

Adaptive M-learning application development, based on UCD and the mLUX framework

Business case: M-learning application for driving licenses

Maurizio Casarini



Author(s) Maurizio Casarini	
Degree programme Business Information Technology	
Report/thesis title Adaptive M-learning application development, based on UCD and the mLUX framework	Number of pages and appendix pages 42 + 13
<p>The mobile application consumer landscape evolves rapidly, and presents consistent challenges to software and service providers. Moreover, implementing M-learning systems poses additional difficulties. In fact, mobile learning application development requires extensive consideration, as the application deals with learning and learners alike. Apps must meet students' essential educational requirements, and encourage students to engage with the application on their mobile devices.</p> <p>System adaptation and context awareness have been consolidated features to leverage m-learning application potential. Firstly, to homogeneously perform on a wide variety of smart devices available on the market. Secondly, data and presentation had to target specific user needs and cognitive capabilities. Finally, context awareness has proved to be particularly valuable for interaction based on handheld devices.</p> <p>This report describes implementation of a web application prototype, to demonstrate how adaptive mobile learning can support the necessary training to obtain a Finnish driving license. Business rationale was provided by Haaga driving school ltd, Helsinki (Finland).</p> <p>Implementation relied on the mLUX development framework for m-learning (Dirin & Nieminen 2015), a User-Centred Design methodology focused on ensuring users involvement throughout the implementation process: from user studies to product development, including usability testing.</p> <p>The implemented application was a HTML5 based rich and contextually adaptive web application, mainly based on Microsoft technology. Open Source tools were utilized to develop additional features, mostly for presentation adaptation and user experience tuning.</p> <p>The mLUX framework resulted in an application fulfilling all of the usability criteria. The prototype evaluation report indicated that the application is easy to use and provides the essential learning materials for driving school candidates.</p>	
Keywords Adaptivity, M-learning, User-Centered Design, web development, ASP.NET MVC	

Table of contents

Acknowledgements.....	1
Abbreviations and terms	2
1 Introduction	3
2 Business case: M-learning application for driving licenses	4
3 Goals, Objectives and Scope	6
3.1 Goal	6
3.2 Objectives	6
3.3 Scope	6
3.4 Out of scope	6
4 Background.....	8
5 Methods	10
5.1 User-Centered Design	10
5.2 mLUX framework	11
5.2.1 mLUX layers	12
5.2.2 mLUX phases	13
6 Design.....	15
6.1 User study.....	15
6.2 Data analysis	15
6.3 Idea Creation	16
6.4 Product concept.....	17
6.5 Need for adaptability and context awareness.....	17
7 Implementation.....	19
7.1 Conceptual system architecture	19
7.2 Content-level adaptation	23
7.3 Trainee module.....	24
7.3.1 Trainee main menu	25
7.3.2 Driving Session self-evaluation feature	26
7.3.3 Theory test feature	28
7.4 Instructor and administrator tools	29
7.5 Usability testing.....	32
7.5.1 Test design	32
7.5.2 Test result	33
8 Conclusion	36
9 Challenges and possible future work.....	37
References	38
Appendices.....	43

Appendix 1. Adaptivity algorithm: determine the most relevant content category.....	43
Appendix 2. Adaptivity algorithm: fetch a list of multimedia resources by category.....	44
Appendix 3. Adaptive algorithm: sort multimedia objects and format for UI layer.....	45
Appendix 4. Detailed usability testing scenario results by test subject.....	46
Appendix 5. Overall questionnaire ratings	51
Appendix 6. Scenario completion rate	53
Appendix 7. Scenario completion time	54
Appendix 8. Adaptation, UI and UX key features by user feedback.....	55

Acknowledgements

I would like to express my appreciation and gratitude to the following people, for their contribution and support over the years spent at Haaga Helia University of Applied Science, in Helsinki (Finland).

Dr. Amir Dirin, my thesis supervisor for granting me the honor to participate this project. For mentoring and introducing me, as Haaga Helia senior lecturer, to new fascinating IT subjects, further fostering my user-centric vision and knowledge of IT implementation methodologies. Among them: User-Centred Design, m-learning, application adaptability, context awareness and usability

Mr. Juhani Välimäki and Mr. Markku Kuitunen, senior lecturers at Haaga Helia for shaping my knowledge of object oriented programming. I know I have to learn more.

Mr. Kari Silpiö, senior lecturer at Haaga Helia for being the best database expert I have ever met.

Mr. Ralf Rehn, senior lecture at Haaga Helia for taking me into a systematic and consistent journey into information system development.

All of Haaga Helia personnel, for enabling a long journey in the smoothest possible way.

Abbreviations and terms

Table 1. A list of abbreviations, acronyms, and terms, used in this report.

Abbreviation	Meaning
App	Software application
ASP.NET	A Microsoft's web application framework
ASP.NET MVC	A Microsoft's MVC-based web development framework
BYOD	Bring your own device
UCD	User-centered design
M-learning	Mobile learning
mLUX	Usability and user experience development framework for m-learning
MVC	Model, view and controller (software architectural pattern)
ORM	Object-relational mapping
REST	Representational state transfer
SOAP	Simple object access protocol
UI	User interface
UX	User experience

1 Introduction

Smartphone's penetration among young users is increasing at a fast pace (Maged et.al, 2011). In fact, mobile device adoption is in continuous positive trend (Tilastokeskus 2012), and they overtook PCs as the most common Web access device (Google Inc. 2015), as predicted by Gartner (2012). Moreover, the numbers of downloaded mobile applications such as game, social networking, entertainment, personal and professional are increasing rapidly (Statista 2015). Accordingly, mobile apps and services providers are constantly competing for users' time and attention (Dirin et.al, 2013), and business entities grounding strategies on mobile are able to engage the constantly connected customers (Google Inc. May 2012.). In short, it can be stated mobile devices transformed consumer digital behavior. Hence, it is paramount to properly implement mobile applications, winning the challenges related to emerging technologies and fast evolving market context.

This thesis paper describes the implementation of an adaptive web application prototype, as a proof of concept for a specific business case: *Haagan Autokoulu M-learning application for driving licenses*. The system was designed applying the UCD principle (Abrams, Maloney-Krichmar & Preece 2004), and relying on the mLUX mobile application framework (Dirin & Nieminen 2015.). Implementation was .NET based.

