As there may be a large amount of literature, related to each of the buildings of interest, such as books, URLs, etc., Melanie and George decide to tag each literature resource with keywords with a corresponding low, medium or high relevance, so it is easy for them to quickly recall the most valuable resources later on and to organize similar ones into categories. Since they always want to be aware of the kind of architecture the other one has already documented, they retrieve each other’s tagged literature resources, and distribute the camera images at the moment they capture them through photocasting channels. They also comment on each other’s photos, and in this way, learn from each other more about the particular architectural buildings.

**Expertise podcasting scenario:** Cindy has just started working as a software engineer in a company. She loves running every day; however, she does not have much free time now, as she has to fulfill her own duties and at the same time be aware of the work of her teammates. John, one of her colleagues, likes speaking a lot, and he uses podcasts as a way of documenting his work progress, the difficulties he has faced and the milestones he has passed. As Cindy has to be up to date with John’s current work, a good solution for her is to subscribe to John’s podcasts. As soon as new podcasts are available, they are automatically downloaded to Cindy’s mobile phone. Next time she goes running, she already has the podcasts stored on her mobile device, and she can listen to them seamlessly while running. In this way she combines her leisure activities with her work duties.
3.2 System Requirements

In order for any software development project to be successful, the functional and non-functional requirements of the system should be clearly defined. The functional requirements of the mobile web services provisioning system for collaborative learning are elaborated in Section 3.2.1. The non-functional requirements of the system, i.e. the quality factors that are examined during the evaluation phase, are reviewed in Section 3.2.2.

3.2.1 Functional Requirements

As it is stressed at the beginning of Section 3, this thesis aims at delivering web services, provided by mobile phones, i.e. without any restriction to physical space. These web services should enhance the learning experience of the users by providing an environment, in which they can collaboratively discover and manage expert’s data, organize their resource data, such as literature resources and broadcast resources, and share their resources and knowledge with other learners, as well as gain new knowledge by asking them particular questions and receiving their feedback.

The functional requirements, related to the discovery and management of expert data, include:

- The system should enable the user to find experts in a specific field within a truly collaborative mobile environment. The users should be able both to search for experts themselves by creating and sending requests to their acquaintances, and to help other learners find experts by forwarding or answering their requests.

- The system should enable the user to manage the personal and expertise data of the potential experts that have been found, as well as his own data.

- As the number of experts discovered can be large, the system should provide search possibilities for experts by name and by expertise field. As expertise is something relevant, various levels of expertise in a field should be available.

- The system should enable the user to inquiry the experts about specific problems he has faced and receive their expert answers.

The functional requirements, regarding the discovery and management of literature resources, embrace:

- The system should enable the user to retrieve the tagged literature resources of the experts. Different types of resource retrieval should be supported, such as articles, inproceedings, proceedings, books, URLs, master and PhD theses, and unpublished resources. Only literature resources with a particular minimum level of relevance of the tags should be retrieved.
The system should enable the user to manage all types of resource data, he has gathered.

As the number of literature resources can be large, the system should provide search opportunities for each of the resource types. Three types of search should be supported: search by title, search by author and search by related tag.

The functional requirements, related to management and broadcasting of image and audio resources, include:

- The system should enable the user to retrieve the broadcast channels of the experts he has already found. Two types of broadcasts should be supported: photocasts and podcasts.
- The user should be able to subscribe to and later on unsubscribe from broadcast channels.
- The user should be able to specify his preferences for the broadcast content, such as whether to automatically receive new content as soon as it is available or only be notified about it, and in case the first option is accepted, whether to download all content or only content smaller than a specific size.
- In case the automatic download option is selected and the new broadcast complies with the user preferences regarding maximum content size, it should be automatically delivered to the mobile phone of the user when it becomes available.
- The user should be able to manage his own photocast and podcast channels, create new photocast episodes by capturing images with the integrated camera device or by browsing the file system of the mobile phone, create new podcast episodes by capturing sound with the integrated audio recording device or selecting an audio file from the file system of the mobile phone, and tag his own broadcast episodes.
- The user should be able to write comments to the broadcast episodes of others and to retrieve the feedback that other users have left to these broadcast episodes.

The main requirements of the MobileHost CoLearn system are generally depicted in the use case diagram in Figure 15, and the detailed requirements of the individual system modules are elaborated separately in Section 3.5 - Section 3.7.

3.2.2 Non-Functional Requirements

The non-functional requirements of the proposed system for collaborative mobile learning, also called quality of service requirements, are the ones, according to which the operation of the system rather than its behaviors are judged. These quality measures, which are defined in the ISO 9126 international standard for the evaluation of software [EAGL96], are discussed in the next paragraphs.
Reliability: Attributes, such as maturity, recoverability and fault tolerance, which ensure the capability of the software to maintain its level of performance under stated conditions for a certain period of time. Sensitive data should be protected and correctly recovered. Mobile learning introduces also another aspect of reliability- network reliability- which is especially important for synchronous learning environments.

Usability: A variety of issues, such as learnability, understandability and operability, which deal with the effort needed for use, and on the individual assessment of such use. Aesthetics of user interface, ease of understanding the system’s functioning and behaviour, and consistency are attributes of easy to learn systems with a rapid learning curve.

Efficiency: A set of factors, such as time behaviour and resource behaviour, which bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions. The system response time must be fast enough to satisfy the learner’s needs. Long waiting times result in user boredom, de-motivation and reduced interest, leading to unwillingness to use the system. Thus, a balance between quality and performance should be maintained, in order to ensure system efficiency.

Maintainability: Attributes, such as stability, analyzability, changeability and testability, which are concerned with the effort needed to make specified modifications. With the rapid technological changes and the continuously changing user requirements, the goal is to conceive the system in its best architectural and modular form, in order to assist easy future maintenance.

Portability: A set of factors, such as installability, replaceability and adaptability, which
ensure the ability of the software to be transferred from one environment to another. In the case of mobile learning, the ability of the application or service to be run by any mobile device is of a paramount importance.

3.3 System Architecture

The overall architecture of the MobileHost CoLearn system is represented in Figure 16. The main parts of the system are the three basic components of a web services based architecture, i.e. service provider, service requester and service registry. The communication between the Mobile Host and the web service requester is done by using SOAP over HTTP. As with any web services based project, the standard WSDL is used to describe the services, and the standard UDDI registry is used for publishing and unpublishing the services. Following the architecture of the Mobile Host [Srir04], the service provider is implemented on a mobile device.

![Figure 16: Overall Architecture of the MobileHost CoLearn System](image)

The Mobile Host listens for incoming HTTP GET/POST requests on a server socket. When a request is received, a socket for communication is created, and a new thread of execution is initiated by creating an instance of the Request Handler. The Request Handler extracts the incoming message from the input stream of the socket, and checks if the message is a web service request or not. If the message is a normal HTTP request, the Request Handler processes the HTTP request just as a standard web server, and returns the response by
writing to the output stream of the socket. If the message is a web service request, sent
over HTTP tunneling, it is diverted to be handled by the Web Service Handler, which
deserializes the request and extracts the service details. The control is then passed on to the
corresponding web service, belonging to one of the collaborative learning modules.

The web services, belonging to the collaborative learning modules, can access the file system,
personal information management databases, the camera and audio recording devices of the
mobile phone. After the business logic of the service method is executed, the response is
returned to the Request Handler. The Request Handler serializes the response and prepares
the HTTP response message, which is then returned to the client by writing to the output
stream of the socket.

3.4 Modules Hierarchy

The MobileHost CoLearn system consists of three main modules: Expertise Management
Module, Expertise Finder Module and Expertise Broadcasting Module. Each of these mod-
ules is composed of diverse sub-modules, each of which containing one or more web services,
as shown in Figure 17.

The Expertise Management Module contains three sub-modules: Expert Management Mod-
ule, Resource Management Module and Broadcast Management Module, under which the
Photocast and Podcast Management Modules are included. The Expert Management Mod-
ule enables the users to administrate the experts’ data, gathered by the Expert Finder
Module. Using the Resource Management Module, the learners can organize their literature
resources, such as articles, inproceedings, proceedings, books, master and PhD thesis, URLs
and unpublished resources, and tag them with keywords with an associated low, medium or
high relevance. The Broadcast Management Module is in charge of the different types of
broadcasts, created by the current user, such as photocasts and podcasts.

The Expertise Finder Module is composed of three sub-modules: Expert Finder Module,
Resource Finder Module and Expert Answer Module. The Expert Finder Module enables
users to search for experts in a specific field via the Expert Search Web Service, and receive
replies from experts, regarding their level of expertise in the field, via the Expert Rating
Web Service. A forwarding mechanism for expert finder requests has also been set up, so
that a true collaborative environment can be created. After having found an expert, users
can utilize the Resource Finder Module in order to retrieve the expert’s literature resources,
such as articles, books, etc., which have been tagged with specific keywords, via the Resource
Finder Web Service. As tagging is quite subjective, a scale of relevance of each tag to each
resource has been introduced. The Expert Answer Module gives the possibility for asking
experts specific questions and receiving their answers via the Problem and Expert Answer
Web Services. The Expertise Finder Module, its sub-modules and web services are described
in detail in Section 3.6.

The Expertise Broadcasting Module consists of two sub-modules: Photocasting Module and
Podcasting Module. These modules enable learners to subscribe to different types of broad-
casts that are delivered by other learners. The types of broadcasts considered are photocasts, dealt with in the Photocasting Module, and podcasts, dealt with in the Podcasting Module. After having retrieved the list of photocasting/podcasting channels, provided by a particular user, via the Available PhotoCh/PodCh Web Services, the learners can subscribe to any of them via the Subscribe PhotoCh/PodCh Web Services and afterwards automatically receive new episode content, as soon as it is available, via the Receive Photocast/Podcast Web Services. As mobile phones are resource limited devices, the learners can set preferences for the received content, such as whether the automatic download of content should be enabled or only a notification should be received, as well as the maximum content size to be downloaded.
in case the first option is accepted. As the system aims at providing a truly collaborative 
learning environment, users can not only receive learning content, but can also comment on 
the broadcast episodes and publish their comments on the broadcasting MobileHost CoLearn 
server via the Comment Photocast/Podcast Web Services, as well as retrieve the comments 
that other learners have left via the Retrieve PhotoComm/PodComm Web Services. The 
users can unsubscribe from any channel via the Unsubscribe PhotoCh/PodCh Web Services. 
The Expertise Broadcasting Module, its sub-modules and web services are elaborated in 
Section 3.7.

3.5 Expertise Management Module Design

The Expertise Management Module deals with the administration of the data of the current 
user, such as literature resources and broadcasts, and data regarding experts he has al-
ready found. The Expertise Management Module is composed of three sub-modules: Expert 
Management Module, which is presented in detail in Section 3.5.1, Resource Management 
Module, elaborated in Section 3.5.2, and Broadcast Management Module, including its two 
sub-modules- Photocast and Podcast Management Modules- discussed in Section 3.5.3.

3.5.1 Expert Management Module Design

The Expert Management Module is in charge of the data regarding experts in different fields, 
gathered by the current user. An overview of the module is presented in Section 3.5.1.1, 
followed by its functional requirements, discussed in Section 3.5.1.2.

3.5.1.1 Overview

The Expert Management Module is responsible for administering the data of the experts, 
either personally known to him or discovered via the Expert Finder Module, and for making 
the expert data available to the rest of the modules of the MobileHost CoLearn system. After 
receiving an Expert Rating answer from a potential expert, as discussed in Section 3.6.1, the 
user can make a subjective decision whether he would really like to consider the person as 
an expert or not. Setting the person as an expert results in permanently storing the expert 
data on the system so that it can be accessed even after restarting the server, and sharing 
the data between all the modules of the MobileHost CoLearn system. After having saved a 
person as an expert, the user can acquire the literature resources that the expert has tagged 
with particular keywords via the Resource Finder Module, inquire about problems he has 
faced via the Expert Answer Module, subscribe to the expert’s photocasting and podcasting 
channels as a result of which he will automatically receive new broadcast content as soon as 
it is available via the Photocasting and Podcasting Modules.

The most important personal information about an expert includes the expert’s names, 
phone number and IP address. The IP address is used when sending web service requests to 
the expert, but as in most of the cases it is dynamic, the phone number can be used as an
identifier for the expert- in case the provided IP address is unreachable at the moment, this implies that most probably it has changed, and a new mapping between a phone number and an IP address should be performed. The e-mail address and other contact data are not compulsory, but serving as an alternative for communication, for example if a request or a response needed to be typed on the phone keypad is too long and complex, the user might prefer to type it on a computer and send it by e-mail than type it on the phone and send it via a web service.

Alongside personal information, the expert’s data embraces information regarding his fields of expertise. As expertise evaluation is quite subjective and relative, a scale from 1 (the lowest) to 10 (the highest) for the expert’s level of expertise in a particular field has been introduced. Additionally a comment is associated with each expert, which is set by the current user, so that he can quickly recall who the expert is.

3.5.1.2 Use Case Diagram

The functional requirements of the Expert Management Module are depicted in the use case diagram in Figure 18.

![Use Case Diagram of the Expert Management Module](image)

Figure 18: Use Case Diagram of the Expert Management Module

The user dealing with the module is a learner, who is organizing the data about experts, which he has already gathered. New experts can be added to the system either by directly inputting the personal and expertise information of the person, or by importing experts’ data from XML files from the file system of the mobile device. Experts’ information can be permanently stored on the file system of the mobile device by exporting it to different XML files. The details of each expert can be viewed and afterwards edited or deleted. Regarding the management of the different expertise fields of each expert, new expertise fields can be added by inputting the field name and the corresponding level of expertise in the field, and afterwards edited or deleted. If the number of experts is too large, the possibility to search
for experts becomes crucial. Two types of expert search are supported: search by name and search by expertise field.

3.5.2 Resource Management Module Design

The Resource Management Module deals with the organization of the literature resource data of the current user. An overview of the module is presented in Section 3.5.2.1 and its functional requirements are discussed in Section 3.5.2.2.

3.5.2.1 Overview

The Resource Management Module is responsible for the administration of the data regarding the different types of literature resources, either directly known to the current user and stored by him on the system, or previously unknown to him and obtained via the Resource Finder Module. The management of 8 types of resources is currently supported by the module: articles, inproceedings, proceedings, books, URLs, master theses, PhD theses and unpublished resources. Each of them is characterized by the following attributes:

- article - title, author, journal, year, volume, pages;
- inproceedings - title, author, booktitle, year;
- proceedings - title, year;
- book - title, author, publisher, year, editor;
- URL - title, author, href;
- master thesis - title, author, school, year;
- PhD thesis - title, author, school, year;
- unpublished - title, author, note.

Related tags are associated with each resource for facilitating the identification of the resource. Each related tag contains the tag name, or keyword, and the relevance of the tag to the particular resource. As tagging is something subjective and tags can have different degrees of relevance to the resource, a three-level scale has been introduced: every tag can be evaluated as having a high, medium or low relevance to the resource.

3.5.2.2 Use Case Diagram

The user, dealing with the Resource Management Module, is a learner, who is organizing the data about literature resources, which he has previously gathered. The variety of functionalities that the system delivers to the user are presented in the form of a use case diagram in Figure 19.
New resources can be added to the system either by directly inputting their details, or by importing them from XML files, stored on the file system of the mobile device. The details of each resource, be it an article in a journal or magazine, an article in a proceedings, a conference proceedings, a book, an URL, a master or a PhD thesis, or an unpublished resource, can be viewed, and afterwards edited or deleted. Each of the different types of resources can be exported to an XML file on the file system of the mobile device. The related tags of each resource can be viewed, and afterwards managed by adding new related tags, editing the data of existing related tags, or deleting related tags. As the amount of resource data can be very large, a very important feature of the system is the possibility for literature search by title, author and related tags.

3.5.3 Broadcast (Photocast and Podcast) Management Module Design

The Broadcast Management Module is in charge of the various types of broadcasts. Two types of broadcasts are currently supported by the MobileHost CoLearn system: image files, also called photocasts, and audio files, also called podcasts. The two sub-modules of the Broadcast Management Module- Photocast Management Module and Podcast Management Module- are at hand for this purpose. As the functionality of these modules is quite similar, their design is presented side by side. An overview of the modules is given in Section 3.5.3.1, followed by their functional requirements in Section 3.5.3.2.
3.5.3.1 Overview

The Photocast and Podcast Management Modules are responsible for the administration of the two types of broadcasts: photocasts and podcasts. Correlated photocasts and podcasts can be organized into so-called channels, through which they can be distributed to other users via the Photocasting/Podcasting Modules.

The user can select if he would like to work with the Photocast/Podcast Management Module alone, or to combine it with the Photocasting/Podcasting Module. In the first case, the user would be able to manage his own image and audio resources, but would not be able to share them with other users and to receive their comments. If he would like to take advantage of the feedback of other learners, he can deploy the Photocast/Podcast Management Module in conjunction with the Photocasting/Podcasting Module, and in this way allow for subscribing to his photocasting/podcasting channels, distributing his photocast/podcast content and receiving other users’ feedback.

The most important data regarding channels, which can be managed via the Broadcast Management Module, includes the channel title, link, description, author and copyright. The episodes within a channel can be created either by adding metadata to existing image/audio resources, permanently stored on the file system of the mobile device, or by capturing new image/audio data with the integrated camera/audio recording device of the mobile phone and adding metadata to it. The most important metadata about an episode embraces its title, enclosure, including url, size and type of resource data, a globally unique identifier (GUID), date of publication, description, author, subtitle, and a list of related tags.

The shared data between the Photocast/Podcast Management Modules and the Photocasting/Podcasting Modules embraces the data about the channel subscribers, including their personal data and their preferences, such as whether to automatically receive new episode content as soon as it is available or only to be notified of its presence and regarding the maximum content size which should be delivered if the first option is checked, as well as the feedback, left from other learners, including their personal data, such as names, phone number, IP address, e-mail address and other contact data, the date of responding and their comment.

3.5.3.2 Use Case Diagram

The user of the Photocast and Podcast Management Modules is a learner, dealing with his own image and audio resources, which he would like to keep only for himself or would like to make available to other learners through the Photocasting and Podcasting Modules. The different types of user interaction with the two modules is depicted in the combined use case diagram in Figure 20. The green-coloured use cases are the ones which belong to the Photocast/Podcast Management Module alone and are completely independent of the status of the other modules, and the orange-coloured use cases are the ones which are available only if the Photocasting/Podcasting Module is deployed as well.

For both types of broadcasts, new channels can be created by inputting the channel metadata. The details of each channel can be viewed, and afterwards edited or deleted. In case the
Photocasting/Podcasting Module is deployed and there are users who have already subscribed to the channel, the list of subscribers and their personal and preference data can be also viewed.

The means of creating new episode content depend on the type of broadcast. In case of a photocast, new episodes can be created either by taking a snapshot with the integrated camera of the mobile phone or by browsing the file system for image files. In case of a podcast, new episodes can be created either by capturing audio with the integrated audio recording device of the mobile phone or by browsing the file system for audio files. In case the Photocasting/Podcasting Module is deployed and learners have left feedback on episodes, their comments can be also viewed.
3.6 Expertise Finder Module Design

The Expertise Finder Module enables users to search for experts in a specific field within a truly collaborative environment, and after having found an expert, to retrieve the expert’s tagged resources and to inquire about specific issues. The Expertise Finder Module consists of three sub-modules: Expert Finder Module, presented in detail in Section 3.6.1, Resource Finder Module, described in Section 3.6.2, and Expert Answer Module, elaborated in Section 3.6.3.

3.6.1 Expert Finder Module Design

The Expert Finder Module is used for finding experts in a specific field within a truly collaborative environment. The functionality of the module is accomplished by two web services: the Expert Search Web Service, which should be available on the mobile devices of the users, who will be asked regarding an expert search from an acquaintance—the requestor himself, or a friend of the requestor, or a friend of a friend of the requestor, etc., and the Expert Rating Web Service, which should be available on the mobile device of the requestor, so that he can receive the responses, which will be sent back from experts.

A user can create an expert finder request, stating that he is searching for an expert in a specific field, and send it to one or more of his acquaintances by calling the Expert Search Web Service, provided by the MobileHost CoLearn applications installed on their mobile devices. As the recipient receives the expert finder request, he can either reply to the requestor, forward the request, or do nothing about it, which is of course not recommended. In case the recipient of the request evaluates himself as being an expert in the field, he can respond back to the requestor, stating his level of expertise, by calling the Expert Rating Web Service, provided by the mobile device of the requestor. In case the recipient of the request does not think that he is an expert in the field, he can forward the request to his own acquaintances, calling the Expert Search Web Service, provided by their mobile devices.

If the requestor is assigned at level 0, all the acquaintances that he has sent his request to are assigned at level 1, all the acquaintances of the acquaintances of the requestor are assigned at level 2, and so on, a single path flow of the expert finder request is presented in Figure 21. Starting from the requestor, the request passes through zero or more intermediaries, before reaching the responder, who evaluates himself as being an expert in the field.

In a real situation, the expert finding flow is expected to be much more complex, in case the requestor sends his request to more than one of his acquaintances, and each of these acquaintances forwards the request to more than one of their own acquaintances, and so on. A sample complex flow, containing various paths through which the different instances of the request travel, is presented in Figure 22 (Note: for simplicity, the response arrows are not depicted in the figure, but the responders can be easily identified by their specific colouring). As it can be seen from Figure 22, there exists also another type of users, called dead-ends, who neither forward the request nor respond to it, and therefore are not of interest for the analysis of the request paths. In the given sample flow, the requestor receives replies from
six different people at five different levels, each of whom evaluates himself as being an expert to one extent or another. The requestor can then make a subjective decision who of the experts to contact, either by considering the expert’s level of expertise which is returned in the reply, or by looking at the intermediaries through whom the instances of the request have travelled before reaching the different experts, and selecting the path consisting of forwarders he knows and trusts more.

The functional requirements of the Expert Finder Module are discussed in Section 3.6.1.1. The Expert Search Web Service is introduced in Section 3.6.1.2, and the Expert Rating Web
Service is presented in Section 3.6.1.3.

3.6.1.1 Use Case Diagram

The main actors of the Expert Finder Module are the requestor, the forwarder and the responder. A requestor is a user, who in his search for an expert in a specific field creates an expert finder request and sends it to one or more of his acquaintances by calling the Expert Search Web Service, deployed on their mobile devices. A recipient of the request can be either a forwarder or a responder, or a dead-end who is of no interest and therefore not paid attention to. A forwarder is a user, who serves as an intermediary in the search for experts, by receiving expert finder requests from a requestor or from another forwarder, does not assess himself as being an expert in the field, and forwards the request to one or more of his acquaintances by calling the Expert Search Web Service, installed on his acquaintances’ mobile devices. A responder is a user, who receives expert finder requests from a requestor or from a forwarder, assesses himself as being an expert in the given field, and responds to the requestor stating his level of expertise in the field by calling the Expert Rating Web Service, deployed on the requestor’s mobile device.

The use case diagram in Figure 23 depicts the functional requirements of the Expert Finder Module, showing the different scenarios of interaction between an actor and a use case. In his search for an expert, the requestor can create new expert finder requests, import and export requests, view the details of a request, edit or delete a request, send his request to other users—either by inputting their IP address or by selecting them from the contact list of the mobile device—and afterwards view the list of users he has sent his request to, receive back replies from potential experts and afterwards view the replies concerning a request, as well as save a responder as an expert. The forwarder and the responder can receive other users’ requests, import and export them, view the details of others’ requests, delete them, reply to them and afterwards view their reply, forward the request to other people and afterwards view the list of people they have forwarded the request to. A thorough explanation of how these use cases are accomplished by means of web service provisioning is elaborated in Section 3.6.1.2 and Section 3.6.1.3.

3.6.1.2 Expert Search Web Service

The Expert Search Web Service, in conjunction with the Expert Rating Web Service, composes the web services suite of the Expert Finder Module. The Expert Search Web Service is responsible for initial sending or forwarding of an expert finder request to other people, and should be provided by the mobile device of the recipient of the request— a potential forwarder or a potential responder.

The format of the SOAP request message depends on the path, that the request has travelled so far. If this is a request, sent from the requestor at level 0 to users at level 1, the SOAP request message contains information about the requestor, such as names, phone number, IP address, e-mail address, and other contact data, as well as information about the request itself, such as title, description, start date and expiry date. If this is a request, sent from a
forwarder at level \( k \) to a user at level \( k+1 \), the SOAP request message consists of the original request message, complemented with information about the path, which the request has passed so far, i.e. information about each of the forwarders from level 1 to level \( k \), including their names, phone numbers and IP addresses, as well as their comments and the dates of forwarding. The path information is used from one side for preventing the forwarding of requests to people, through whom the request has already passed, in this way preventing the cycling of the request through the same people again and again; and from the other side, it is used for facilitating the requestor in his subjective decision who of the experts to contact, in case two or more replies from different experts are received, as he can check who the users at the intermediaries are and select the expert, who is a result of the path that he trusts more.

A sample Expert Search SOAP request message from a forwarder at level 2, who does not assess himself as being an expert in the given field, to a user at level 3, who is a potential expert or forwarder, is given in Figure 61 in Appendix A. As a forwarder initiates the request, the message consists of the message that he has received, complemented with his own
forwarder data, his comment and date of forwarding, i.e. altogether the message is composed of the original request message and the information about the two current forwards.

A typical Expert Search SOAP response message, no matter from which level $k+1$ to level $k$ it has been sent, to an Expert Search SOAP request, sent from level $k$ to level $k+1$, is given in Figure 62 in Appendix A. Such a response message contains status information regarding the processing of the request at the recipient side, in this case a message indicating successful receiving and processing of the SOAP request.

### 3.6.1.3 Expert Rating Web Service


If the recipient of the expert finder request assesses himself as being an expert in the given field, he can reply to the requestor by calling the Expert Rating Web Service, provided by the requestor’s mobile device. The Expert Rating request message that he sends consists of the Expert Search request message that he has received, complemented with his own responder data, such as names, phone number, IP address, e-mail address and other contact data, his own rating for his level of expertise in the field, his comment and the date of response.

Altogether, the Expert Rating request message from an expert at level $n$ to the requestor at level 0 consists of:

- the original request message, i.e. information about the requestor at level 0, request title, description, start date and expiry date;
- information about the path, which the request has passed before reaching the expert, i.e. information about each of the forwarders from level 1 to level $n-1$, including their names, phone numbers and IP addresses, as well as their comments and the dates of forwarding;
- the response of the expert at level $n$, including his own responder data, such as names, phone number, IP address, e-mail address and other contact data, as well as the responder’s rating for his level of knowledge in the field, his comment and the date of response.

A sample Expert Rating SOAP request message from an expert at level 3 to the requestor at level 0 is given in Figure 24. As this is a request that an expert at level 3 initiates, the request message is composed of the original request message, complemented with the path that the request has travelled before reaching the expert, consisting of two forwards, and the actual expert response.

A typical Expert Rating SOAP response message, sent from the requestor at level 0 to an expert at an arbitrary level, contains status information regarding the successful or unsuccessful processing of the request on the mobile device of the user, who is searching for an
expert. The format of the Expert Rating SOAP response message is almost the same as the one of the Expert Search SOAP response message, given in Figure 62 in Appendix A.
3.6.2 Resource Finder Module Design

The Resource Finder Module enables learners to retrieve the tagged literature resources of an expert, they have already found through the Expert Finder Module and saved in the Expert Management Module. Eight types of resources are currently supported for retrieval by the Resource Finder Module: articles, inproceedings, proceedings, books, URLs, master and PhD theses, and unpublished resources. Each resource is associated with a list of tags with a particular relevance to the resource. As tagging is quite subjective, three levels of tag relevance have been introduced: low, medium and high. Each of the retrieved resources can be saved and afterwards managed via the Resource Management Module.

The functionality of the module is fulfilled by the Resource Finder Web Service, which should be available on the mobile device of the expert and is responsible for handling Resource Finder requests, sent by other learners. The use case diagram of the Resource Finder Module is presented in Section 3.6.2.1 and the Resource Finder Web Service is described in Section 3.6.2.2.

3.6.2.1 Use Case Diagram

The different scenarios of user interaction with the Resource Finder Module are presented in Figure 25.

![Figure 25: Use Case Diagram of the Resource Finder Module](image)

The main actor of the module is a learner, who is searching for literature resources on a particular topic. The learner can create new requests and afterwards view them. He can send requests to experts and view the automatic replies received from the instances of the Resource Finder Web Service, deployed on the mobile devices of the experts. For each of the expert responses the learner can view a summary of the retrieved resources and afterwards view the details of every resource, be it an article, inproceedings, proceedings, book, URL, master or PhD thesis, or an unpublished resource. If the learner finds a resource to be of
interest to him, he can save the resource and afterwards manage its data in the Resource Management Module. If the learner has found the needed resources on the topic, he can delete the request.

### 3.6.2.2 Resource Finder Web Service

The Resource Finder Web Service should be available on the mobile devices of the experts and is responsible for receiving Resource Finder SOAP requests from other learners, accessing the expert’s literature databases in the Resource Management Module, and retrieving the ones, which have been tagged with a particular tag with a specified minimum tag relevance.