This document provides an outline of the overall project (e.g. rationale, goals, and objectives), theoretical grounding principles, and system implementation.

2 Business case: M-learning application for driving licenses

Haagan Autokoulu is a driving school providing services to customers willing to acquire a new driving license, and/or upgrade an already owned one. Their offerings are designed to meet customers' expectations, a critical factor for a company willing to strengthen its competitive advantage, and to widen its customer base. This strategy is pursued in several ways.

In fact, the company:

- Offers *what* clients want, e.g. targeting several driving card types, both in relation to theory and driving test (*Haagan Autokoulu*.)
- Provides the services *where* clients want, by having several offices located in the Greater Helsinki area. Hence it allows clients to attend theory classes in a convenient location (*Haagan Autokoulu*, 2011.)
- Leverages technology to increase business processes efficiency, e.g.: company website, email, providing driving simulators for training (*Maurizio Casarini*, 2012.)

Nonetheless limitations occur, e.g.: offices are available only in the Greater Helsinki area, theory classes are delivered according to a specific schedule. Moreover the same content is delivered to all customers having the same goal (e.g. getting the same driving card type, such as passenger car.), without user-specific customization.

Therefore, in order to overtake such limitations, it is necessary to evaluate their own nature. Physical constraints (e.g. offices locations, training personalization) are hard to bypass, requiring consistent investments (e.g. facilities, equipment, personnel). Technology, on the other hand, offers considerable potential, especially accordingly to the previously introduced latest trends.

Hence, *Haagan Autokoulu* management is interested in evaluating whether and how leveraging adaptive mobile learning would improve training effectiveness and efficiency. Accordingly, following contacts with *Hagaa Helia*'s representatives, an agreement was reached to address the issue.

Therefore, an adaptive mobile learning web application prototype will be developed, to provide the following benefits:

- Enable Haagan Autokoulu management to better understand the potential of adaptive mobile learning. In fact, the decision of heavily investing in a fully featured application cannot be grounded on solid evidence, at the present state.
- The mobile learning landscape is rapidly changing, making very difficult to characterize its unique nature (Sharples et al. 2007, 5.) Hence, an empirical approach, within Haagan Autokoulu specific context, is expected to provide useful insights.

3 Goals, Objectives and Scope

3.1 Goal

This thesis project is meant to create a working web application prototype, to demonstrate how adaptive mobile learning can support the training process necessary to obtain a Finnish driving license.

3.2 Objectives

Primary project objectives are:

- Determine a set of application features, adequate to fulfill project goal, e.g. a proof of concept implementation.
- Evaluate which features are needed to implement M-learning system adaptation, in this business case context.
- Implement the above mentioned features, in a working application prototype

3.3 Scope

The project will be in scope if the delivered application prototype:

- Implements all of the features, as determined in accordance with stakeholders' representatives.
- It meets the expected quality standards, as per the *Usability testing* paragraph.
- It is implemented and hosted on Microsoft environment, as per request of Haagan Autokoulu management. Open source technologies are allowed to integrate Microsoft technologies, to enable project goal fulfilment.

3.4 Out of scope

The project will **not**:

- Deliver a fully featured web application. In fact, a well-defined and limited amount of features/requirements will have to be agreed upon within the stakeholders' representatives, including required adaptation features.

- Include a full spectrum research. In fact, the project is system based. Accordingly, the research objectives and results will be limited, in accordance with project objectives.
- Be a *state of the art* technological product. In fact, it will be a prototype, and such will have to be considered. This is due to the nature of the project itself (e.g. resources limitations, not implemented by a seasoned software developer/graphic designer), and its technological focus (e.g. adaptation, particularly through HTML5).

4 Background

The mobile application consumer landscape evolves rapidly, and presents consistent challenges to software and service providers, as previously stated. Moreover, implementing M-learning systems poses additional difficulties.

In fact, Mobile learning application development requires extensive consideration, as the application deals with learning and learners alike. Apps must meet students' essential educational requirements, and also encourage students to engage with the application on their mobile devices (Keinonen, Lindholm & Kiljander 2003; Seong 2006, 2-3; Zhag & Adipat 2005, 293-294; Mostakhdemin-hosseini, 2009). Moreover, M-learning systems should follow specific pedagogical principles and requirements (Kukulaska-Houlme, 2007; Trinder, Magill & Roy 2007; Corlett & Sharples, 2004; Mostakhdemin-Hosseini, 2008).

Accordingly, development of the *M-learning application for driving licenses* required extensive functional and non-functional considerations. The system had to enable driving school candidates to complete the compulsory theory lessons on their smartphones. The application functionality implementation associated with two distinct challenges: content preparation and user interface design. The content of driving licenses theory lessons must follow the standard mandates and regulations defined by government organization. Additionally, the content of the theory lessons must be easy to comprehend, simple to use and smooth to navigate. As a result, multi-formatted content e.g. audio, video and animations are considered the best approach.

The application follows the user's learning progress by conducting evaluation test after each theory lesson. Moreover, the application provides a self-evaluation form after each practical driving session. The application then forwards the completed self-evaluation form to proper instructors at the driving school. Moreover, it provides them further notification, if immediate attention is required, according to submitted self-evaluation form report content.

The system also recommends the follow-up learning lessons based on user's knowledge and reading history, a functionality part of application adaptivity. There have already been many initiatives (Vainio & Ahnonen, 2003) and (Jäppinen et.al 2004, 110) providing adaptive mobile learning applications, to improving the usability and learning processes.

Content adaptivity and application usability were considered as major factors for easing the learning process and knowledge creations. The non-functional requirements of the

proposed driving license application are as important as the functional requirements. The application should be reliable, the user should also feel secure to use and most of all be usable. In this paper however, focus is mainly on the functional requirements and the development steps, based on the mLUX UCD-based framework for mobile learning application development.

5 Methods

Methodological cornerstones for developing the adaptive m-learning application for driving licenses were:

- The User Centered Design (UCD) principles (Abrams & al. 2004).
- The mLUX development framework (Dirin & Nieminen 2015).

A brief description of these items is provided in this chapter.