A Resource Finder SOAP request contains the list of literature types, the learner is interested in, and the tag represented by its tag name and minimum tag relevance. When the Resource Finder Web Service receives a request, the resources of the expert are accessed, and if a resource is of one of the requested literature types and it has been tagged by the expert with the particular tag name and has at least the requested tag relevance as a minimum, the resource is returned in the Resource Finder response. A sample Resource Finder SOAP request message, requesting resources from two resource types- books and inproceedings- with a tag web services with at least a medium relevance to the resource, is given in Figure 26.

```
POST / HTTP/1.1
SOAPAction: resourceFinder
Content-Type: text/xml
Content-Length: 609
User-Agent: KSOAP/2.0
Host: 123.123.123.123:8080

<?xml version='1.0' encoding='UTF-8'?>
 xmlns:SOAP-ENC="http://www.w3.org/2001/12/soap-encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <SOAP-ENV:Body>
  <ns1:resourceFinder xmlns:sn="http://www.mobilehost.com/"
   xmlns:ns="http://www.mobilehost.com/">
   <resourceTypes>
    <resourceType xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="xsd:anyType[2]">
     <resourceType xsi:type="xsd:string">books</resourceType>
     <resourceType xsi:type="xsd:string">inproceedings</resourceType>
    </resourceType>
   </resourceTypes>
   <relatedTag xsi:type="xsd:string">
    <tag xsi:type="xsd:string">web services</tag>
    <minRelevance xsi:type="xsd:string">medium</minRelevance>
   </relatedTag>
  </ns1:resourceFinder>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 26: Resource Finder SOAP Request Message

A sample Resource Finder SOAP response message, containing the resources matching the given criteria, is shown in Figure 27.
Figure 27: Resource Finder SOAP Response Message
3.6.3 Expert Answer Module Design

The Expert Answer Module is used for finding solutions to encountered problems by inquiring experts and receiving their expert answers. The functional requirements of the Expert Answer Module are presented in Section 3.6.3.1, and its two web services- the Problem Web Service and the Expert Answer Web Service- are discussed in Section 3.6.3.2 and Section 3.6.3.3 respectively.

3.6.3.1 Use Case Diagram

The main actors of the Expert Answer Module are the problem requestor and the problem responder. A problem requestor is a user, who has faced a problem and sends problem requests to experts, who he has already found through the Expert Finder Module, by calling the instances of the Problem Web Service, provided by the mobile devices of the experts. A problem responder is an expert, who has been approached for help by a problem requestor and responds back to him stating his view for a solution to the problem, by calling the Expert Answer Web Service, deployed on the mobile device of the problem requestor.

![Use Case Diagram of the Expert Answer Module](image)

Figure 28: Use Case Diagram of the Expert Answer Module

The use case diagram, which designates the functional requirements of the Expert Answer Module, is presented in Figure 28. In his search for a problem solution, the problem requestor can create new problem requests, view the details of a problem, send his problem to experts and afterwards view the list of experts he has sent the request to, receive replies from the experts and afterwards view the details of their answers, or delete the problem after a solution has been found and it is no longer a problem. The problem responder can receive other users’ problem requests, view the details of these requests, reply to them and afterwards view his reply, or delete them. A thorough explanation of how these interactions between the different
3.6.3.2 Problem Web Service

The Problem Web Service, together with the Expert Answer Web Service, builds the web services suite of the Expert Answer Module. The Problem Web Service has to be deployed on the mobile devices of the users, who evaluate themselves as being experts in some field, and are willing to help other learners who face problems in this field by answering their questions.

As a user encounters a problem, he can send problem requests to one or more of the other users, whom he has already found and marked as being experts in the field through the Expert Finder Module. The Problem SOAP request message contains information about the requestor, such as names, phone number, IP address, e-mail address and other contact data, and information about the problem itself, such as title and description. The user has to input only the problem title and description, and the data about the requestor is automatically generated based on the current user of the MobileHost CoLearn application. A sample SOAP request message to the Problem Web Service is given in Figure 63 in Appendix A.

The SOAP response message to the problem request contains notification about the status of the processing of the request on the expert side.

3.6.3.3 Expert Answer Web Service

The Expert Answer Web Service is the second service, which completes the web services suite of the Expert Answer Module. It has to be provided by the mobile device of the problem requestor, so that he can receive back the answers from the experts.

After having received a problem request and if able to help, the expert can send his expert answer to the problem requestor by calling his Expert Answer Web Service. The SOAP request message contains the original problem information, which has been received through the Problem Web Service, such as requestor data, title and description, as well as in addition the responder data, such as names, phone number, IP address, e-mail address and other contact data, and his expert answer regarding the problem. A sample SOAP request message of the Expert Answer Web Service is given in Figure 29.

The SOAP response message to the expert answer request contains information about the successful or unsuccessful processing of the request on the mobile device of the person, who has initiated the problem request.
3.7 Expertise Broadcasting (Photocasting and Podcasting) Module Design

The Expertise Broadcasting Module works in conjunction with the Broadcast Management Module, and is in charge of the distribution of the image and audio files of the current user of the MobileHost CoLearn system to other learners. If the Broadcast Management
Module is not deployed, then the learner would be able to retrieve others’ broadcasts, but would not be able to make his own image and audio data accessible to other learners. If the Broadcast Management Module is also deployed, then the collaboration would be double sided, as the learner would be both a provider and a receiver of broadcast content. The two sub-modules of the Expertise Finder Module, namely the Photocasting Module and the Podcasting Module, are discussed side by side, as their functionality is almost the same and their web services are analogical.

The functional requirements of the Photocasting and Podcasting Modules are presented in Section 3.7.1, and the web services suite of the module is discussed in Section 3.7.2 - Section 3.7.7.

### 3.7.1 Use Case Diagram

The types of users, who interact with the Photocasting Module, are the photocasting provider and the photocasting receiver. A photocasting provider is a user, who manages his own photo content via the Photocast Management Module, and makes it available to other learners via the Photocasting Module. A photocasting receiver is a user, who is interested in the photocasts of another learner, subscribes to his photocasting channels, and automatically receives new content as soon as it is available. The functional requirements of the Photocasting Module are depicted in Figure 30. The functional requirements of the Podcasting Module are completely the same.

![Use Case Diagram of the Photocasting Module](image)

Figure 30: Use Case Diagram of the Photocasting Module

As already mentioned, the Photocasting Module is closely related to the Photocast Management Module. At the time when a learner creates a new photocast episode in Photocast
Management Module, either by capturing an image with the camera device of the mobile phone or by browsing the file system for an image file, the photocast episode is published in the respective photocasting channel, and in this way distributed to other learners, who have subscribed to the channel. If there are subscribers to the particular photocasing channel, the photocasting provider can see their personal data, and if not willing to make his content accessible to a particular subscriber, he can remove him from the subscribers list. He can also receive feedback from other learners, regarding a specific photocast episode, and view its content, including the personal details of the responder, his comment and the date of commenting.

A learner, who would like to subscribe to photocasting channels of other learners, can first retrieve the list of available channels, then subscribe to any of them and automatically receive photocast episodes as soon as they become available. He can also comment on episodes and retrieve the comments, which other learners have left. In case there are many photocasts, stored on the system, the user can perform search by tag and author. If he is not satisfied with the photocasting content, he can later on unsubscribe from the channel.

3.7.2 Available Photocasting/Podcasting Channels Web Services

The Available Photocasting/Podcasting Channels Web Services are responsible for retrieving the list of channels, which a particular user has made available to other learners. As the structure of the SOAP request and response messages of the two web services is the same, only the format of the Available Photocasting Web Service is discussed.

The Available Photocasting Channels SOAP request message contains no parameters, and simply demands for the photocasting channels, provided by a particular learner. The response, which is automatically sent back, contains the list of channels, including their most important data, such as title, link, description, author and copyright. A sample Available Photocasting Channels SOAP response message, containing the details of channels 'Red Book Species' and 'Aachen Architecture', is shown in Figure 31.

3.7.3 Subscribe to Photocasting/Podcasting Channel Web Services

The Subscribe to Photocasting/Podcasting Channel Web Services enable the learner to subscribe to a particular photocasting/podcasting channel, after having retrieved its details via the Available Photocasting/Podcasting Channels Web Services. As the structure of the SOAP request and response messages of the two web services is the same, only the one related to photocasting is presented.

The Subscribe to Photocasting Channel SOAP request message contains an identifier for the channel of interest, which in this case is the unique channel link attribute, the personal data of the subscriber, as well as his preferences: whether to automatically receive new content as soon as it is available or only be notified of its availability, and the maximum size of the content to be downloaded in the case the first option is accepted. A sample Subscribe to Photocasting Channel SOAP request message, demanding a subscription to the previously
Figure 31: Available Photocasting Channels SOAP Response Message

retrieved 'Red Book Species' channel, is shown in Figure 32. The Subscribe to Photocasting Channel SOAP response message contains notification about the successful or unsuccessful subscription of the learner to the particular channel.

3.7.4 Receive Photocast/Podcast Web Services

The Receive Photocast/Podcast Web Services are in charge of the automatic receipt of photocasting/podcasting episodes, sent automatically by the MobileHost CoLearn system of a learner, who creates a new photocasting/podcasting episode within one of his channels, to a learner, who has already subscribed to the specific channel. As the structure of the SOAP
3.7.5 Comment Photocast/Podcast Web Services

The Comment Photocast/Podcast Web Services enable the learner to leave feedback to a photocast/podcast episode, he has already received via the Receive Photocast/Podcast Web Services. As the structure of the SOAP request and response messages of the two web services is the same, only the format of the Receive Photocast Web Service is presented here.

The Receive Photocast SOAP request message contains all the data, related to the photocasting episode, such as the identifier of the channel it belongs to and the permanent identifier of the episode itself, the episode title, URL, size and type of image, publication date, description, author, subtitle, and a list of tags, specified by the photocast provider, as well as the data of the image itself, which is transferred over the air as a base64 encoded string. A sample Receive Photocast SOAP request message, dealing with an episode 'Pink Pigeon in Mauritius', belonging to the already subscribed to channel 'Red Book Species', is given in Figure 33. The Receive Photocast SOAP response message contains the status of the request processing on the mobile phone of the learner, who accepts the photocast.
services is the same, only the one regarding photocasting is reviewed here.

The Comment Photocast SOAP request message contains the global identifier of the photocast episode, to which the learner would like to leave a comment, and his feedback, which embraces the personal information of the learner, his comment and the date of commenting. A sample Comment Photocast SOAP request message, leaving feedback on the ‘Pink Pigeon in Mauritius’ episode, which has been previously received, is shown in Figure 34. The Comment Photocast SOAP response message contains a notification about the successful or unsuccessful receipt of the comment on the side of the learner, who provides the photocast.
### 3.7.6 Retrieve Photocast/Podcast Comments Web Services

The Retrieve Photocast/Podcast Comments Web Services allow the learner to retrieve the feedback, that other learners have left, to a photocasting/podcasting episode, which he has already received from another learner. As the structure of the SOAP request and response messages of the two web services is the same, only the Retrieve Photocast Comments Web Service is discussed here.

The Retrieve Photocast Comments SOAP request message contains the global identifier of the photocast episode, which feedback the learner would like to retrieve. The Retrieve Photocast Comments SOAP response message contains the list of feedback from other learners, including their personal information, comments and the dates of commenting. A sample Retrieve Photocast Comments SOAP response message, demanding for the feedback left to the ‘Pink Pigeon in Mauritius’ episode, is given in Figure 35. The response contains the list of all feedback to the episode, which in this case is only one.

![Figure 34: Comment Photocast SOAP Request Message](image)
3.7.7 Unsubscribe from Photocasting/Podcasting Channel Web Services

The Unsubscribe from Photocasting/Podcasting Channel Web Services allow the learner to unsubscribe from a channel, in case he is not satisfied with its content. As the structure of the SOAP request and response messages of the two web services is the same, only the one related to photocasting is presented.

The Unsubscribe from Photocasting Channel SOAP request message contains the phone number of the learner who would like to be unsubscribed, which acts as an identifier of the person, and an identifier for the channel, which is the unique channel link attribute. The Unsubscribe from Photocasting Channel SOAP response message conveys information regarding the status of the unsubscription process.
3.8 Summary

This section elaborated on the design of the proposed architecture for collaborative mobile learning. The scenarios for potential use of the system were presented, followed by the systematic discussion of the overall functional and non-functional requirements that should be fulfilled. The overall system architecture was discussed, and the hierarchy of the collaborative learning modules was reviewed. Last, but not least, the requirements of each of the individual modules of the MobileHost CoLearn system were worked out in detail.
4 MobileHost CoLearn System Implementation

This section elaborates on the implementation of the MobileHost CoLearn system. It starts with a presentation of the technologies and tools being used in section 4.1. The overall implementation of the system and its three main modules- Expertise Management, Expertise Finder and Expertise Broadcasting Modules- is discussed in Section 4.2, without paying attention to the particularities of the individual sub-modules, which are presented separately: the detailed implementation of the Expert Management Module is given in Section 4.3, the Resource Management Module in Section 4.4, the Photocast and the Podcast Management Modules in Section 4.5, the Expert Finder Module in Section 4.6, the Resource Finder Module- in Section 4.7, the Expert Answer Module- in Section 4.8, and the Photocasting and the Podcasting Modules- in Section 4.9.

4.1 Technologies and Tools Used

This section reviews the technologies and tools, which have been used for the implementation of the MobileHost CoLearn system, such as the Java 2 Platform Micro Edition (J2ME), Eclipse SDK and EclipseME plug-in, NetBeans IDE and NetBeans Mobility Pack, Sun WTK and Sony Ericsson SDK, kSOAP2 and kXML2.

4.1.1 J2ME

As this master thesis concentrates on using mobile phones as web service providers in the mobile learning domain, J2ME is the most proper environment for developing the proposed architecture, as it provides a robust and flexible environment for applications, running on a broad range of resource-constrained devices, as discussed in Section 2.7. PersonalJava- the technology originally used for implementing the Mobile Host- is another possibility; however, as PersonalJava will soon begin the Sun End of Life process, it is advisable that Personal Java projects move to J2ME as well [Sun06b].

4.1.1.1 CLDC 1.1 / MIDP 2.0

As the thesis aims at delivering a collaborative learning environment for mobile phones and smart phones, the most suitable J2ME environment is the combination of the Connected Limited Device Configuration (CLDC) with the Mobile Information Device Profile (MIDP), as depicted in Figure 36. As discussed in Section 2.7, the typical requirements for CLDC are 16-bit to 32-bit CPU and 32MB to 512MB RAM. MIDP supports an LCD orientated GUI API. Almost all new mobile phones come with a MIDP implementation, and it is now the de facto standard for downloadable cell phone games and applications. More precisely, JSR-139 CLDC 1.1 and JSR-118 MIDP 2.0 have been used for the implementation of the MobileHost CoLearn system.
4.1.1.2 Required Optional Packages

The requirement for CLDC 1.1 / MIDP 2.0 support is not enough for the proper running of the MobileHost CoLearn application on the mobile phone. The application extensively accesses the file system, the audio recording and camera devices, utilizes wireless messaging, and accesses the personal information management databases of the phone. All these operations are neither part of the CLDC specification nor of the MIDP specification, but are at hand via optional packages, which should be supported by the phone in order for the complete functionality of the MobileHost CoLearn system to be available to the user.

**JSR-75 File Connection Optional Package (FCOP):** JSR-75 refers to the so-called PDA Optional Packages, the first one of which is FCOP. The Generic Connection Framework (GCF) defined by the CLDC provides the basic scaffolding for file input/output, but it is up to specific implementations to expose this capability to applications. From one side this is good, because it allows the CLDC to be ported to devices without a file system, for example devices running the Palm operating system which do not support file systems in main memory but only on memory expansion cards. If a file system is supported, however, the standard way to have read and write access to it is through FCOP. The MobileHost CoLearn system extensively accesses the file system of the mobile device for reading and writing data regarding the registered users, the experts, the literature resources, the photo-cast and podcast resources, etc. That is why the FCOP support on the mobile phone is so essential for running the application.

**JSR-75 Personal Information Management Optional Package (PIMOP):** PIMOP is the second so-called PDA optional package, specified by JSR-75. Most mobile phones manufactured today have the ability to manage information of importance to the end user, such as a calendar of appointments, a file of contacts, or a list of things to do. This ability is referred to as personal information management, or PIM for short. The PIM data is stored persistently on the device and is normally accessed by the user via a couple of special-purpose applications. Such data includes the contact database, which follows the vCard 2.1 specification of the Internet Engineering Task Force (IETF), the event and to-do databases, which adhere to the vCalendar 1.0 specification of the Internet Mail Consortium (IMC). The
data from the contact, event and to-do databases is made available to J2ME applications via
the PIMOP. The MobileHost CoLearn system needs PIMOP support in order to send web
service requests to people from the contact list, but its availability is not crucial: if PIMOP
is not supported by the mobile phone, then only the possibility for sending expert finder web
service requests to people from the contact list would be disabled.

**JSR-135 Mobile Media API (MMAPI)**: MMAPI provides for the capturing and viewing
of multimedia content. If the mobile device supports the playback of specific audio or
video formats, or the recording of audio or video content, these capabilities are generally
exposed to J2ME applications via the MMAPI optional package. From the image formats,
such as the Portable Network Graphics (PNG), Joint Photographic Experts Group format
(JPEG) and Graphics Interchange Format (GIF), the video file formats, such as Moving
Picture Experts Group (MPEG), and the audio file formats such as Waveform Audio (WAV),
MPEG-1 Audio Layer 3 (MP3), Musical Instrument Digital Interface (MIDI), etc., all MIDP
2.0 implementations support mandatory only PNG, MPEG and WAV. The MobileHost
CoLearn system requires MMAPI support in order to access the integrated camera and
audio devices of the mobile phone in order to capture and playback photocast and podcast
content. If MMAPI is not supported by the mobile phone, the Photocast and Podcast
Management, Photocasting and Podcasting Modules would be disabled, but the rest of the
modules would work without problems.

**JSR-120 Wireless Messaging API 1.1 (WMA)**: WMA allows applications to send and
receive wireless messages. WMA is focuses on two protocols: Short Message Service (SMS)
and Cell Broadcast Service (CBS). SMS allows for sending messages, generally less than 160
characters and containing only text, between devices. The WMA provides full support of
two-way messaging, allowing an application to act both as a client in order to send messages
and as a server in order to receive messages. CBS provides the ability to send messages as
large as 1395 bytes in ether an ASCII or binary format to many devices within a geographical
area. These messages are generally repeated at periodic intervals to the devices registered
in the cell, so that devices just coming into range, or being turned on in the cell, have the
opportunity to receive the message, even though the first transmission was missed [Hemp04].
The MobileHost CoLearn system requires WMA support in order to send notifications to
users as soon as a new web service request arrives, but this is not crucial. If the mobile device
does not support WMA, then the user would not receive notifications, but would still receive
the requests and be able to deal with them at the time when he checks them manually.

### 4.1.2 Eclipse SDK and EclipseME Plug-In

Eclipse SDK 3.2 is the Software Development Kit (SDK), which has has been used for imple-
menting the proposed architecture for collaborative mobile learning. It is a free open-source
SDK, which provides software developers with all the tools needed to create professional
cross-platform desktop, web and enterprise applications. EclipseME is an Eclipse plugin to
support the development of mobile applications. EclipseME is in charge of the connection
of the Eclipse development environment with wireless toolkits, so that the user can focus
on the development of the application itself, rather than worrying about the special needs of J2ME development. Through these wireless toolkits EclipseME provides CLDC/MIDP support to the mobile applications, developed under Eclipse. As the proposed architecture for collaborative mobile learning is targeted at mobile phones, the combination of Eclipse and EclipseME is a very suitable choice for an implementation platform. NetBeans IDE and its NetBeans Mobility Pack is another possibility, which however has some disadvantages, as presented in the next section.

4.1.3 NetBeans IDE and NetBeans Mobility Pack

NetBeans IDE 5.5 is an Integrated Development Environment (IDE), which has been used for the design of the screen flow of the MobileHost CoLearn application. The NetBeans Mobility Pack, which is a NetBeans add-on, supports for creation, testing and debugging of applications, which will run on J2ME enabled mobile devices. It supports CLDC 1.0 and 1.1 and MIDP 1.0 and 2.0, SVG graphics (JSR-226), JUnit testing, MIDlet signing, certificate management, integrated over-the-air (OTA) emulation, push registry emulation, WMA and MMAP, etc. [SuCo06a, SuCo06b]. As it turns out however, the biggest disadvantage of NetBeans is that the automatically generated code, based on the manually designed screen flow, is not clear and is organized in only one MIDlet class. That is why NetBeans and NetBeans Mobility Pack have been used only for designing the screen flow of the MobileHost CoLearn application, but the actual implementation has been performed under Eclipse.

4.1.4 Sun Java Wireless Toolkit for CLDC and Sony Ericsson SDK

During the early stages of the master thesis, the Sun Java Wireless Toolkit 2.5 for CLDC has been used in conjunction with Eclipse and EclipseME for the mobile phone emulation. Sun WTK supports MIDP 2.0, CLDC 1.1, WMA 2.0, MMAP 1.1, Web Services 1.0, FCOP and PIMOP, Bluetooth and OBEX APIs, 3D Graphics, Security and Trust Services API, Location API, etc. It provides four mobile phone skins and works very good, but due to some instabilities encountered while accessing the file system, a shift to the Sony Ericsson SDK has been made.

Sony Ericsson SDK 2.2.4 also supports the above mentioned APIs, and has emulator skins for almost all Sony Ericsson phones. It has turned out to be an incredibly useful tool for figuring out which optional packages are supported on which phone models and for checking which buttons are mapped to which soft keys on the particular mobile device. Unlike Sun WTK, it has never caused problems while accessing the file system. It has been very comfortable for testing the user interface for three of the mobile phones, used for real evaluation of the MobileHost CoLearn application- Sony Ericsson P800, P910 and W810i, but as it is not universal, Sun WTK has been used for fully verifying the functionality of the system. Moreover, an error which was faced with a Nokia N70 phone could not be detected by the Sony Ericsson SDK, but was caught by the Sun WTK. All in all, both toolkits have been very useful at some point of time for the emulation of the application.
4.1.5 kSOAP2 and kXML2

As SOAP is a key ingredient of any web services based architecture, a parser needs to be present on the provider device. The common XML and SOAP packages, such as Xerces (for XML) and Axis (for SOAP) are not only far too large and resource-intensive, but also depend on features of the Java runtime, which simply do not exist on resource constraint devices, such as smart phones [McHu03]. In order to accommodate the limited footprint of the KVM- a low-memory virtual machine running on microdevices- a lightweight parser is needed. Such a parser is kSOAP2, based on kXML2, which has been utilized in this master thesis.

kXML, with current version kXML2, is a small open source XML pull parser, specially designed for constrained environments [Bala03]. kSOAP, with current version kSOAP2, is an open source project for SOAP parsing on the J2ME/MIDP platforms, based on kXML [Yuan02]. kSOAP has special type mapping and marshalling mechanisms and provides programming transparency between a Java program and a SOAP message. A programmer just feeds Java objects into the SOAP writer, sends the message, waits for the server response, and then reads Java objects directly from the SOAP parser [Srir04]. Both kSOAP2 and kXML2 are thin, easy to use, and well documented [Enhy06], and utilizing them for the SOAP message parsing for smart phones is currently the best possible solution.

4.2 Overall MobileHost CoLearn Implementation

As the MobileHost CoLearn system is quite complex, the presentation of the system classes is divided into two batches: classes, related to the graphical user interface (GUI), and classes, not related to the GUI. The main non-GUI classes of the MobileHost CoLearn system are discussed in Section 4.2.1, and the main GUI classes are presented in Section 4.2.2.

4.2.1 Overall Class Diagram

The overall class diagram of the MobileHost CoLearn system is presented in Figure 37. The core class of the system is the MobCoLearn class. Class User represents a registered user, containing data about the user’s names, phone number, e-mail address, other contact data, and copyright. Class Server denotes the MobileHost server, which listens for incoming requests through a server socket connection on a specific port. Whenever a new request arrives, the server creates a new instance of the request handler RequestHandler, which checks if the request is a normal HTTP request or a web service request, and handles it in case of an HTTP request, or passes it to the web service handler WSHandler in case of a web service request.

The main modules of the application- Expertise Management, Expertise Finder and Expertise Broadcasting Modules- are represented by classes EMgmtModule, EFinderModule and ECastingModule respectively. The Expertise Management Module embraces its four submodules: Expert Management Module- EMModule, Resource Management Module- RModule, Resource Management Module-
Photocast Management Module–PhotoMModule, Podcast Management Module–PodMModule, and provides operations for deploying and undeploying of the module, respectively its sub-modules. The Expertise Finder Module includes its three sub-modules: Expert Finder Module–EFModule, Resource Finder Module–RFModule, Expert Answer Module–EAModule, and provides possibility for deploying and undeploying the its sub-modules and their web services. The Broadcasting Module contains its two sub-modules: Photocasting Module–PhotoCModule, and Podcasting Module–PodCModule, and allows for deploying and undeploying of the sub-modules and their web services. The implementation details of the sub-modules are omitted for simplicity and elaboted in the class diagrams of the individual modules in Section 4.3 - Section 4.9.

4.2.2 Overall Screen Flow

The GUI classes of the MobileHost CoLearn system can be presented in the form of a screen flow diagram, as shown in Figure 38. Upon starting the application, the learner should configure the user and server settings, before being able to deal with the web service modules. Screen SelectUser displays a list of currently registered users. The learner can directly login with any of the registered users, or first view the details of a user in the ViewUser screen, and, if needed, modify them in the EditUser screen. If the learner would like to create a new user account, he can do so in the AddUser screen. Setting up the server configuration is done in the ConfigureServer screen.
Figure 38: Overall Screen Flow of the MobileHost CoLearn Application
Screen **MainScreen**, as the name suggests, represents the main screen of the MobileHost CoLearn application. It gives a possibility to the user to select if he would like to manage the server status, view or edit the current user and server settings, or work with the currently deployed web service modules. The server can be started and stopped via the **ServerStatus** screen. The current user and server settings can be viewed in the **ViewUserSettings** and **ViewServerSettings** screens respectively by selecting the type of settings in the **Settings** screen, and if needed, these settings can be modified in the **EditUserSettings** and **EditServerSettings** screens.

The list of currently deployed modules, if any, can be viewed in the **Modules** screen. New web service modules can be deployed and currently deployed modules can be undeployed in the **ManageModules** screen. The **ExpertiseMgmtModule**, **ExpertiseFinderModule** and **ExpertiseCastingModule** screens are the main screens of the Expertise Management, Expertise Finder and Expertise Broadcasting Modules. Starting from these screens, the user has access to their sub-modules, which are elaborated in Section 4.3 - Section 4.9.

Notifications regarding the successful or unsuccessful execution of particular actions are displayed in the **AddedUser**, **EditedUser**, **ServerStatusChanged**, **EditedUserSettings**, **EditedServerSettings** and **ManagedModules** alerts.

### 4.3 Expert Management Module Implementation

As the Expert Management Module is a sub-module of the Expertise Management Module, its classes are contained inside package **expertmgmt**, which is a sub-package of **emgmt** - the package of the Expertise Management Module. The core non-GUI classes of the Expert Management Module and their relationships are presented in Section 4.3.1, and the GUI classes and their screen flow are reviewed in Section 4.3.2. The permanent storage of expert data on the file system of the mobile device is discussed in Section 4.3.3.

#### 4.3.1 Expert Management Class Diagram

The main non-GUI classes of the Expert Management Module are presented in Figure 39. The classes, related to the parsing of experts data from XML files, are omitted for simplicity. The GUI classes are also omitted and presented separately in Figure 40.

The Expert Management Module is represented by class **EMModule**, which contains the list of experts, the user is currently dealing with - experts, who most probably have been found through the Expert Finder Module, or who the learner knows and manually adds to the system. Class **EMModule** contains operations for deploying and undeploying the module, and managing of the list of experts, including adding, editing and deleting of experts, searching for experts by name and by expertise field, importing and exporting of experts data from and to XML files stored on the file system of the mobile device. Class **Expert** extends class **AbstractLearner** which is presented in detail in Section 4.6.1, and embraces the data of an expert, including personal information, such as names, IP address, phone number, e-mail address and other contact data, and expertise information, regarding the expert’s fields of
expertise, as well as a comment set by the current user for facilitating the identification of the person. Class `ExpertiseField` represents a field of expertise, and contains the field name and the corresponding level of expertise on a scale from 1 (the lowest) to 10 (the highest).

### 4.3.2 Expert Management Screen Flow

The screen flow of the Expert Management Module, related to the administering of expert data, is presented in Figure 40. The main screen of the module is the `ExpertMgmtModule` screen, which displays the names of the experts, currently being dealt with. This screen gives the possibility to the current user to select if he would like to add new experts, import and export experts, or view the details of an expert. Adding personal and expertise information for a new expert can be done in the `AddExpert` screen. Importing of experts data from XML files stored on the file system of the mobile device can be done by selecting the name of the XML file in the `FileBrowser` screen, viewing the contents of the file in the `ViewFile` screen and confirming the import operation in the `ImportExpertsConfirm` screen. Viewing of the details of a particular expert can be done in the `ViewExpert` screen after selecting the expert name in the main `ExpertMgmtModule` screen. After viewing the expert data, it can be edited in the `EditExpert` screen or deleted after confirming the operation in the `DeleteExpertConfirm` screen. Experts can be exported to XML files on the file system of the mobile device by selecting the names of the experts who should be exported in the `ExportExperts` screen and inputting the name of the XML file which will be created in the `ExportExpertsFile` screen.

Search for experts can be done by selecting one of the two possible types of search- search by name or search by expertise field- in the `SearchExperts` screen, and inputting the name or expertise field data in the `SearchByName` or `SearchByField` screens. The results from the two types of search are then shown in the `SearchedByName` and `SearchedByField` screens.
Figure 40: Screen Flow of the Expert Management Module
A list of the expertise fields for a particular expert can be viewed in the ManageFields screen. New fields of expertise can be added in the AddExpertiseField screen, current field information can be edited in the EditExpertiseField screen, and expertise fields can be deleted after confirmation in the DeleteExpertsConfirm screen.

Alerts ImportedExperts, AddedExpert, AddedExpertiseField, EditedExpertiseField, DeletedExpertiseField, EditedExpert and ExportedExperts provide notifications for successful or unsuccessful execution of the corresponding operations.

4.3.3 Storage of Experts’ Data

The permanent storage of experts’ data on the file system of the mobile phone is realized in XML files with an arbitrary name, chosen by the current user during the performing of the export operation. All these XML files are stored in the default for this purpose folder expertmgmt. The experts’ data can be later on imported from these files.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<experts xmlns="http://www.mobilehost.com/">
  <expert id="1">
    <name>Roger M. Jones</name>
    <phone>+4917652672250</phone>
    <ip>137.226.113.122</ip>
    <email>roger.jones@uni-karlsruhe.de</email>
    <other>also available by msn: roger.jones@hotmail.com</other>
    <expertiseFields>
      <expertiseField id="1">
        <field>Artificial Intelligence</field>
        <levelOfExpertise>9</levelOfExpertise>
      </expertiseField>
      <expertiseField id="2">
        <field>Mathematics</field>
        <levelOfExpertise>10</levelOfExpertise>
      </expertiseField>
    </expertiseFields>
    <comment>PhD student at Uni Karlsruhe, a colleague of Jamie</comment>
  </expert>
</experts>
```

Figure 41: Storage of Experts’ Data

The structure of an XML file, containing experts’ data, is shown in Figure 41. For simplicity the file contains data for only one expert, but in a real situation it is likely to contain a list of experts. The data about each expert embraces both the personal information, such as names, IP address, phone number, e-mail address and other contact data, and the expertise information covering the expertise fields of the person, as well as a comment set by the current user for easy identification of the expert.
4.4 Resource Management Module Implementation

The Resource Management Module is implemented as a sub-module of the Expertise Management Module, and as such, its classes are contained inside package `resourcemgmt`, which is a sub-package of `emgmt` - the package of the Expertise Management Module. The non-GUI classes of the Resource Management Module are described in Section 4.4.1, the GUI classes are presented in Section 4.4.2, and the permanent storage of the different types of resources on the mobile device is discussed in Section 4.4.3.

4.4.1 Resource Management Class Diagram

The core non-GUI classes of the Resource Management Module and their relationships are presented in Figure 42. The classes, related to the parsing of the different types of resource data are omitted for simplicity. The GUI classes are also omitted and reviewed in Section 4.4.2.

The class, representing the Resource Management Module, is called `RMModule`. It contains the list of resources of each resource type- articles, inproceedings, proceedings, books, URLs, master and PhD theses, unpublished resources- and provides the possibility to add new resources, edit or delete existing resources, import and export resources of each resource type, search for resources by title, author and related tags, as well as provides operations for deploying and undeploying of the module.

Class `AbstractResource` embraces the generic characteristics of all the resource types, such as title, author and related tags. Class `RelatedTag` represents a tag with a high, low or medium relevance to the specific resource. The different types of resources are represented by classes `Article`, `InProceedings`, `Proceedings`, `Book`, `Url`, `MasterThesis`, `PhDThesis` and `Unpublished`, all of which extend `AbstractResource` and contain additional resource-specific data. Class `Article` stands for an article in a journal or magazine, and contains additional attributes related to the journal name, year, volume and pages. Class `InProceedings` denotes an article in a conference proceedings, with additional attributes booktitle and year. Class `Proceedings` represents a conference proceedings and contains the title and year of the proceedings. Class `Book` designated a book with a title, author, publisher, year and editor. Class `Url` stands for a URL with its href. Class `AbstractThesis` is an abstract class, also extending `AbstractResource`, which contains the generic attributes of a thesis, such as title, author, school and year. The two supported types of thesis- master thesis and PhD thesis- are represented by classes `MasterThesis` and `PhDThesis`, which extend `AbstractThesis` and provide methods for proper exporting of the thesis data to XML files. Class `Unpublished` designates an unpublished resource and contains the resource title and author, as well as a note.

4.4.2 Resource Management Screen Flow

The screen flow diagram of the Resource Management Module is given in Figure 43.
4 MOBILEHOST COLEARN SYSTEM IMPLEMENTATION

Figure 42: Class Diagram of the Resource Management Module
Figure 43: Screen Flow of the Resource Management Module
The main screen of the Resource Management Module is `ResourceMgmtModule`, which displays a summary of the resources containing the number of resource items from each resource type currently being worked with. The user has the possibility to import and export resources from and to XML files, stored on the file system of the mobile device, after selecting the resource types to be imported or exported in the `ImportResourceTypes` and `ExportResourceTypes` screens respectively. The titles of resources, belonging to a particular resource type, can be viewed in the `ResourceTypeItems` screen, which also gives the possibility to add a new resource from the currently being dealt with resource type in the `AddResource` screen and view the details of a selected resource in the `ViewResource` screen. The currently viewed resource can be deleted via the `DeleteResource` screen. The tags, related to a particular resource, can be viewed and organized via the `ManageRelatedTags` screen: new tags alongside their high, medium or low relevance to the particular resource can be added in the `AddRelatedTag` screen, or already existing tags can be deleted after confirmation of the operation in the `DeleteRelatedTagConfirm` screen.

If there is a large number of resources currently being dealt with, the search functionality turns out to be very useful. Three types of search are currently supported: search by title, search by author and search by related tag. The search criteria for each of them can be defined in the `SearchByTitle`, `SearchByAuthor` and `SearchByTag` screens, and the results from the search are afterwards displayed in the `SearchedByTitle`, `SearchedByAuthor` and `SearchedByTag` screens respectively.

The alerts `ImportedResources`, `AddedResource`, `DeletedResource`, `ExportedResources`, `AddedRelatedTag` and `DeletedRelatedTag` notify the user about the status of the particular operation being performed.

### 4.4.3 Storage of Resource Data

Various XML files, placed under the default for this purpose folder `resourcemgmt` on the file system of the mobile phone, are used for the permanent storage of resource data. The databases of the individual literature types are stored by default in files `article.xml`, `inproceedings.xml`, `proceedings.xml`, `book.xml`, `url.xml`, `masterthesis.xml`, `phdthesis.xml` and `unpublished.xml`. A sample books XML file is shown in Figure 44, listing data about a book, including its title, author, editor, publisher and year of publishing, as well as its related tags. For simplicity the given resource file embraces information only about one resource, but in a real situation it is likely to contain a list of resources.

### 4.5 Photocast and Podcast Management Modules Implementation

As the functionalities of the Photocast and Podcast Management Modules are pretty similar, as described in Section 3.5.3, their implementations are presented side by side. The non-GUI classes, constituting the modules, are discussed in Section 4.5.1 and their GUI classes are presented in Section 4.5.2.
4.5.1 Photocast and Podcast Management Class Diagram

The class diagram of the Photocast and Podcast Management Modules is presented in Figure 45. The main classes, representing the modules, are `PhotoMModule` and `PodMModule`, which contain the list of photocast and podcast channels, created by the current user, and provide methods for adding, editing and deleting of channels, as well as for searching for channel episodes by tags. Both `PhotoMModule` and `PodMModule` extend the abstract class `BroadMModule`, which is in charge of the deployment and undeployment of the modules.