5.1 User-Centered Design

The term *User Centered Design* originated by Don Norman's book *User-Centred System Design, New Perspective on Human-Computer Interaction* (Norman & Draper, 1986). UCD, as well as development of interactive systems and devices, has an increasing importance in product development organizations (Nieminen, 2004, 1). In addition, UCD is the most common method for developing a smart product. Gould & Lewis (1985, 301) and Gould, Boies & Ulkeson (1997, 235) argued that users' involvement must be continuous, in a usable system, and design modified according to their feedback. User-Centered Design cuts both costs (Bosert 1991; 2; Gulliksen & al. 2003, 405) and improves usability, since it continually focuses, as early as possible, on the essential customer's needs. Moreover, user involvement has been confirmed as the most recurring project success factor, in IT systems development (The Standish Group 2014, 8). For all of these reasons UCD proved to be particularly valuable to tackle all of the project challenges, among them those derived from the fast changing landscape of the m-learning domain, and the need to fully exploit its potential.

In user-centred design, user's requirements are the focus in all stages of the product development lifecycle. The three different design solutions for UCD are, as defined by ISO (2010) and Sharp & Rogers (2006, 295, 306):

- *Cooperative* design; designers and users involved in all stages.
- *Participatory* design; users' occasionally participate in the design process.
- *Contextual* design; design based on the actual context.

The UCD phases are as follow, and a more detailed process template is shown in Figure 1 (Gulliksen & al. 2003, 401-403):

- Get to know the users.
- Analyse user tasks and goals.
- Establish usability requirements.
- Prototype and then design concept.
- Usability test of the concept.
- Repeat the stages as there are more features/ services.

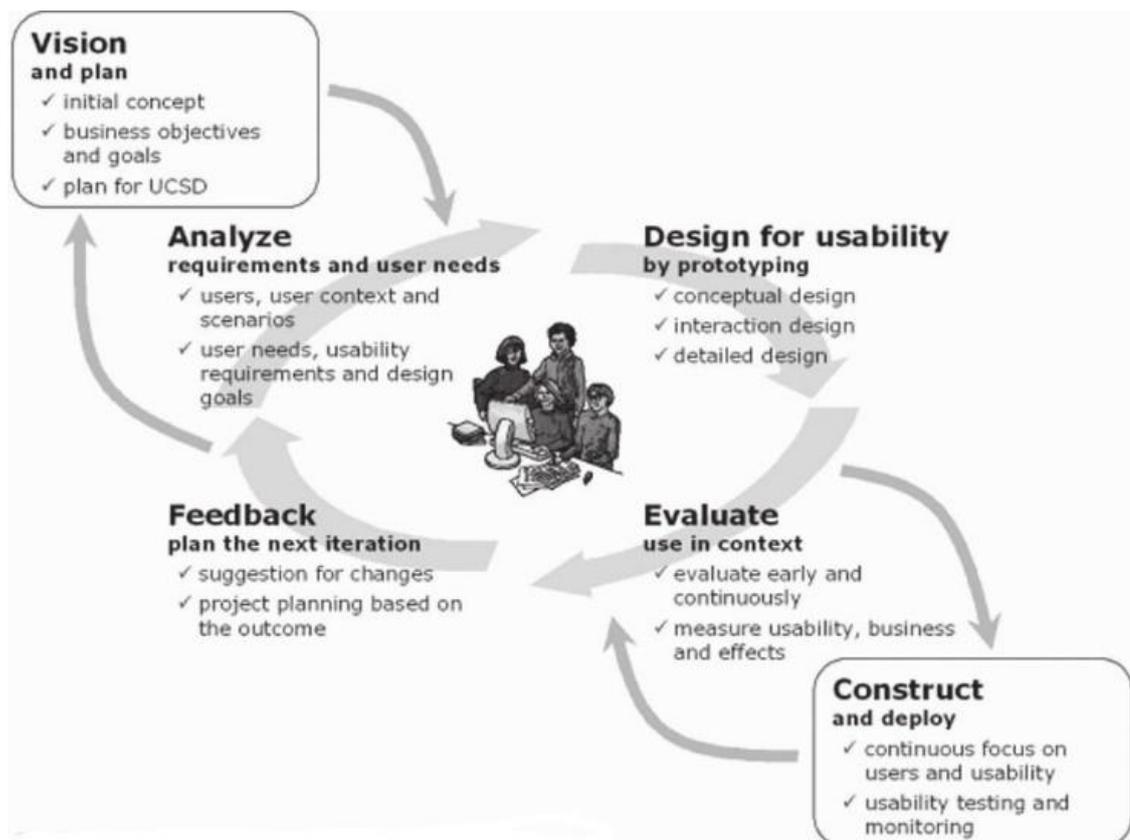


Figure 1. User-centred system design process (Gulliksen & al. 2003, 401).

5.2 mLUX framework

This paragraph provides an outline of the mLUX methodology, and describes the layers and phases it is modeled upon. mLUX (Dirin and Nieminen 2015) is a framework to design and develop usable mobile learning applications. It is focused on usability and user experience, and grounded on UCD principle as described by Abras & al. (2004). In fact, it centers design and development on users' need, throughout the system implementation lifecycle (Dirin & Nieminen 2015, 37). Moreover, its main goal is to ensure fulfillment of stakeholders' educational requirements (Dirin & Nieminen 2015, 37), primarily students

but also teachers and others, by efficiently and efficaciously supporting implementation of usable mobile learning applications.

5.2.1 mLUX layers

The mLux framework consists of three layers, as described by Dirin & Nieminen (2015, 40-43):

- The *role-players* are actors involved with the m-learning application, in one or more its stages. The term *role-player* is particularly suited in the m-learning development context, because explicitly adds the concept of *role* to the more often used of *actor* or *stakeholder*. In fact, due to the typically collaborative nature of UCD-based solutions, previously herein hinted at, it is paramount to understand not only who the users are, but also their role and needs in relation to the application (Abrás & al. 2004, 4). Role-players are not limited to the obvious application users (e.g. students, teachers and administrators), but also

...many invisible role-players, individuals, organizations, systems, and technical systems are involved in the design and development of m-learning applications. These invisible role-players are not the actual users of the application, but they have a significant influence on the design and usage of the m-learning application (Dirin & Nieminen 2015, 40.)