The abstract class `Episode` embraces generic data about an episode, such as title, enclosure, guid, publication date, description, author, subtitle, a list of tags, and eventually the feedback received by other users in case the Photocasting and Podcasting Modules are also deployed (Note: The methods denoted as ‘....()’ are the get and set methods for all the attributes of the class). The enclosure of an episode is represented by class `EpEnclosure`, which contains the URL, length and type of the episode. Class `Feedback` represents feedback, received by another user, and holds the data about the author of the feedback, his comment and the date of commenting. Classes `PhotoEpisode` and `PodEpisode` extend the abstract class `Episode`, providing additional methods for creating the SOAP request envelopes for the Receive Photocast and Receive Podcast Web Services respectively: this is needed in case the Photocasting and Podcasting Modules are deployed, so that the photocast or podcast episode can be sent to the subscribed users by calling the instances of the respective web services.
service, provided by the mobile phones of the users.

Class Channel is an abstract class, embracing the generic attributes, which are common to the different types of broadcast channels, such as title, link, description, author and copyright, as well as the list of channel subscribers in case the Photocasting and Podcasting Modules are deployed. The data of a subscriber is encapsulated in class Subscriber, which extends AbstractLearner presented in Section 4.6, and additionally holds the preferences of the user regarding the automatic download of content and the maximum size of content to be downloaded. Classes PhotoChannel and PodChannel extend Channel, containing the photocast and podcast episodes, belonging to the respective type of channel and providing methods for the management of the channel episodes.

Figure 45: Class Diagram of the Photocast and Podcast Management Modules
4.5.2 Photocast and Podcast Management Screen Flow

The GUI classes of the Photocast Management Module are presented in the screen flow diagram in Figure 46. `PhotocastMgmtModule` is the main screen of the module, displaying the list of channel titles, already created by the current user. The details of a channel can be viewed in the `ChannelDetails` screen, and afterwards edited or deleted via the `EditChannel` or `DeleteChannelConfirm` screens. In case the Photocasting Module is deployed, the list of channel subscribers can be viewed in the `ViewSubscribers` screen. New channels can be added after inputting the channel metadata in the `AddChannel` screen.

The list of episode titles of a particular channel are at hand in the `Episodes` screen. The photo image of an episode can be viewed in the `ViewEpisode` screen, and the details of the image are available in the `ViewEpisodeDetails` screen. The episode details can be edited via the `EditEpisode` screen, or the episode can be deleted after confirming the operation in the `DeleteEpisodeConfirm` screen. The list of comments, received from other users via the Photocasting Module in case it is deployed, can be viewed in the `ViewEpisodeComments` screen. New channel episodes can be created either by capturing an image with the integrated camera of the mobile phone in the `CameraCapture` screen and previewing the recorded image in the `ViewCameraSnapshot` screen, or by browsing the file system for image files in the `FileSystemBrowser` screen and previewing the image in the `ViewImage` screen, and then inputting the episode metadata in the `AddEpisodeDetails` screen. In case the Photocasting Module is deployed, at the time of its creation the new photocast is broadcast to the users, currently subscribed to the channel. Search for channel episodes can be performed by inputting the required tag in the `SearchEpisodesByTag` screen; the results are then displayed in the `SearchedEpisodesByTag` screen. Alerts `AddedChannel`, `AddedAndEvtlBCEpisode`, `EditedEpisode`, `DeletedEpisode`, `EditedChannel` and `DeletedChannel` display appropriate messages regarding the status of the performed operation.

The GUI classes of the Podcast Management Module are analogical to the ones of the Photocast Management Module, with the minor difference that the creation of new episode content can be performed after recording an audio file with the integrated audio capture device of the mobile phone or after browsing the file system for audio files.
Figure 46: Screen Flow of the Photocast Management Module
4.6 Expert Finder Module Implementation

The Expert Finder Module is implemented as a sub-module of the Expertise Finder Module. The core non-GUI classes of the module and their relationships are presented in Section 4.6.1, and the GUI classes and their screen flow are discussed in Section 4.6.2.

4.6.1 Expert Finder Class Diagram

The class diagram of the non-GUI classes of the Expert Finder Module, showing their attributes and operations and the relationships between them, is presented in Figure 47. The classes, related to parsing of XML files on the file system of the mobile device, permanently storing the data about requests belonging to the current user and requests received from other users, are omitted for simplicity. The presented classes are related to the types of expert search requests, separated into requests of the current user and requests of the other users. The classes are contained within the `expertfinder` package, which is a subpackage of the `efinder` package- the package of the Expertise Finder Module.

Class `AbstractLearner` is an abstract class, which encapsulates generic information about a learner, such as names, phone number, IP address, e-mail address and other contact data. Classes `Requestor`, `Forwarder` and `Responder` are subclasses of `AbstractLearner`, representing the different types of users on the request path: these classes do not have any additional attributes, but implement the operation `toXml()`, so that the requestor, forwarder and responder data can be properly exported to an XML file.

Class `AbstractRequest` is an abstract class, which embraces generic information about a request, be it a current user’s request or another user’s request, such as title, description, start and expiry dates, requestor. Class `MyRequest` is a subclass of `AbstractRequest`, representing a request created by the current user, and adding attributes related to the list of people the request has been sent to and the list of replies which have been received back from experts. Class `OthersRequest` is also a subclass of `AbstractRequest`, representing a request created by another user and received by the current user, which additionally comprises the list of forwards that the request has passed before reaching the current user, the list of forwards that the current user has sent the request to, and the response of the current user, in case it exists. Class `MyForward` represents a forward of the current user, containing the IP address of the recipient and a corresponding comment input by the current user.

Class `Forward` contains data about a forward from a user who does not assess himself as being an expert in the field to another user, including forward id within the path that the request has passed, forwarder, date and comment from the forwarder to the recipient. Class `Response` represents the response of a potential expert, including data about the responder, his own rating for his level of expertise in the field, an optional comment, and the date of response. Class `Reply` embraces information about a response from a potential expert and the path that the request has travelled before reaching the expert.
Figure 47: Class Diagram of the Expert Finder Module
Classes ExpertSearchService and ExpertRatingService represent the two web services of the Expert Finder Module- the Expert Search Web Service and the Expert Rating Web Service. Class EFModule represents the Expert Finder Module and contains the instances of the two web services, the lists of requests created by the current user and the list of requests received from other users.

4.6.2 Expert Finder Screen Flow

As the functionality of the Expert Finder Module encompasses plenty of operations that could be performed, the screen flow of the module is quite complex and therefore separated into two diagrams: Figure 48 presents the screen flow related to requests belonging to the current user, and Figure 64 in Appendix B shows the screen flow for requests created by other users and received by the current user.

ExpertFinderModule is the main screen of the Expert Finder Module, which gives the possibility to the user to select if he would like to deal with his own requests or handle the requests of the others. After selecting one of these possibilities, the user is transferred either to the MyRequests or OthersRequests screens and their respective sub-flows.

The sub-flow, starting from screen MyRequests and dealing with the requests created by the current user, is presented in Figure 48. This sub-flow allows the user to add, view, send, edit, delete, import and export requests, receive and view request answers, save experts. Importing of already saved requests is done by browsing the file system of the mobile device in the FileBrowser screen, viewing the contents of a file in the ViewFile screen, and confirming the import operation in the ImportMyRequestsConfirm screen. Adding details for a new request is at hand in the AddRequest screen. Viewing the details of an existing request is available in the ViewMyRequest screen after selecting the request title in the MyRequests screen. Exporting of requests can be performed after selecting the list of requests to be exported in the ExportMyRequests screen and inputting the file name under which the requests will be stored in the ExportMyRequestsFile screen. After having viewed a request, the user can send it to his acquaintances by selecting one of the available methods for sending in the SendRequestTo screen and afterwards inputting an IP address, inputting a phone number, or selecting a person from the contact list stored on the mobile device in the SendRequestToIP, SendRequestToPhone and SendRequestToContacts screens respectively. The current user can also view the list of people he has sent his request to in the SentTo screen. He can view the names of potential experts who have sent answers back to him in the Replies screen, and afterwards view the details of a reply in the ViewReply screen. If the current user really thinks that the responder is an expert and would like to collaborate with him, he can save him as an expert in the AddExpert screen. Upon performing any of these actions, corresponding alerts are displayed to the user in the ImportedMyRequests, ExportedMyRequests, AddedRequest, EditedRequest, DeletedRequest, SentRequest and AddedExpert alerts.
Figure 48: Screen Flow for My Requests of the Expert Finder Module
The screen sub-flow for others’ requests, shown in Figure 64 in Appendix B, enables the current user to view the list of request titles in the OthersRequests screen, view the details of a request in the ViewOthersRequest screen upon selecting the request title in the OthersRequests screen, reply to the request in case he evaluates himself as an expert in the ReplyToRequest screen, forward the request to his acquaintances if he does not asses himself as being an expert after inputting his own comment and selecting one of the possible methods for request sending in the ForwardRequest, ForwardRequestTo, ForwardRequestToIP, ForwardRequestToPhone and ForwardRequestToContacts screens, view the people he has already forwarded the request to in the ViewMyForwards screen, view the reply if any that he has sent back to the requestor in the ViewMyReply screen, delete the request after confirmation in the DeleteRequestConfirm screen. The current user can also import previously saved others’ requests by browsing the mobile phone file system in the FileBrowser screen, viewing the contents of the target file in the ViewFile screen, and confirming the import in the ImportOthersRequestsConfirm screen. Exporting of requests is at hand by selecting the list of requests to be exported in the ExportOthersRequests screen and inputting the file name under which the requests will be saved in the ExportOthersRequestsFile screen. Screens ImportedOthersRequests, RepliedToRequest, ForwardedRequest, DeletedRequest and ExportedOthersRequests contain appropriate notifications regarding a performed operation.

4.7 Resource Finder Module Implementation

The Resource Finder Module is also implemented as a sub-module of the Expertise Finder Module, and as such its classes are contained under package resourcefinder, which is a sub-package of the efinder- the package of the Expertise Finder Module. The non-GUI classes of the Resource Finder Module are presented in Section 4.7.1 and the GUI classes are discussed in Section 4.7.2.

4.7.1 Resource Finder Class Diagram

The class diagram of the Resource Finder Module is presented in Figure 49. The class, representing the Resource Finder Module is RFModule, which contains the list of resource finder requests, created by the current user, and the instance of the Resource Finder Web Service. ResourceFinderService is the class, representing the Resource Finder Web Service, which is responsible for receiving requests from other learners.

Class RequestRF embraces the data about a resource finder request, created by the current user. It contains the list of the requested literature types, such as articles, inproceedings, proceedings, books, URLs, master and PhD theses, and unpublished resources, as well as the data about the requested tag and its minimum relevance, and the list of responses, received by the Resource Finder Web Service, deployed on the mobile phones of the experts. Class ResponseRF represents a response to a Resource Finder request, containing the data about the expert and the different types of resources retrieved.
4.7.2 Resource Finder Screen Flow

The GUI classes, comprising the Resource Finder Module, are presented in the screen flow diagram in Figure 65 in Appendix B. The main module screen is the ResourceFinderModule screen, which displays the list of requests, created by the current user. New requests can be added by selecting the literature types of interest in the AddRequestResourceTypes screen and inputting the tag and its minimum relevance in the AddRequestRelatedTag screen. The details of existing requests can be viewed in the ViewRequest screen. A request can be sent to an expert after selecting the expert’s name in the SendRequestToExpert screen. The list of experts-responders is available in the ResponsesFrom screen, and the overview of the retrieved resources including literature types and number of resources per literature type can be viewed in the ResponseOverview screen. The list of resource titles, belonging to a particular literature type, is available in the ResourceTypeItems screen. The details of each retrieved resource can be viewed in the ViewResource screen. If the resource is of interest to the current user, it can be saved via the SaveResource screen. If the needed resources on the particular topic have already been retrieved, the request can be deleted after confirming the delete operation in the DeleteRequestConfirm screen. Alerts AddedRequest, SentRequestToExpert, SavedResource and DeletedRequest contain notifications about the successful or unsuccessful status of the corresponding performed operations.
4.8 Expert Answer Module Implementation

The Expert Answer Module is implemented as a sub-module of the Expertise Finder Module. The next sections elaborate on the Expert Answer Module implementation, introducing its non-GUI classes in Section 4.8.1 and its GUI classes in Section 4.8.2.

4.8.1 Expert Answer Class Diagram

The main classes, comprising the Expert Answer Module, are presented in the class diagram in Figure 50. They are related to the types of problems, divided into problems of the current user and problems of the other users, the proposed expert solutions to these problems, and the web services which make it possible. These classes are contained in the expertanswer package, which is a subpackage of the efinder package— the package of the Expertise Finder Module. The complex structure of the GUI classes is omitted for simplicity from Figure 50, and presented separately in the form of a screen flow diagram in Section 4.8.2.

AbstractProblem is an abstract class, which contains general information about a problem, such as problem title and description, as well as information about the the person who has faced the problem, such as names, phone number, IP address, e-mail address and other contact data. Class MyProblem represents a problem that the current user encounters and sends to other learners, who he has already found through the Expert Finder Module and has marked as being experts in the appointed field to one extent or another. MyProblem is a subclass of AbstractProblem, which additionally holds the list of experts, to whom the problem has already been sent, as well as a list of responses to the problem, which have been received back from experts. Class OthersProblem represents a problem, which the current user has received from other users, as they evaluate him as being an expert in the field. Class OthersProblem is also a subclass of AbstractProblem and contains supplementary information about the response that the current user has sent back to the requestor. Class ProblemResponse corresponds to a proposed expert solution to a problem, containing the responder data, such as names, phone number, IP address, e-mail address and other contact data, as well as his answer to the problem.

Classes ProblemService and ExpertAnswerService encapsulate data about the two web services of the Expert Answer Module. The main method of these services is the invoke() method, which gets as a parameter the body of the SOAP request message in the form of a byte array, deserializes the request into a SoapSerializationEnvelope, extracts the Java objects from the hierarchy of SoapObjects, processes the request, serialized it again into a byte array, and returns it back to the WSHandler class. The web service classes contain also instances of the TimestampsServer class, which are used for performance analysis, in case the performance analysis mode is enabled.

Class EAModule denotes the Expert Answer Module and contains instances of the two web services ProblemService and ExpertAnswerService, as well as the arrays of the two types of problems MyProblems and OthersProblems.
4.8.2 Expert Answer Screen Flow

The GUI classes, comprising the Expert Answer Module, are presented in the form of a screen flow diagram in Figure 51.
Figure 51: Screen Flow of the Expert Answer Module
They are related to the different possible types of interactions between the users and the
system, and generally divided into two sub-networks of screens, relevant to the problems
that the current user has faced and the problems that the other users have encountered.

The main screen of the Expert Answer Module is the ExpertAnswerModule, which gives the
possibility to the user to select one of the two possible sub-screen flows of the module, i.e.
the sub-flow dealing with the problems of the current user, or the sub-flow dealing with the
problems of the other users. After the user makes his selection by pressing the appropriate
button, he can see either the list of titles of his own problems in the MyProblems screen or
the list of titles of others’ problems in the OthersProblems screen.

The screen sub-flow, starting from the MyProblems screen, allows the user to perform
different operations on his own problems, such as add a new problem by going to the
AddProblem screen, view the details of a problem in the ViewMyProblem screen by se-
lecting the problem title in the MyProblems screen, send the problem to an expert that
he has already found through the Expert Finder Module by selecting the expert name
from the SendProblemToExpert screen, view the list of experts that he has already sent
the problem to in the SentToExperts screen, view the names of the experts who have
already replied to his problem in the RepliesFromExperts screen, view the details of
an expert reply in the ViewReply screen, delete a problem by confirming the action in the
DeleteMyProblemConfirm screen. For clarity of the performed problem operations,
appropriate notifications are displayed in the AddedProblem, SentProblemToExpert and
DeletedMyProblem alerts.

The screen sub-flow, starting from the OthersProblems screen, lets the user view the details
of another user’s problem in the ViewOthersProblem screen by selecting the problem title
in the OthersProblems screen, reply to the problem requestor by inputting his answer in
the ReplyToOthersProblem screen, view the reply that he has already sent to the requestor
in the ViewMyReply screen, or delete another user’s problem by confirming the action in the
DeleteOthersRequestConfirm screen. Appropriate notices regarding the successful or un-
successful completion of problem operations are displayed by the RepliedToOthersProblem
and DeletedOthersRequest alerts.

4.9 Photocasting and Podcasting Modules Implementation

As the functionalities of the Photocasting and Podcasting Modules are almost the same, as
described in Section 3.7, their implementations are presented side by side. The non-GUI
classes, constituting the modules, are discussed in Section 4.9.1 and their GUI classes are
presented in Section 4.9.2.

4.9.1 Photocasting and Podcasting Module Class Diagram

The class diagram of the Photocasting Module is presented in Figure 52 (the class diagram
of the Podcasting Module is omitted for simplicity, as it is the same). The main class,
representing the module, is PhotoCModule, which extends class ECastingModule presented
in Section 4.2.1, and contains the list of channels of other learners, the current user is subscribed to, and provides possibilities for dealing with the channels, as well as for deploying and undeploying of the module and its web services.

Class PhotoOthChannel extends class PhotoChannel, discussed in Section 4.5.1. It represents a channel, which is managed by another user, and to which the current user is subscribed. Class PhotoOthEpisode extends class PhotoEpisode, presented in Section 4.5.1, and represents an episode, which the current user has received from another learner. The web services suite of the Photocasting Module is represented by classes AvailablePhotoChService, SubscribePhotoChService, ReceivePhotocastService, CommentPhotocastService, RetrievePhotoCommentsService and UnsubscribePhotoChService.

### 4.9.2 Photocasting and Podcasting Module Screen Flow

The GUI classes of the Photocasting Module are presented in the screen flow diagram in Figure 46. The GUI classes of the Podcasting Module are quite similar, and therefore omitted for simplicity.
Figure 53: Screen Flow of the Photocasting Module
PhotocastingModule is the main screen of the Photocasting Module, displaying the list of channel titles, the current user is subscribed to. The user can subscribe to new channels by first retrieving the available channels of an expert by selecting the expert’s name in the SelectExpert screen, viewing the list of available channels in the AvailableExpertChannels screen, and setting his preferences for a specific channel in the ChannelPreferences screen. The details of a channel, the user is already subscribed to, can be viewed in the ChannelInfo screen.

The list of photocasting episodes, received through a particular channel, are available in the ChannelEpisodes screen. Each image episode can be previewed in the ViewEpisode screen, and the metadata of the image is at hand in the EpisodeInfo screen. The user can view the list of already retrieved comments for a photocast episode in the EpisodeComments screen, leave his own feedback to the photocast in the SendComment screen or delete the photocast after confirming the delete operation in the DeleteEpisodeConfirm screen.

In case there are many photocasts stored on the system, the user can search for articular ones by tag and author, after selecting the search choice in the SearchEpisodes screen, inputting the particular tag or author name in the SearchByTag and SearchByAuthor screens, and viewing the search results in the SearchedByTag and SearchedByAuthor screens.

The user is able to unsubscribe from a channel after selecting the channel title in the PhotocasingModule screen and confirming the operation in the UnsubscribeChannelConfirm screen. Alerts RetrievingChannels, SubscribingToChannel, RetrievingComments, SentComment, DeletedEpisode and UnsubscribedChannel provide notifications regarding the specific action performed.

4.10 Summary

This section elaborated on the implementation of the MobileHost CoLearn system. It started with a presentation of the technologies and tools being used, followed by the overall implementation of the system and its three main modules- Expertise Management, Expertise Finder and Expertise Broadcasting Modules. The particularities of the individual sub-modules and their web services were also presented in detail.
5 Evaluation

This section covers the evaluation of the MobileHost CoLearn system, including user evaluation, presented in Section 5.1, and performance analysis, discussed in Section 5.2.

The proposed architecture for a mobile web service provisioning based collaborative learning environment has been evaluated according to the software quality measures, defined in the international standard ISO 9126 [EAGL96]. These quality measures are divided into six main categories: functionality, reliability, usability, efficiency, maintainability and portability, each of which containing various measures, as shown in Figure 54. As not all of these measures can be evaluated, only a subset of them has been chosen, as discussed in Section 5.1 and Section 5.2.

![ISO 9126 software quality measures](Cons06)

Versatile methods for evaluation of the technical, semantic and effectiveness success of the developed mobile web services based system for supporting collaborative learning activities have been considered throughout pursuing the implementation phases and during the user and performance evaluation phases of the master thesis.

The technical success of the system has been measured, making sure that all the functional requirements of the system have been fulfilled. Especially important for the development of mobile learning applications is the performance, as mobile learning usually occurs on the way, while learners are trying to use their small fragments of free time. Using a slow learning system would make the learners reluctant to use the mobile environment. Using a fast system would make it interesting and engaging for them to further use the system. These are the reasons why performance testing has been performed throughout all the phases.
of this master thesis.

The semantic success and effectiveness success of the developed system has been evaluated with the help of user testing and a preliminary created questionnaire. Bearing in mind the mobility of the learners, the system should be able to guideline and support the learners in their new learning situations, when and where it is necessary. This is the reason why testing the usability of the system and the user satisfaction with the system and its information resources are so important.

5.1 User Evaluation

This section covers the user evaluation of the developed MobileHost CoLearn system. The methodology for user evaluation is presented in Section 5.1.1, followed by the personal profiles of the users and the test profiles, given in Section 5.1.2. The results from the user tests are discussed afterwards, including the overall system evaluation in Section 5.1.3 and the user satisfaction and suggestions for further improvement regarding each of the individual modules in Section 5.1.4 - Section 5.1.6.

5.1.1 Methodology for User Evaluation

The methodology employed in order to test the user experience with the MobileHost CoLearn system as a whole and the user satisfaction with the Expertise Management, Expertise Finder and Expertise Broadcasting Modules, is described in the next paragraphs. The users had to perform the following tasks in order to evaluate the functionality of the system and its modules:

1. Read an abbreviated version of the system description.
2. Start the MobileHost CoLearn application, configure the user and server settings, start the server.
3. Deploy the three main modules: Expertise Management, Expertise Finder and Expertise Broadcasting Modules.
5. Receive expert finder requests from other users via the Expert Finder Module. Forward one of the requests and reply to another request.
6. Save one of the found experts from the Expert Finder Module and manage its personal and expertise data in the Expert Management Module. Create additional expertise fields for an arbitrary expert.
7. Search for tagged literature resources of an arbitrary type via the Resource Finder Module.

9. Ask a question to an arbitrary expert via the Expert Answer Module and receive his answer. Reply to a question, asked by another user.

10. Create a photocasting channel via the Photocast Management Module. Create one photocasting episode by selecting an image file from the file system of the mobile device. Create another photocasting episode by capturing an image with the camera of the mobile device.

11. Retrieve the photocast channels of an expert via the Photocasting Module. Subscribe to one of the channels, receive and view a channel episode, comment to an episode, retrieve other’s comments, unsubscribe from the channel.

12. Create a podcasting channel via the Podcast Management Module. Create one podcasting episode by selecting an audio file from the file system of the mobile device. Create another podcasting episode by capturing audio from the mobile device.

13. Retrieve the podcast channels of an expert via the Podcasting Module. Subscribe to and unsubscribe from one of the channels.

14. Give feedback about their user experience in the prepared questionnaire.

The user questionnaire is given in Appendix C. The questionnaire has been separated into various sections, covering the personal profile of the user, the test profile, the user’s previous experience with mobile applications and collaborative learning applications, the user’s overall system evaluation, the evaluation of the each of the system modules, and final remarks.

The user profile section embraces the personal data of the user, such as names, sex, age, occupation/major, brand and type of mobile device used in everyday life, mobile phone functionalities being comfortable with, such as text input (SMS, E-mail), calendar and scheduling functionalities, camera recording (pictures, video), audio recording, picture viewing, video playback, audio playback. The test profile section includes the brand and type of mobile device used during the evaluation session, the date and duration of the test. The general questions section covers the familiarity of the users with mobile applications, collaborative learning applications and web services based collaborative learning applications.

The overall system evaluation section is related to the user experience with the system as a whole, without paying attention to the individual modules. The section includes 25 questions, which are a subset of the 50-question database of the Software Usability Measurement Inventory (SUMI)- a rigorously tested and proven method of measuring software quality from the end user’s point of view [UCCI07], which has assisted with the detection of usability flaws.
The evaluation section of the Expert Management Module includes questions regarding the problems faced during interaction with the module, such as viewing and managing the personal and expertise data of the experts, the comments regarding the faced problems, and the personal opinion of the user whether the module provides the necessary functionalities for managing the data regarding experts, their fields of expertise and their level of expertise. The questions in the evaluation section of the Resource Management Module are similar, as well as the ones in the sections related to the Photocast Management and Podcast Management Modules, which additionally cover the creation of photocasting and podcasting channels, the creation and broadcasting of photocast and podcast episodes: the creation of photocast episodes by capturing pictures with the camera of the mobile device or by browsing the file system, and the creation of new podcast episodes by capturing audio from the mobile device or browsing the file system of the mobile device for audio files.

The Expert Finder Module section embraces questions regarding the problems encountered during the managing and viewing of the user’s own requests and the requests of the other users, sending of own requests to other users, forwarding and replying to other users’ requests, as well as comments regarding the problems. It also covers the user assessment regarding the extend to which the module provides the needed functionalities for finding experts in specific fields, whether the current user thinks that the module has helped him find experts, whether the user is willing to help other users to find experts, and whether the user thinks that other users would be willing to help him find an expert. Some of the questions in the Resource Finder Module section are quite similar, other questions are related to the personal judgement of the user whether he has been able to define tags and their relevance to resources, matching the tags and relevance defined by the experts and whether the module has helped him find useful literature resources. The questions in the Expert Answer Module are also similar, related to the problems faced during the user interaction with the module, comments regarding these problems, and suggestions for further improvement.

The questions in the Photocasting and Podcasting Modules evaluation sections are related to the problems encountered while retrieving the list of available broadcasting channels of experts, subscribing to broadcasting channels, receiving broadcasting episodes, commenting to episodes and retrieving other user’s comments, and unsubscribing from channels, as well as the user’s comments regarding these problems and suggestions for further improvement.

The last section of the user questionnaire covers the final remarks of the users: their personal opinion whether the MobileHost CoLearn system encourages collaborative learning activities, whether they think that they can learn from fellows and their expertise by using the system, whether they feel willing to help others by using the system, and whether they are willing to use the system in real. The section also covers the user’s suggestions for collaborative learning scenarios, which they would like to have available in the next releases of the system, as well as other issues, which have not been covered by the predefined questions.
5.1.2 User and Test Profiles

The user evaluation of the MobileHost CoLearn system has been performed in two sessions. The first session involved 4 users. As there were two Subscriber Identity Module (SIM) cards with a public IP address available at that time, two users used mobile devices and the other two users used emulators. 3 users took part in the second session, one of them using a mobile device and two of them using emulators, as at that time only one SIM card was available. Altogether, 5 females and 2 males tested the system, most of them currently students at the age range 26-31 years. Both sessions durated a little bit more than 2 hours.

All users were using mobile phones in their everyday life, most of them were feeling comfortable with the text input capabilities of their mobile device, the calendar and scheduling functionalities, the capturing of images with the phone camera and the viewing of images, but only some of them have used audio recording before. Two of them have previously used the TomTom navigation system on their own mobile phones, most of them have played mobile games, one of them has used the MP3 player of the mobile device, but none of them has used a mobile application for collaborative learning before and none of them was aware of any mobile web service provisioning based projects for collaborative learning.

5.1.3 Overall User Evaluation of the MobileHost CoLearn System

As already mentioned, the statements related to the overall system usability are a subset of the statements, defined by the SUMI inventory. They embrace the user’s opinion towards the usability of the system, including measures such as learnability and understandability, the reliability of the system, such as fault tolerance and recoverability, the maintainability of the system, such as stability, the efficiency of the system such as time and resource behaviour, and the functionality of the system, such as accuracy and suitability. SUMI is backed by an extensive reference database embedded in an effective analysis and report generation tool. But, as this tool is highly priced, it has not been used for the analysis of the user feedback. Instead, the System Usability Scale (SUS) [Broo07], which is also based on a 5-grade scale, has been used for the evaluation of the results.

The result calculated by SUS yields a single number in the range from 0 to 100, representing a composite measure of the overall usability of the system being studied. Each possible answer is assigned a scale position from 0 (strongly disagree) to 4 (strongly agree). The points, contributed by each answer to the total score, depend on whether the statement is positive or negative and on the scale position number of the answer selected by the user. The contribution of each answer to the total score is calculated by the following rule: if the statement is positive, the statement answer contributes \((\text{selected scale position} - 1)\) points to the total score; if the statement is negative, the statement answer contributes \((5 - \text{selected scale position})\) points to the total score. As there are totally 10 questions in the SUS questionnaire, the sum of the points is multiplied by 2.5, so that a number in the range from 0 to 100 can be obtained.

In our case, the overall system evaluation includes 25 questions. As the goal is to obtain a
final score in the range from 0 to 100 as well, the SUS score calculation has been used, with the slight difference that the total score of the answer contributions is not multiplied by 2.5, as there are 25 instead of 10 statements, which are being evaluated by the users.

Based on the above described calculations, the final scores for the overall satisfaction of each of the 7 users of the MobileHost CoLearn system turned out to be in the range from 78 to 95 points. 1 of them is below 80 points, more precisely 78 points. 3 of them are between 80 and 90 points, more precisely 82, 85 and 89 points. The other 3 of them are above 90 points, more precisely 91, 95 and 95 points respectively. This results in an average user satisfaction of 87.86 points out of 100 points, or approximately 88%.

5.1.4 User Evaluation of the Expertise Management Module

The next paragraphs discuss the user evaluation of the sub-modules of the Expertise Management Module: Expert Management, Resource Management and Broadcast Management, including Photocast and Podcast Management Modules.

Regarding the Expert Management Module, the users were quite satisfied with the search possibilities for experts by name and by expertise field with a minimum level of expertise. One of the users did not know how to edit the skill level at first glance, but still was very happy with the presence of various levels of expertise. As an advice for further improvement some of the users suggested to have user photos available, because if there are many experts in the database, it would be easier to recall who the expert is by looking at his photo.

The users were very satisfied with the 8 types of literature resources supported by the Resource Management Module and the associated tags with three levels of relevance. They liked the search possibilities for all the literature types by title, author and tag with a minimum relevance. Two of the users mentioned that for some of the literature types it would be useful to have an abstract at hand.

Regarding the Photocast and Broadcast Management Modules, the users enjoyed the different ways of creating broadcast episodes: creating of photocast episodes by either taking a snapshot with the camera of the mobile phone or browsing the file system for image files, and creating of podcast episodes either by capturing audio from the mobile device of browsing the file system for audio files. One of the users thought that the phone crashed while taking a picture with the camera, and this was due to the fact that she was not pressing the right button (at that time there were two buttons on the screen, and she was uncertain of their meaning; one of them was removed later; the additional button seems to be a device-specific button, created automatically by the mobile phone). With another user the starting of the camera was a bit slow, but with most of the users the speed was normal, as fast as the build-in camera functionalities of the phone.
5 EVALUATION

5.1.5 User Evaluation of the Expertise Finder Module

The results from the user evaluation of the sub-modules of the Expertise Finder Module, namely the Expert Finder, Resource Finder and Expert Answer Modules, are discussed in the next paragraphs.

The users were truly satisfied with the collaborative environment that the Expert Finder Module creates. They were happy with the fact that they could not only see the path that the request has travelled before reaching them, but also read the comments that each forwarder has made to the next forwarder on the path. As a suggestion for further improvement they mentioned that they would like to have the user photos available, and to have not only a textual, but also a graphical visualization of the path, as much as this is possible on the screen of the specific mobile device being used. All in all, the users found the user interface of the Expert Finder Module to be simple and intuitive, and were very satisfied with the seamless coupling of the Expert Search and the Expert Rating Web Services with the user interface. The users said that they would also like to have a mechanism for automatic discovery of experts, so that if they do not know anyone who might be expert in the field, the system could perform an automatic search.

Regarding the Resource Finder Module, the users were pleased with the variety of literature resources, which can be retrieved, such as articles, books, etc. As some of the users were not really sure if the tags, which they search for, match the tags, which the experts have defined, they suggested that it would be better to have a wizard for related tags. But, as tagging is something subjective, they concluded that it is best to have predefined categories, so that first a search by related tag can be performed, and if no results are returned, then a search by a predefined category can be carried out. Additionally, the users would like to be able to retrieve the latest resources, added by an expert to his literature database, and to be able to view the latest tags, used by an expert.

The Expert Answer Module received only positive feedback, as the users did not come across any flaws in the way it works and did not have any difficulties in using the module. The users were very pleased with the seamless integration of the Problem and Expert Answer Web Services. Their only remark referred to the fact that sometimes a problem might be too complex and its answer might be too long, and as input capabilities for mobile devices are not as advanced as input capabilities for desktop computers, they would probably prefer to use another means of communication with the problem requestor than typing long answers and sending them through the Expert Answer Web Service. They concluded that if they were at the position of a problem requestor, they would by all means use the Problem Web Service in order to inquire experts about the problem, but if they were an expert and the answer to the problem is not so simple, they would prefer to contact the problem requestor by writing an e-mail or calling him back on the phone, therefore they were very happy with the fact that they could see the complete contact data of the problem requestor, including his e-mail address, phone number, etc.