- The *context of use* is related to application capability to create user-centric customized learning environment, and it is a recurring characteristic in m-learning. Furthermore, it is a major component of system adaptability (Dolog 2008, 23). In fact, to fulfill nowadays educational requirements, technology enhanced learning must provide user specific support, matching individual skills and competences (Aroyo & al. 2006, 7; Gilliot & Garlatti 2009, 2; Jongyi, Eui-Ho, Junyoung & SuYeon 2009, 7448). Accordingly, several factors must be accounted for, and according to Dirin & Nieminen (2015, 41) they are conceptualized in the *social* (e.g. social acceptance of m-learning as a suitable medium), *physical* (e.g. in relation to time and location constraints), and *educational* context (e.g. learning assignments, material, topics, outcomes, and pedagogical approaches).
- The *process*, a four-phased methodology meant to guide project participants throughout application implementation. It is briefly described in next paragraph.

5.2.2 mLUX phases

mLUX-based implementations are structured into four phases, meant to organize key project activities, displayed in figure 2, and herein described in sequence.

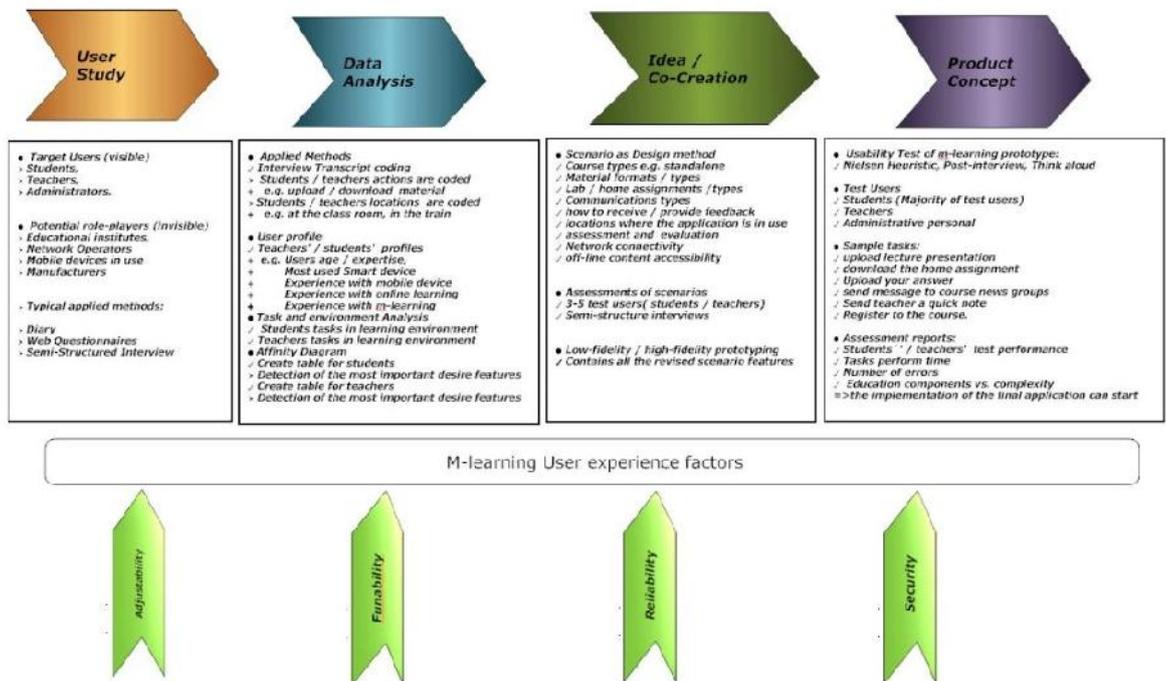


Figure 2. Overview of the mLUX adaptive learning application development framework (Dirin & Nieminen 2015, 43).

The **user study** phase is the essence of the UCD for mobile learning application development. At this phase target role-players are identified, and their real needs are investigated by applying various user study methods such as a diary, interview etc. The user study method such as a diary helps the designers to learn about the user and their environment in which they usually interact. Standard UCD methodology recommends 3-10 target users' involvement at various user studies stage (Usability Research Group, 2002).

Goal of **data analysis** phase is to sort out primary data resulting from the user study phase. Gathered raw data, is processed to identify users' real needs. Various analysis methods enable classification and categorization of user requirements (Dirin & Nieminen 2015, 42). Affinity diagrams are the recommended for categorizing and prioritizing the requirements.

During **idea creation** phase categorized and prioritized requirements must be assessed and confirmed, in collaboration with potential test users. The objective is to ensure that designers properly understood users' needs. Additionally, this is yet another opportunity to

involve users, and strengthen validity of the proposed requirements. The categorized requirements are presented as scenarios. Scenarios are extremely efficacious in the mobile learning application concept design method, as they introduce tasks in the user's language (Dirin & Nieminen 2015, 42), and often avoids technical terms and complexities. Moreover, scenario value has been previously confirmed by Rosson & Carrol (2002, 20).

In the **product concept** phase application concept is modified, according to the outcome of the previous phase, and ready to be implemented as a non-functional prototype. The prototype consists of the potential user interface components and the navigations of various screens. In the last stage of users' involvement in the design process, usability testing is relied upon to evaluate the proposed application prototype. The target application design refinement is based on usability testing results. The UCD for mobile learning application is an iterative design method. Moreover, the concept development mandates the users' involvement at all phases of m-learning application development, minimizing application failure to effectively meet requirements, and maximizing application usage among target audience.

6 Design

The m learning application for driving school was designed with particular focus on adaptivity and usability, in accordance to the adopted mLUX framework, in order to fulfill well defined functional and non-functional requirements. This chapter provides an outline of the design process.

6.1 User study

The first phase of the mLUX framework mandates the direct involvement of a potential application users' sample for understanding users' needs and expectations (Dirin & Nieminen 2015, 41). User study was conducted, with trainees and instructors at the Haaga driving school in Helsinki, and carried out in Helsinki metropolitan area. Several methods were applied such as diary, web questionnaire and semi-structured interview. Study subjects were seven driving license candidates. The age distribution of the users was between 18 and 25 years old. In addition, five instructors were also involved as potential users of the mobile learning application, and administrator tool concept design. The age distribution of the instructors was between 20 and 55 years old.

Web questionnaire and diary were utilized to learn about our users' daily activities, types of smart gadget in use, most frequently used mobile application or most often downloaded, and the level of computer knowledge. Individual semi-structured interview were then scheduled, to improve knowledge of study subjects and their smart devices usage habits. The interview sessions took on the average between 20 and 30 minutes.

The outcome of the user study was primary data, to be later on sorted out, in accordance with the mLUX framework (Dirin & Nieminen 2015, 42).