Regarding the Expert Finder and Expert Answer Modules, the users said that it would be better if upon arrival of a new request they do not receive an SMS message, but the specific
screen related to the request pops up. This problem could be overcome if the used mobile phone supports running of more than one MIDlet at a time. In this case, using the Push Registry of the mobile device, the SMS notification sent to the recipient of the request could be automatically processed and a second MIDlet could be started. This solution has not been integrated in the thesis, as most of the mobile phones in use nowadays, such as the Sony Ericsson W810i phone used in the thesis, do not support running of two MIDlets at the same time.

5.1.6 User Evaluation of the Expertise Broadcasting Module

The two sub-modules of the Expertise Broadcasting Module, namely the Photocasting and Podcasting Modules, received positive feedback regarding the retrieval of available photocasting and podcasting channels and especially regarding the possibility for defining user preferences upon subscription: selecting whether to download the content automatically or only to be notified about its availability, and defining the maximum content size to be automatically downloaded in case this option is selected. They encountered no problems broadcasting their own small photocasts and receiving other learners’ small photocasts, but faced problems while transferring large podcasts. The users were very happy with the possibility to leave feedback on each photocast episode and to view the comments that other users have written. The suggestions for further improvement included the tagging of an image and of particular areas of the image, similarly to the way that it is available on Flickr\footnote{http://www.flickr.com/}. In addition, the users would like to have vodcasing available in the future releases of the system.

5.2 Performance Analysis

The web services of the various modules of the MobileHost CoLearn system have been extensively tested regarding their performance. The methodology for performance analysis is presented in Section 5.2.1, followed by the results from the performance analysis of the Expert Search and Expert Rating Web Services in Section 5.2.2, the Resource Finder Web Service in Section 5.2.3, the Problem and Expert Answer Web Services in Section 5.2.4, the Receive Photocast and Receive Podcast Web Services in Section 5.2.5.

5.2.1 Methodology for Performance Analysis

This section describes the methodology employed and the experiments conducted in order to evaluate the performance of the MobileHost CoLearn system. The main goal of the performance analysis was not only to observe the total times at the server and client sides for the web services of the different MobileHost CoLearn modules, but also to determine the division of the total times for the individual tasks, such as creation of the SOAP request and response envelopes, serialization and deserialization, processing and transmission times.
The scenario for evaluation involves repetition of complete request-response cycles, in which the timestamps on the client and server sides were taken, and the times needed for performing the individual operations were calculated. In total, 10 timestamps need to be taken for a complete performance evaluation of a web service: 5 timestamps on the client side and 5 timestamps on the server side.

Timestamps, taken on the client side, include:

- $T_{S_{c}\text{Start}}$ - timestamp at the beginning of the creation of the SOAP request envelope;
- $T_{S_{c}\text{RequestCreated}}$ - timestamp at the end of the creation of the SOAP request envelope;
- $T_{S_{c}\text{RequestSerialized}}$ - timestamp at the end of the serialization of the SOAP request envelope and the beginning of the transmission of the request message from the client to the server;
- $T_{S_{c}\text{ResponseReceived}}$ - timestamp at the completion of the transmission of the server response to the client;
- $T_{S_{c}\text{ResponseProcessed}}$ - timestamp at the end of the processing of the server response message.

Timestamps, taken on the server side, embrace:

- $T_{S_{s}\text{RequestArrived}}$ - timestamp when an incoming request arrives at the server listening socket;
- $T_{S_{s}\text{RequestRead}}$ - timestamp after the incoming request has been read from the server listening socket;
- $T_{S_{s}\text{RequestDeserialized}}$ - timestamp at the end of the deserialization of the SOAP request envelope;
- $T_{S_{s}\text{RequestProcessed}}$ - timestamp at the end of the processing of the request;
- $T_{S_{s}\text{ResponseSerialized}}$ - timestamp at the end of the serialization of the response envelope and the beginning of the response transmission to the client.

The timestamps taken at the server and client sides during a request-response scenario are depicted in Figure 55. As it can be seen, taking of the timestamps on the client side is pretty straightforward, while taking of the timestamps on the server side is much more complicated, as different levels are involved. Timestamp $T_{S_{s}\text{RequestArrived}}$ is taken by the server instance, at the time when a new incoming request arrives at its listening socket. After the server creates a new instance of the request handler for handling the particular request and the request handler reads the data from the input stream of the opened socket connection, the next timestamp $T_{S_{s}\text{RequestRead}}$ is taken. As it is a web service request, the request handler on its side creates an instance of the web service handler, which determines which web
service exactly is called and creates an instance of this particular web service. The next three timestamps—\(T_{S_s}^{\text{RequestDeserialized}}, T_{S_s}^{\text{RequestProcessed}}\) and \(T_{S_s}^{\text{ResponseSerialized}}\)—are taken by the specific web service instance during the handling of the request and creating of the response, which is afterwards returned to the client.

![Figure 55: Timestamps Taken for Performance Analysis](image)

Next, based on the client and server timestamps, the times for performing the individual operations were calculated as follows:

- Time for creation of the SOAP request envelope on the client side:
  \[ t_{reqCr} = T_{S_c}^{\text{RequestCreated}} - T_{S_c}^{\text{Start}} \]

- Time for serialization of the SOAP request envelope on the client side:
  \[ t_{reqS} = T_{S_c}^{\text{RequestSerialized}} - T_{S_c}^{\text{RequestCreated}} \]

- Time for transmission of the SOAP request message from the client to the server:
  \[ t_{reqTr} = T_{S_s}^{\text{RequestRead}} - T_{S_c}^{\text{RequestSerialized}} \]

- Time for deserialization of the SOAP request envelope on the server:
  \[ t_{reqDs} = T_{S_s}^{\text{RequestDeserialized}} - T_{S_s}^{\text{RequestRead}} \]

- Time for processing the request on the server:
  \[ t_{reqProcess} = T_{S_s}^{\text{RequestProcessed}} - T_{S_s}^{\text{RequestDeserialized}} \]
5 EVALUATION

- Time for serialization of the SOAP response envelope on the server:
  \[ t_{resS} = TS_{ResponseSerialized} - TS_{RequestProcessed} \]

- Time for transmission of the SOAP response message from the server to the client:
  \[ t_{resTr} = TS_{ResponseReceived} - TS_{ResponseSerialized} \]

- Time for processing of the received response on the client:
  \[ t_{resProcess} = TS_{ResponseProcessed} - TS_{ResponseReceived} \]

- Total time for transmission, including time for transmission of the request from the client to the server and time for transmission of the response from the server to the client:
  \[ t_{Tr} = t_{reqTr} + t_{resTr} \]

- Total time taken on the client side:
  \[ t_{cTotal} = TS_{ResponseProcessed} - TS_{Start} \]

- Total time on the server side, including receiving of the request, processing of the request headers and actual web service processing:
  \[ t_{sTotal} = TS_{ResponseSerialized} - TS_{RequestArrived} \]

The performance evaluation was performed using two Sony Ericsson mobile phones: Sony Ericsson P990i and Sony Ericsson W810i, which were alternatingly exchanged as a client and a server. Each of the experiments was repeated 7 times and the mean values of the calculated times was used for creating the performance analysis charts.

5.2.2 Performance Analysis of the Expert Search and Expert Rating Web Services

As the size of an Expert Search SOAP request message varies, depending on the path that the request has passed so far, a scalability analysis has been performed for evaluating the time increase, caused by the presence of one more forward on the path. The overall scalability analysis of the Expert Search Web Service, tested in case of 0, 2, 4, 6, 8, 10, 15 and 20 forwards, is shown in Figure 56, with a SOAP body message size of 1.07 KB, 1.68 KB, 2.29 KB, 2.90 KB, 3.50 KB, 4.11 KB, 5.63 KB and 7.15 KB respectively.

The scalability analysis shows that the presence of one more forward in the request message increases the total time on the client with 0.276 seconds on average, and the total time on the server with 0.265 seconds on average. This is quite acceptable, as the difference between the initial expert finder request with 0 forwards and an expert finder request containing 20 forwards is only 5.524 seconds on the client side and 5.298 seconds on the server side.
The performance of an Expert Rating SOAP request, containing $n$ forwards, is analogue to the performance of the Expert Search SOAP request, containing $n-1$ forwards, as the message size and the depth of the SOAP objects hierarchy are alike.

The division of the total time for performing the individual operations on the client and server sides for an Expert Search SOAP request without forwards is presented in Figure 57. The transmission time of 5.087 seconds constitutes approximately 86% of the total time; the time for creating and serialization the request on the client side is only 0.003 seconds, comprising 0% of the total time; the time for deserialization and processing of the request and serialization of the response on the server side is 0.443 seconds, or 7% of the total time; the time for processing the response on the client is 0.416 seconds- 7% of the total time.
5.2.3 Performance Analysis of the Resource Finder Web Service

The main goal of the performance analysis of the Resource Finder Web Service is to discover the dependency of the total server and client times on the number of literature types requested and the number of resources retrieved. For this purpose tests have been performed for retrieving 8 and 24 resources from 1, 2, 4 and 8 different resource types. The total times on the server for each of these cases are shown in Figure 58, and the total client times are given in Figure 59.

![Performance Analysis of the Resource Finder Web Service Server](image)

Figure 58: Performance Analysis of the Resource Finder Web Service Server

The total server times in Figure 58 show no clear dependency neither on the number of literature types requested nor on the number of resources retrieved. Neither of the curves is constantly increasing or constantly decreasing. The difference between the times, in case of retrieving 8 and 24 resources from 1, 2, 4 and 8 literature types, is in terms of milliseconds, as all times are slightly higher than 1 second, more precisely within the range 1.032 to 1.260 seconds. In general, but not always, the times in case of retrieving 24 resources are slightly higher than when retrieving 8 resources, as more time is needed for the creation and serialization of the SOAP response message.

As it can be seen from Figure 59, the total client time does not depend on the number of literature types requested, but there is a clear dependency of the client time on the number of resources retrieved. The times in case of retrieving 8 resources are within the range 6.311 seconds to 6.818 seconds, and the times in case of retrieving 24 resources range from 8.132 to 8.996 seconds. The more resources are retrieved, the more time is needed for transmission and processing of the SOAP response message, therefore the higher the total time on the client is.
5.2.4 Performance Analysis of the Problem and Expert Answer Web Services

The results from the performance analysis of the web services belonging to the Expert Answer Module- the Problem Web Service and the Expert Answer Web Service- are presented in Figure 60. The timestamps are taken for SOAP request envelope sizes of 1.3KB for the Problem Web Service and 1.7KB for the Expert Answer Web Service- the normally expected SOAP body sizes, as we know that an empty from content Problem request message is 842 bytes, an empty Expert Answer request message is 1153 bytes, and as we suppose that a user’s data is 100 characters, the title and description of the problem input by the requestor contain 400 characters, and the answer input by the expert is 500 characters.

As it can be seen from Figure 60, the time needed on the client side is 6.365 seconds for the Problem Web Service and 6.951 seconds for the Expert Answer Web Service, around 83% of which is spent for transmission. The times for creating of the SOAP envelope and its serialization on the client side are unnoticeable on the chart, as they are only 0.001 seconds. The performance of the server is very good: the total time on the server, including reading of the stream through the socket connection, deserialization, processing and serialization, is 1.213 seconds for the Problem Web Service and 1.276 seconds for the Expert Answer Web Service, with the processing time being only 0.298 seconds and 0.335 seconds respectively. In general, the times needed for the Expert Answer Web Service are slightly bigger than the ones for the Problem Web Service, as the expert answer message contains not only the original problem information, but also the expert contact data and the expert answer, and therefore slightly more time is needed for transmission, serialization, deserialization and processing of the request.
5.2.5 Performance Analysis of the Receive Photocast and Receive Podcast Web Services

The broadcasting of image content takes approximately 4.82 seconds for a picture size of 2.4KB. The Receive Photocast and Receive Podcast Web Services could not be properly evaluated, as tests with large audio data files could not be realistically performed. This is due to the fact that the Mobile Host server currently does not support the full HTTP 1.1 specification, more specifically SOAP with attachments. It supports the chunked transfer encoding, whereby large data is transferred after being broken down into chunks: this turns out to work reasonably good with image files, but is not sufficient for the proper transfer of large audio data. Moreover, the binary image and audio data is sent over the air after being base64 encoded, which causes an overhead of approximately 33% of the real binary data size. This is why no realistic performance analysis of the Receive Podcast Web Service could be performed.

5.3 Summary

This section covered the evaluation of the developed MobileHost CoLearn system. The methodology employed in order to test the user experience with the system as a whole and the user satisfaction with each of the individual modules, namely the Expertise Management, Expertise Finder and Expertise Broadcasting Modules, was presented first, followed by the presentation of the results from the user evaluation and a discussion of the user suggestions for further improvement. Next, the performance analysis of the web services, belonging to the various system modules, was extensively presented, including the methodology used for evaluation and the results from the performance analysis.
5 EVALUATION
6 Conclusion And Future Work

The MobileHost CoLearn system is the main outcome of this master thesis, providing a mobile collaborative learning environment, based on web service provisioning from mobile phones. This section summarizes the design, development and evaluation of the MobileHost CoLearn system, overviews encountered issues and outlines various perspectives for further development of the system.

6.1 Conclusion

This master thesis is the first of its kind, combining the mobile web service provisioning domain with the collaborative learning domain, in order to support learners in finding exactly the valuable up-to-date information that they need within the constantly increasing overflow of information. The main objectives of the developed MobileHost CoLearn system are to enable learners to:

• Manage their own personal and expertise data and make it available to other learners;
• Search for experts in a specific field within a truly collaborative mobile environment; Expand the expert finder search beyond the borders of the social network of the current user by allowing the forwarding of expert finder requests an arbitrary number of times, until a real expert has been found;
• Manage the data of the experts that have been found; Narrow down the list of experts by filtering by name and expertise field with a specific minimum level of expertise;
• Contact experts for further assistance and inquiry them about problems faced;
• Manage their own databases of literature resources; organize them into categories of articles, inproceedings, proceedings, books, URLs, master thesis, PhD thesis, and unpublished resources; Tag them with specific keywords, associated with a three-level scale of relevance to the resource; Narrow down the list of literature by filtering the resources by title, author, tags; Make the literature databases accessible to other learners;
• Retrieve the different types of literature resources of other learners by specifying the tag of interest and its minimum relevance to the resource;
• Manage their photocasting and podcasting channels; Create new photocast episodes by capturing images with the integrated camera device or browsing the file system of the mobile phone, and create new podcast episodes by capturing sound with the integrated audio recording device or browsing the file system of the mobile phone; Manage the metadata of the channels and their episodes; Tag them with specific keywords;
• Subscribe to photocasting and podcasting channels; Define preferences, such as whether to automatically download the content or just be notified of its availability, and the maximum size of the downloaded content; Unsubscribe from the photocasting and podcasting channels later on;

• Automatically receive new broadcasts as soon as they are available, in case their content complies with the user preferences; Narrow down the list of broadcast episodes by filtering by author and tags;

• Write comments to the received broadcasts; Retrieve the feedback that other learners have left.

In order to fulfill these functional requirements, the MobileHost CoLearn system has been designed, containing three main modules: Expertise Management Module, Expertise Finder Module, and Expertise Broadcasting Module, as discussed in Section 3:

• Expertise Management Module: deals with the administration of the data, gathered by the current user, such as expert data and literature resource data, as well as with his own broadcast data, such as photocasts and podcasts;

• Expertise Finder Module: enables learners to search for experts in a specific field, and after having found an expert, to ask him questions regarding issues they have faced; allows for the retrieval of the tagged literature resources of other learners;

• Expertise Broadcasting Module: supports the subscription to photocasting and podcasting channels, managed by other learners, and the automatic retrieval of their up-to-date content.

The developed MobileHost CoLearn system has been extensively tested regarding its user satisfaction and performance analysis, with results elaborated in Section 5. Overall, the users were very satisfied with the functionality of the system, particularly with the collaborative expert finder opportunities, and the variety of literature resources and broadcasts, which are supported. They were particularly pleased with the fact the all they need, in order to use the collaborative learning environment, is their mobile phones, which they all carry on with them every day anyway. As a summary the users concluded that they are willing to use the system in real, as long as the costs of the generated network traffic are not too high.

The detailed performance analysis of MobileHost CoLearn system under different scenarios has clearly shown that the system is realistic and can scale to the demands of large collaborative learning groups in mobile networks.

6.2 Future Work

All in all, the MobileHost CoLearn system provides the grounds and proves the feasibility of a mobile collaborative learning environment, based on web service provisioning from mobile
phones. There are however a couple of perspectives, in which the system could be further improved: the first one is related to the usability of the currently developed modules, the second one is concerned with the broadening of the functionality spectrum of the system, and the third one is associated with the potential technical enhancements of the system.

Regarding the usability of the MobileHost CoLearn system, an improvement is possible through the employment of ontologies for the expertise fields in the Expert Finder Module and the literature tags in the Resource Finder Module. Currently, the Expert Finder Module enables the learner to collaboratively discover experts, who can help him with a particular problem he has faced. As the flow of the expert finder request might be complex, and the learners might not respond quickly, it could take some time before the learner receives answers back. However, this is not suitable in situations, where the learner needs immediate help. In such cases, it would be better to first automatically discover people with a general knowledge in a field, and then narrow down the list of potential experts by further inquiring them about the particularities of the problem. This possibility for automation of the discovery process can be made available through the employment of ontologies of the different expertise fields, and providing the results from the discovery through a request-response web service. Some of the discovered by these means learners with a general knowledge in the field might not be able to answer the particular question, but still the process of finding the right person might be faster. This is similar with the tags, associated with the literature resources in the Resource Finder Module. As tagging is something subjective, the tags defined by the experts and the tags searched for by the non-experienced users might greatly vary. A solution for this problem is the introduction of a generic mechanism for tags, which could be achieved through developing tag ontologies for the different possible literature domains.

Regarding the functionality spectrum of the MobileHost CoLearn system, further improvement is envisioned in the extendability of the system with further collaborative learning components, the most important one being related to collaborative editing of text and image documents. The support for multimedia feedback is of significant importance, as pointed out by the users. Currently, the feedback exchanged between learners can be only in textual form, for example the commenting opportunities of the Expert Finder, Expert Answer, Photocasting and Podcasting Modules. In the next releases of the system, the image, audio and video feedback could be supported.

Regarding the technical aspects of the MobileHost CoLearn system, it can be further improved to support the full HTTP 1.1 specification. The MobileHost server currently supports the chunked transfer encoding, but does not support SOAP with Attachments. The chunked transfer encoding is used when a large amount of data is being transferred, by breaking the complete stream into smaller chunks and sending them in series. But still, encapsulation of binary data, such as the image and audio files, used in the Photocasting and Podcasting Modules, can not be done without additional encoding and decoding of the data. The overhead of encoding binary data in a form, suitable for a SOAP message, is often significant, both in terms of bytes added due to the encoding, as well as in terms of processor overhead for performing the encoding and decoding. For example, in the base64 encoding defined by
RFC 2045, which is used by the MobileHost CoLearn system through kSOAP2, the encoded data is approximately 33% larger than the unencoded one. This has turned out to be acceptable for the distribution of photocasts, but not good enough for podcasts. In order for podcasting to work well with large audio content, a good solution would be to improve the MobileHost server to support SOAP with Attachments. In this way, by using a multipart MIME structure for transport, the photocasts and podcasts could be attached to the SOAP message in their native format, for example PNG and WAV - the employed by default J2ME MMAPI formats.

\footnote{RFC 2045 specification is available at http://rfc.net/rfc2045.html}
Appendices
A Additional SOAP Request and Response Messages

POST / HTTP/1.1
SOAPAction: expertSearch
Content-Type: text/xml
Content-Length: 2010
User-Agent: KSOAP/2.0
Host: 137.226.232.31:8868

<?xml version='1.0' encoding='UTF-8'>
  xmlns:SOAP-ENC="http://www.w3.org/2001/12/soap-encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns0:expertSearch xmlns:ns0="http://www.mobilehost.com/" xmlns="http://www.mobilehost.com/*">
      <request xsi:type="ns0:request">
        <requestor xsi:type="ns0:learner">
          <phone xsi:type="xsd:string">+4917623500216</phone>
          <ip xsi:type="xsd:string">134.130.122.57</ip>
          <name xsi:type="xsd:string">iliyana Ivanova</name>
          <email xsi:type="xsd:string">iliyana@yahoo.com</email>
          <other xsi:type="xsd:string">icq:12341234</other>
        </requestor>
        <title xsi:type="xsd:string">JME JSR-75 Permissions in JAD</title>
        <description xsi:type="xsd:string">Help me to configure JSR-75 permissions in my JAD file</description>
        <expiryDate xsi:type="xsd:dateTime">Fri Jun 15 10:25:22 UTC 2007</expiryDate>
      </request>
    </ns0:expertSearch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Figure 61: Expert Search SOAP Request Message (Level 2 to Level 3)
Figure 62: Expert Search SOAP Response Message (Level \( k+1 \) to Level \( k \))

```
HTTP/1.1 200 OK
Server: MobileHost CoLearn
Date: Thu May 31 10:25:32 UTC 2007
Content-Length: 679
Content-Type: application/soap+xml; charset=utx-8
Request-ID: 8
Connection: close

<?xml version="1.0" encoding="UTF-8">
  <SOAP-ENV:Body>
    <ns1:expertSearchResponse xmlns:ns1="http://www.mobilehost.com/">
      <ns1:status xsi:type="xsd:string">Your expert search request was successfully received!</ns1:status>
    </ns1:expertSearchResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 63: Problem SOAP Request Message

```
POST / HTTP/1.1
SOAPAction: problem
Content-Type: text/xml
Content-Length: 1158
User-Agent: KSOAP/2.0
Host: 137.226.232.31:8608

<?xml version="1.0" encoding="UTF-8">
  <SOAP-ENV:Body>
    <ns1:problems xmlns:ns1="http://www.mobilehost.com/">
      <ns1:problem xsi:type='ns1:problem'>
        <requestor xsi:type='ns1:learner'>
          <phone xsi:type='xsd:string'>++4417623580216</phone>
          <ip xsi:type='xsd:string'>131.130.122.57</ip>
          <name xsi:type='xsd:string'>Tliyana Ivanova</name>
          <email xsi:type='xsd:string'>tliyana@yahoo.com</email>
          <other xsi:type='xsd:string'>tcq:12341234</other>
        </requestor>
        <title xsi:type='xsd:string'>Security Issue for Mobile Media API</title>
        <description xsi:type='xsd:string'>I gave MIDlet permissions for audio and video control to java.microedition.media.control.RecordControl, but still got a Security Exception... are there some other permissions that I should give to the MIDlet?</description>
      </ns1:problem>
    </ns1:problems>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
B Screen Flow Diagrams
Figure 64: Screen Flow for Others’ Requests of the Expert Finder Module
Figure 65: Screen Flow of the Resource Finder Module
C User Questionnaire

User Evaluation Steps

Dear User,

Please perform the following tasks in order to evaluate the functionality of the MobileHost CoLearn system and its modules:

1. Read the abbreviated version of the system description.
2. Start the MobileHost CoLearn application, configure the user and server settings, start the server.
3. Deploy the three main modules: Expertise Management, Expertise Finder and Expertise Broadcasting Module.
5. Receive expert finder requests from other users via the Expert Finder Module. Forward one of the requests and reply to another request.
6. Save one of the found experts from the Expert Finder Module and manage its personal and expertise data in the Expert Management Module. Create additional expertise fields for an arbitrary expert.
7. Search for tagged literature resources of an arbitrary type via the Resource Finder Module.
8. Save the retrieved literature resources from the Resource Finder Module and manage their data in the Resource Management Module. Create additional related tags for the resources.
9. Ask a question to an arbitrary expert via the Expert Answer Module and receive his answer. Reply to a question, asked by another user.
10. Create a photocasting channel via the Photocast Management Module. Create one photocasting episode by selecting an image file from the file system of the mobile device. Create another photocasting episode by capturing an image with the camera of the mobile device.
11. Retrieve the photocast channels of an expert via the Photocasting Module. Subscribe to one of the channels, receive and view a channel episode, comment to an episode, retrieve other's comments, unsubscribe from the channel.
12. Create a podcasting channel via the Podcast Management Module. Create one podcasting episode by selecting an audio file from the file system of the mobile device. Create another podcasting episode by capturing audio from the mobile device.
13. Retrieve the podcasting channels of an expert via the Podcasting Module. Subscribe to and unsubscribe from one of the podcasting channels.
14. Give us feedback about your user experience with the MobileHost CoLearn system via the questionnaire.
System Description

The MobileHost CoLearn system provides a mobile collaborative learning environment, based on web service provisioning from mobile devices. The system contains three main modules:

- Expertise Management Module — Enables the user of the mobile device to manage the data regarding experts in specific fields via the Expert Management Module, manage different types of resource data, such as articles, proceedings, proceedings, books, URLS, master and PhD theses, and unpublished resources via the Resource Management Module, create and manage podcasts and podcasts in the Photocast and Podcast Management Modules;
- Expertise Finder Module — It allows the user to search for experts in a specific field via the Expert Finder Module, search for resources via the Resource Finder Module, and ask experts for help via the Expert Answer Module;
- Expertise Broadcasting Module — It enables the user to retrieve the available photocast and podcast channels of a particular expert, subscribe to channels, receive channel episodes, and unsubscribe from channels via the Photocasting and Podcasting Modules.

After logging into the MobileHost CoLearn system and defining the user and server settings, the user can select which of the main modules to be deployed. At the time when a main module is deployed, its submodules are deployed as well.
User Questionnaire

Dear User,

Please take a moment to give us your precious feedback about the MobileHost CoLearn System. The questionnaire covers issues regarding your overall system evaluation, as well as your satisfaction with each of the individual MobileHost CoLearn modules. Your opinion is highly appreciated.

User Profile

Name: .................................................................................................................................

Occupation / Major: ...........................................................................................................

Sex  □ male  □ female

Age: ........... years

Brand and type of mobile device, used in everyday life: ....................................................

Mobile phone functionalities, you feel comfortable with:

☑ text input (SMS, E-mail)
☑ calendar and scheduling functionalities
☑ camera recording (pictures, video)
☑ audio recording
☑ picture viewing
☑ video playback
☑ audio playback

Test Profile

Brand and type of mobile device, used during the test: .....................................................

Date of test: .........................

Test duration: ........................

General Questions

Have you used any mobile applications before? If yes, please list their names.

______________________________________________________________________________

______________________________________________________________________________
Have you used any mobile applications for collaborative learning before? If yes, please list their names.

Are you aware of any mobile web service provisioning based applications for collaborative learning? If yes, please list their names.

### Overall System Evaluation

Against each statement there are five boxes, representing your response to that statement. You should mark the 1st box if you strongly disagree with the statement, mark the 2nd box if you generally disagree, mark the 3rd box if you are undecided, mark the 4th box if you generally agree, and mark the 5th box if you strongly agree with the statement.

1. It takes too much time to learn how to work with the system.
2. I can understand and act on the information provided by the system.
3. It is easy to make the system do exactly what you want.
4. It is easy to see at a glance what the options at each stage are.
5. There are too many steps required to get something to work.
6. I sometimes don’t know what to do next.
7. I sometimes wonder if I’m using the right command.
8. I feel safer if I use only a few familiar commands or operations.
9. It is relatively easy to move from one part of a task to another.
10. The system has a very attractive presentation.
11. The organization of the menus or information lists is quite logical.
12. The system allows the user to be economic of keystrokes.
13. The system occasionally behaves in an unexpected way.
14. Error prevention messages are not adequate.
15. The system had at some time stopped unexpectedly.
16. If this system stops, it is not easy to restart it.
17. The speed of this system is fast enough.
18. The help information provided by the system is not very useful. □ □ □ □ □
19. The instructions and prompts are helpful. □ □ □ □ □
20. The system documentation is very informative. □ □ □ □ □
21. The way the system information is presented is clear and understandable. □ □ □ □ □
22. It is obvious that the user needs have been taken into consideration. □ □ □ □ □
23. Tasks can be performed in a straightforward manner using this system. □ □ □ □ □
24. I feel in command of this system when I am using it. □ □ □ □ □
25. Working with this system is a pleasure. □ □ □ □ □
26. I would recommend this system to my colleagues. □ □ □ □ □

**Expert Management Module Evaluation**

Did you encounter any problems while:

- viewing experts data yes □ □
- managing experts data □ □
- viewing expertise fields □ □
- managing expertise fields □ □

Comments regarding technical problems you have faced while using the module:


Do you feel that the Expert Management Module provides the necessary functionalities for managing the data regarding experts, their fields of expertise and their level of expertise? If no, what is missing?


**Resource Management Module Evaluation**

Did you encounter any problems while:

- viewing resource data □ □
- managing resource data □ □
Comments regarding technical problems you have faced while using the module:

Do you feel that the Resource Management Module provides the necessary functionalities for managing the resource data of articles, proceedings, books, URLs, master and PhD theses, unpublished resources? If no, what is missing?

Podcast Management Module Evaluation

Did you encounter any problems while:

- viewing podcast data [ ] [ ]
- managing podcast data [ ] [ ]
- capturing audio from the mobile device [ ] [ ]
- broadcasting your podcasts [ ] [ ]

Comments regarding technical problems you have faced while using the module:

Do you feel that the Podcast Management Module provides the necessary functionalities for managing your podcasts? If no, what is missing?
Comments regarding technical problems you have faced while using the module:
__________________________________________________________________________________________

Do you feel that the Podcast Management Module provide the necessary functionalities for managing your podcasts? If no, what is missing?
__________________________________________________________________________________________

**Expert Finder Module Evaluation**

Did you encounter any problems while:
- viewing your own requests
- managing your own requests
- viewing others requests
- sending your own request
- forwarding another user's request
- replying to another user's request

Comments regarding problems you have faced while using the module:
__________________________________________________________________________________________

Do you think that the Expert Finder Module provides the needed functionalities for finding experts in specific fields? If no, what is missing?
__________________________________________________________________________________________

Do you think that the Expert Finder Module has helped you find an expert?

☐ yes  ☐ no  ☐ not sure

Are you willing to help other users find an expert?

☐ yes  ☐ no  ☐ not sure

Do you think that the other users are willing to help you find an expert?

☐ yes  ☐ no  ☐ not sure

**Resource Finder Module Evaluation**

Did you encounter any problems while:
- viewing your requests

Comments regarding problems you have faced while using the module:
__________________________________________________________________________________________
C USER QUESTIONNAIRE

- managing your requests ☐ ☐
- sending your request and receiving replies ☐ ☐

Comments regarding problems you have faced while using the module:

Do you think that you have been able to define tags and their relevance to resources, matching the tags and relevance defined by the expert?
☑ yes ☐ no ☐ not sure

Do you think that the Resource Finder Module has helped you find useful resources?
☑ yes ☐ no ☐ not sure

Expert Answer Module Evaluation

Did you encounter any problems while:
- viewing your requests ☐ ☐
- managing your requests ☐ ☐
- viewing another user’s requests ☐ ☐
- sending your request ☐ ☐
- replying to another user’s request ☐ ☐
- receiving replies from other users ☐ ☐

Comments regarding problems you have faced while using the module:

Suggestions for further improvement

Photocasting Module Evaluation

Did you encounter any problems while:
- subscribing to other users’ photocasts ☐ ☐
- unsubscribing from other users’ photocasts ☐ ☐
- receiving another user’s photocasts ☐ ☐
- viewing other users’ photocasts ☐ ☐
- writing comments to photocast episodes ☐ ☐
- retrieving the comments left by other users ☐ ☐
Comments regarding problems you have faced while using the module:


Suggestions for further improvement:


Podcasting Module Evaluation

Did you encounter any problems while:
- subscribing to other users' podcasts
- unsubscribing from other users' podcasts
- receiving another user's podcasts
- viewing other users' podcasts
- writing comments to podcast episodes
- retrieving the comments left by other users

Comments regarding problems you have faced while using the module:


Suggestions for further improvement:


Final Remarks

Do you find that the MobileHost CoLearn system encourages collaborative learning activities?
- yes  ☐  no  ☐  not sure  ☑

Do you think that you can learn from fellows and their expertise by using the MobileHost CoLearn system?
- yes  ☐  no  ☐  not sure  ☑
Do you feel willing to help others by using the MobileHost CoLearn system?

☐ yes ☐ no ☐ not sure

Are you willing to use the MobileHost CoLearn system in real?

☐ yes ☐ no ☐ not sure

Are there some other scenarios for collaborative learning, which you would like to be available by the MobileHost CoLearn system?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Issues, which you would like to mention, but were not covered by the above questions:

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Thank you for taking your time in completing this form!
Your answers will help us shape the future of the MobileHost CoLearn system!
D Bibliography

References


[CJFW07] Mohamed Amine Chatti, Matthias Jarke, and Dirk Frosch-Wilke. The future of e-learning: a shift to knowledge networking and social software.


REFERENCES


REFERENCES


REFERENCES


[Sun06a] Sun Microsystems Inc. The java me platform the most ubiquitous application platform for mobile devices. Available online from: http://java.sun.com/javame/index.jsp [Retrieved on 25.11.2006 22:00].


REFERENCES