6.2 Data analysis

Raw data resulting from the user study was analyzed relying upon various data analysis methods, such as transcript coding and affinity diagram to explore the users' real needs and requirements for the target application. The result was a prioritized list of requirements.

Overall findings highlighted the following main patterns:

- Smartphones are intensively used by all study subjects, and are the preferred mobile device overtaking tablets and hybrid solutions.
- The application must support several kind of smartphones (e.g. entry-level vs. high end-devices, old models vs. most recent releases). Moreover, four different OSs and six web browser families were assessed out of seven users.
- The most user-desired features were in relation to practicing for theory tests, and support for driving sessions (e.g. GPS tracking).
- The application fun and easiness of usage were confirmed requirements, e.g. through the provisioning of multimedia content over mere written text.

6.3 Idea Creation

In accordance with the mLUX framework (Dirin & Nieminen 2015, 42), scenarios were designed presenting the found requirements as potential functions of the target application, and they were:

- A futuristic scenario, describing an application implementation based on users' requirements, and relying on future technology far more advanced than what is now available. A kind of science fiction-like landscape, offering all of the imaginable technological possibilities.
- A second scenario grounding app implementation on current state-of-the-art technology.
- A final one showing app implementation based on resources actually available to implement the prototype (e.g. financial, time, and knowledge related).

In the following figure a sample scenario of the target application is presented:

A bell rings in Tomi's mind, after submitting the self-evaluation feedback. Considering the driving session was not-so-positive, at least he could compensate with good scores in theoretical knowledge. Therefore he decides to take a theory test, using the same web application, while waiting for the bus to take him home. A specifically designed function is available, in the main website menu. ...

Figure 3. Scenario for theory test of driving m-learning application.

Scenarios were shared with six trainees. Three of them being among the participants to the user study, and the rest were newly involved subjects. The scenario related to the administrator and instructor tools were also shared with three driving school instructors.

Test users were asked to run the scenarios one by one, and were then interviewed. The collected feedback was analyzed, further grounding functional and non-functional requirements on users' contribution. Hence, providing additional value to the application idea.

6.4 Product concept

The co-created idea, including all of the requirements, was enabled the design of a paper-based prototype of the target application. The prototype was then further reviewed with the test users for final revisions of the concept, by executing a usability evaluation test. Testing was carried out at the media-lab in Haaga-Helia University of Applied Science. Through usability evaluation we assessed the functionality and the user interface of the target mobile learning application.

Finally the *m-learning application for driving school* was designed and proposed for implementation. In all processes of the application design and development the user experience factors such entertainment; delightfulness and adjustability of the learning system were in the center of the design theme (Dirin and Nieminen, 2013; Dirin & Nieminen 2015, 42, 43).

6.5 Need for adaptability and context awareness

The need for context awareness implementation and adaptability emerged immediately during the user studies and later phases. In fact, firstly the application needed to perform homogeneously on a wide variety of smart devices available on the market, e.g. operating systems, settings configuration, screen resolutions, supported technologies (Casarini & al. 2012, 8-9, 22). Secondly, awareness of user behavior, positioning in time and space were user requirements, and in accordance with Dirin & Nieminen's (2015, 41) methodology. Thirdly, data and presentation had to target specific user needs and capabilities, to maximize efficiency and efficacy of the learning process (Dirin & Nieminen 2015, 41; Aroyo & al. 2006, 7; Gilliot & Garlatti 2009, 2; Jongyi, Eui-Ho, Junyoung & SuYeon 2009, 7448). Fourthly, not only learning application adaptability has been a recurring trend (Billsus et al, 2002, 35), but even more importantly context awareness has proved to be particularly val-

uable, if not a necessity, for interaction based on handheld devices (Brusilovski & Maybury 2002, 22). In short, it was necessary to escape the one-size-fits-all approach of traditional learning platforms.

7 Implementation

7.1 Conceptual system architecture

The first challenge to be addressed, during user studies, was to select the most adequate smartphone platform for prototype implementation. There are many smartphones in the market, having different development and usability requirements. The most practical conclusion, considering project peculiarities, was to develop a cross platform application, due to the impossibility to anticipate the potential users' smart devices requirements (Casarini & al. 2012, 8-9, 22).

HTML5, JQuery, and derived libraries were selected as the most appropriate client-side technologies to develop the application, supporting all the latest smartphones and web browser families (Dirin et.al, 2013). In fact, HTML5 is intrinsically cross-platform, and provides consistent UI across the latest smart devices. Moreover, it can be paired with client side libraries available in large numbers (e.g. JavaScript and JQuery based) to code dynamic user interfaces, and provide rich user experience. Moreover, it was ensured that the selected HTML5 API's were supported by the most common browser families, hence the latest smartphones. Back-end implementation, server and development environment were based on Microsoft Technologies, such as ASP.NET MVC4, SQL Server 2012, SQL Server management Studio, Visual Studio 2013, as per project requirements (e.g. project sponsor existing IT infrastructure).

System design evaluations (e.g. multi-OS support) led to the engineering of an adaptive context-aware hypermedia system, based on HTML5 standards. In fact, web browser runtime provides a cross-platform environment. Furthermore, it was possible to determine a list of supported HTML5 features and APIs, including Geolocation, client side libraries, as well as third party openly available resources enabling fulfillment of requirements. Figure 4 presents the m-learning application for driving school's architecture. It exposes architectural components providing key system functionalities, while black-boxing elements not strictly necessary to acquire understanding of the application architectural flow.

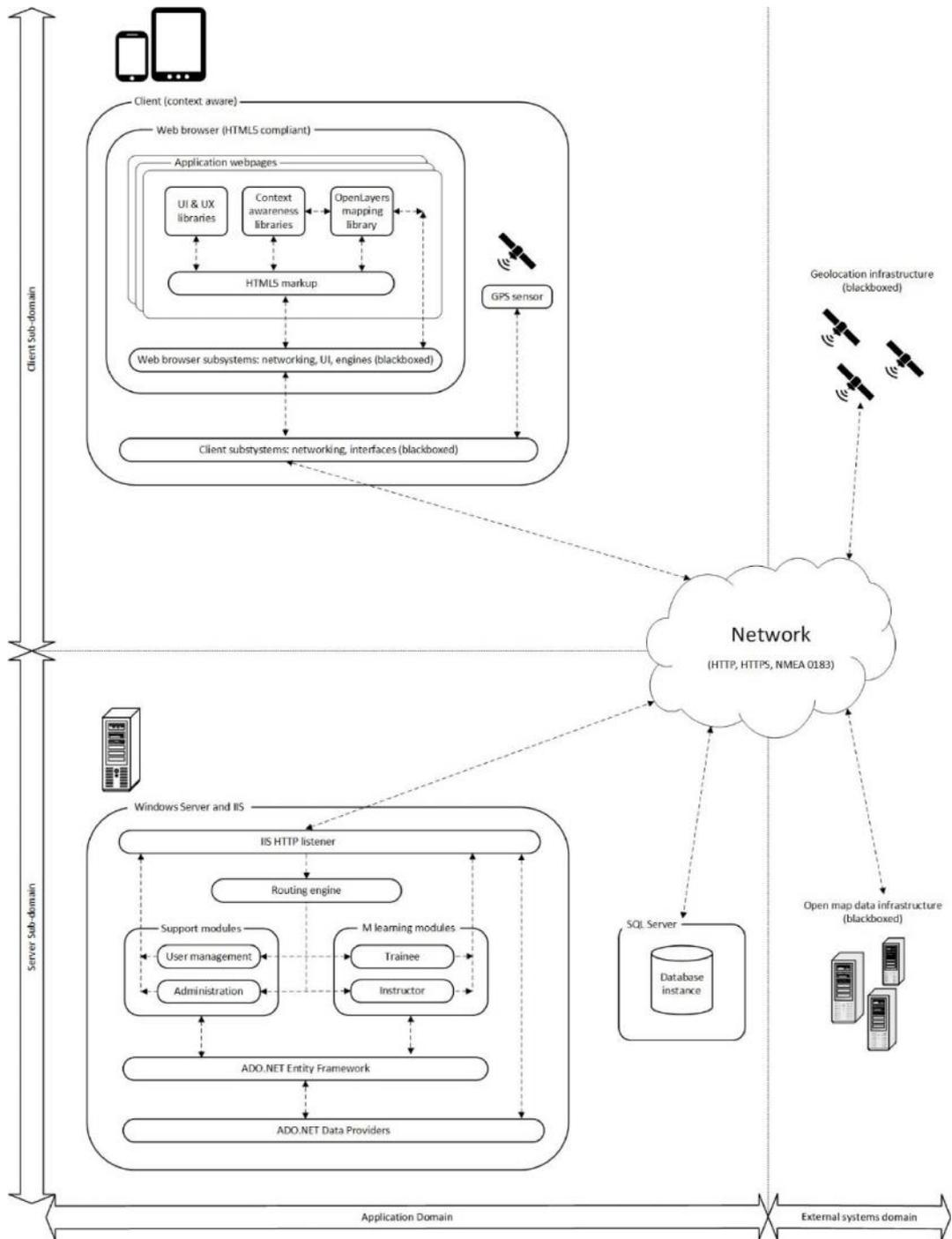


Figure 4. Architecture of the mobile learning application for driving school.

System architecture is logically organized into two areas of competence, the *application* and *external systems* domains. The former is further divided into *server* and *client* sub-domains.

External systems consistently provide fundamental resources not implementable within the application domain, due to cost and infrastructural constraints. In fact, it is a must for modern learning systems to access shared and reusable resources (Aroyo et.al 2006, 5), and achieve content critical mass, in order to adapt dynamically to context and users

(Balacheff 2006, 3). External system domain allows context awareness in relation to specific user location in time, via GPS tracking, and all of the tools needed for geospatial data management (e.g. standardization, retrieval, manipulation, and mapping).

The **client subdomain** is mainly constituted by web browser runtime environment, relying on subsystems at smart device kernel level for hardware interfacing, e.g. networking, UI rendering, GPS sensor management (Grosskurth & Godfrey 2003, 5). It is responsible for presentation adaptation (e.g. fluid layout), hence to enable homogeneous and consistent user experience, and user-related spatial context-awareness (e.g. geolocation). All of the client side adaptation features rely on HTML5 standards, JavaScript or JQuery based open and application defined libraries, for presentation adaptation and UX tuning. Furthermore, it is the application delivery agent, enabling user-application dialogue, and acts as a communication node between external systems and server side infrastructures, according to modalities defined and delivered by the latter.

The **server subdomain** is the application core, where system adaptation and context-awareness are defined and persisted, in relation to all of the five complementary models described by (Aroyo et.al 2006, 8-11): domain, user, context, instructional, and adaptation. Its main components are: web server (IIS), database server (SQL Server), and application server (Windows Server). The first handles HTTP/HTTPS requests between client, application and database server. The second provides data persistence and management capabilities, including user and context model. The latter defines and enforces all of the application logic within four core modules, including implementation and management of domain, instruction and adaptation models (Aroyo et.al 2006, 8-11).

Application server **outer subsystems**, provide technological support (e.g. .NET platform; Freeman 2012, 11), and interconnectivity. In fact, communication with the database server is ensured via the *ADO.NET* data providers, while the *Entity Framework* grants ORM implementation, matching relational database with object oriented paradigm of the executable code, run in the application server (Chadwick, Snyder & Panda 2012, 60; Freeman 2012, 97; Galloway, Haack, Wilson & Allen 2012, 3, 76). The routing engine redirects incoming requests from IIS web server to core modules (Chadwick & al. 2012, 15-16), according to standardized and highly customizable rules (Freeman 2012, 11). Moreover, routing engine configuration enables implementation of REST architecture, the most widespread for application interoperability over HTTP, actually overtaking SOAP (Freeman 2012, 6). Figure 5 shows RESTful routing configuration, via ASP.NET MVC routing engine, as implemented in the target system. In this project context, configuration required

coding in the *Global.asax.cs* standard class, first by defining routing rules in the *RegisterRoutes* method, and then loading them at application start using the *Application_Start* method (Freeman 2012, 193-194; Galloway & al. 2012, 224-225).

```
namespace DrivingSchoolApplication
{
    .....

    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            // Route name
            "Default",
            // URL with parameters
            "{controller}/{action}/{id}",
            // Parameter defaults
            new
            { controller = "Trainees", action = "Performance",
              id = UrlParameter.Optional
            }
        );

        routes.MapRoute(
            // Route name
            "DefaultForTrainees",
            // URL with parameters
            "{controller}/{action}",
            // Parameter defaults
            new
            { controller = "Trainees", action = "Performance",
              id = UrlParameter.Optional
            }
        );
    }

    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        // Bind custom model validators
        ModelBinders.Binders.Add(typeof(TheoryTestQuestions),
            new TheoryTestQuestionsValidatingModelBinder());

        RegisterGlobalFilters(GlobalFilters.Filters);
        RegisterRoutes(RouteTable.Routes);
    }
}
```

Figure 5. Implementation of RESTful routing.

Inner subsystems, in the application server, are the four core modules, implementing support and main m-learning functionalities. They receive requests mapped via routing engine, enforce application logic, interact with the database instance, and then address outgoing responses to IIS HTTP handler.

The support modules are user management and administration. The former provides overall management to uniquely identify end-users, and defines the user model (e.g. settings, preferences, learning objectives). The latter enables system administrations, such as role-user coupling to ensure role based resource access. The leftovers modules define, enforce, and track the learning process itself, in relation to its key participants: instructors and trainees. They are herein overviewed.

7.2 Content-level adaptation

The actual content of the learning materials are provided from the server by application query. The application assesses the users' knowledge, based on the previous learning activities. The content is then adaptively provided by the application to students, in two different circumstances, as per the *presentation adaptation* described by Brusilovski (2001, 97), and in relation to the main trainee module functionalities:

- Contextually adapted content suggestion in the **trainee tool main menu**, along with menu entries, e.g. theory test, self-evaluation form submission with and without GPS integration. In sort, the training features available to trainees.
- Adapted content as outcome of **theory test** taken by a trainees. Both theory test and result content are contextually user-targeted and categorized by study field (e.g. road signals, speed limit or administrative road regulations), according to subject performance.
- Adapted content immediately after reporting a **driving session self-assessment** form to instructors, be it inclusive or exclusive of GPS tracking function. Content is adaptive with similar modalities, as per the theory test case. Nonetheless, system adaptively differentiates between the two functionalities when monitoring, reporting, or enabling instructors to define learning outcome at user level (*user model*, according to Aroyo et al. 2006, 9).

Content adaptation is enforced via a chain of interactions between architectural domains, accessing resources persisted either in the database server or external domains (e.g.

open map data infrastructure). The retrieval algorithm implements a four-step logic, described in the following list:

- At first the most relevant topic category (e.g. subject) is determined, in accordance to user performance, and learning outcome target level set by instructors. Learning objectives are connected to user-related settings (directly and indirectly), e.g. user identity and driving license type, according to a cascading logic. In fact, user specific setting override driving license type settings, being finer grained (appendix 1).
- Then, a list of relevant multimedia resources are fetched, belonging to the selected highest priority category, if any, or uncategorized otherwise (appendix 2). It is noteworthy to notice how categories, multimedia files, and their correlation can be set by instructors via the administrator interface. Hence, determining application adaptation outcome.
- Moreover, each and every multimedia resources is contextually prioritized, according to user-related settings, e.g. direct relation to the user identity or derived attributes.
- Finally, multimedia resources are sorted out according to priority, and formatted to match UI specific data model (appendix 3), for further client side manipulation. Hence, presentation layer adaptation.

The algorithm implementation relies on LINQ (Freeman 2012, 97), Entity Framework (Chadwick et al. 2012, 60), and .NET classes for data retrieval, in memory manipulation, and delivery to the client domain. The same overall four-phased logic is enforced in the retrieval of content in response to theory test completion, but involves different context dependent variables.

7.3 Trainee module

With the help of this application the driving license candidates are able to study, complete and pass the compulsory driving school theory test. Additionally driving school instructors receive reports about students' practical driving session performance.

Reports consist of students' self-evaluations on the driving experience along with other data such as GPS tracks, durations of the driving practice and driven route, taken by students while practicing driving session. All of these features are mirrored in the two main server inner domains, e.g. monitoring for instructors, execution and reporting for trainees.

The application UI is consistent, interactive and supports touch and non-touch smart devices. The application provides content to users in various formats depending on querying context. Nonetheless, the application supports video, audio, animation and text-based formats.

7.3.1 Trainee main menu

The trainee main menu provides entries to trigger task execution, among those available to trainee. Moreover, it implements two adaptation features. The first is the provisioning of contextually selected learning material, along with menu entries. The second is to suggest which task the trainee should execute, according to task performance history, and already executed tasks in the previous part of the day. In fact, only one driving session per day is expected. The following is a screenshot of the trainee module main menu (figure 6). Contextually provided adaptive content can be found on top of the screen. The suggested feature is highlighted by a thick white trimming.

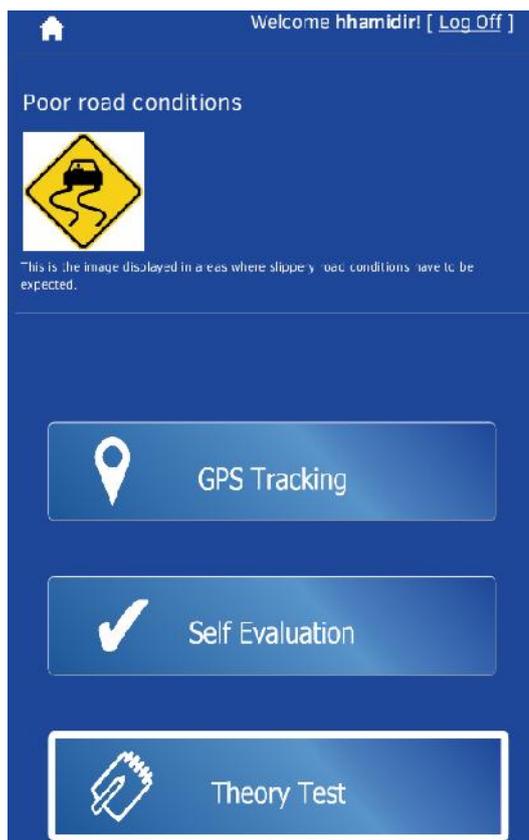


Figure 6. A screenshot of the trainee module main menu.

7.3.2 Driving Session self-evaluation feature

Driving session is the practical activity supported by the application. It can be used in two functional modalities:

- Correlated GPS tracking and self-evaluation form submission, in case of application usage during driving activities.
- Self-evaluation form submission only, with no GPS integration, in case of late reporting e.g. no mobile device available during driving session.

Flow of the application self-evaluation form submission, required in both functional feature implementations, is modeled in the following sequence state diagram. Figure 7 refers the implementation without GPS integration.

driving school

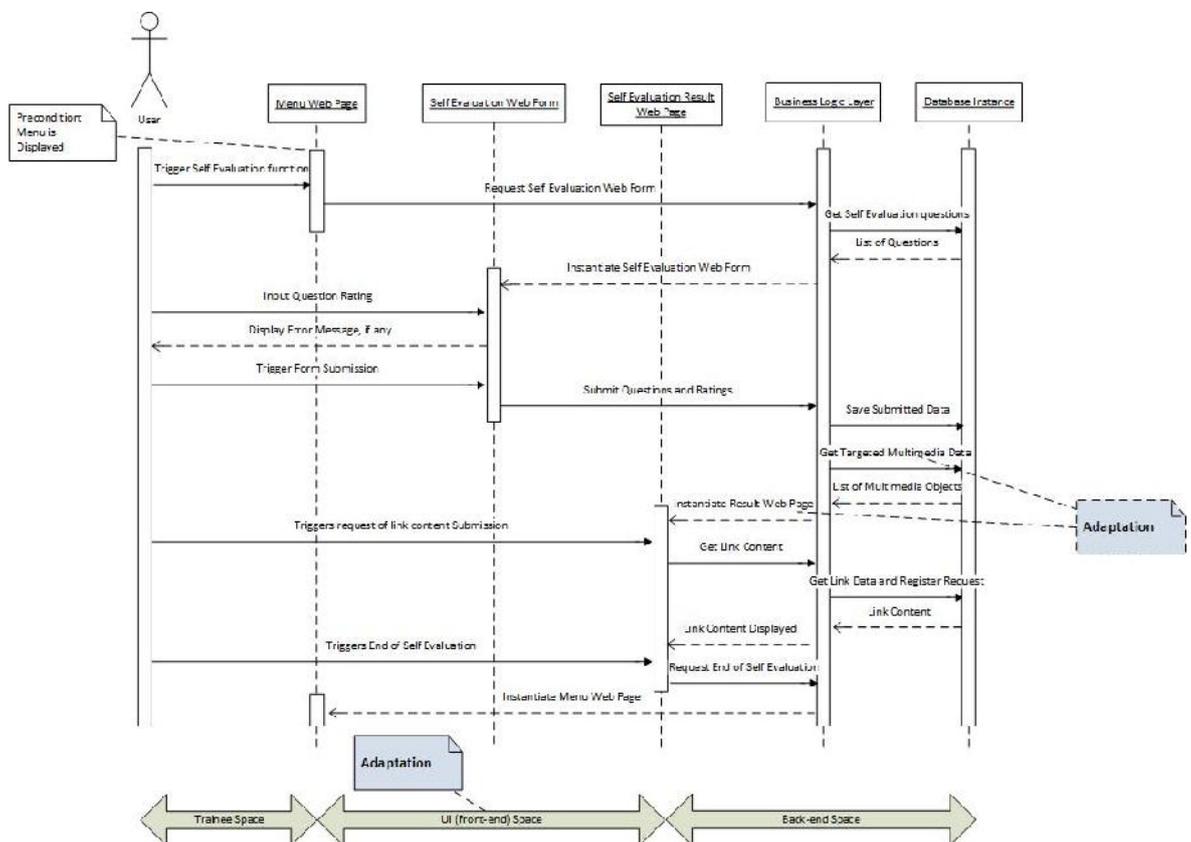


Figure 7 Sequence diagram of m-learning driving application, Self-evaluation concept.

The content of the report form and the questions are customized by the driving school instructors. The content of the report form personalized by instructors based on students' theory achievements and previous driving experience reports. The application reacts in-

stantly on the reported form after driving session. If severe errors occur during driving sessions the application warn the student about the seriousness and prompt a proper content for further studies.

The application UI is customized both for touch and n-touch devices. Instructors can determine the available value range for each and every feedback form question. Furthermore, they can organize them by creating specific categories, and assign feedback form entries to categories. Finally, the UI layer dynamically converts the evaluation form into a wizard (e.g. step by step procedure), using a client-side JavaScript library. In fact, the following screenshot (figure 8) shows the driving session self-evaluations application partial form, e.g. showing one of the wizard-like steps. The entries category, for the displayed step, is visible on the upper UI part, right under session date.

★ Welcome hhamidir! [Log Off]

Driving Session Self Evaluation

Session Date: 7.12.2012 4:18

Vaikeat olosuhteet

En ole ajanut pimeällä?

0 7 0

Liukas keli ei aiheuta minulle mitään vaikeuksia?

0 7 6

Vesisateella ei tarvitse vähentää nopeutta?

0 7 0

< Back

Submit

Figure 8. Self-evaluation form after the driving session

7.3.3 Theory test feature

The theory test training features is the most desired theoretical support activity, according to the conducted user study. Each theory sessions consist of an evaluation test. Passing the evaluation test is prerequisite for starting a new one, hence potentially practice different topics.

In fact, users are unable to select them as they are contextually generated and adaptively provided by the application itself. In case the number of failed answers exceed an instructor set learning objective (e.g. percentage ratio, expressing user-specific target performance level), the application discards the lesson and recommends the trainees to re-take the lesson. The application actually generates a new theory test, focused on the topics related to the poor user performance, and removes the failed test. In fact, it is archived to feed historical data, becoming unavailable to students, but visible as an historical record to the instructors. Moreover, it generates user-performance statistical data, enabling system's user-specific adaptability, in relation to both overall performance and topic specific issues.

Trainees are entitled to suspend the self-evaluation test at any time, in order to exploit in the most agile way their time (e.g. reuse small chunks of time otherwise wasted, such as short commuting trips or inactivity time frames). The application however keeps track of the user's readings and self-evaluation suspended activities. As a result users may continue the reading tasks or self-evaluation test in their convenient time. Furthermore, the application will not provide a new test until the pending one is executed. This feature forbids trainees from skipping undesired tests. Hence, bypass instructor designed content and challenges.

Figure 9 displays a screenshot, for one of the questions part of a user-specific contextually generated theory test. Along with the question and possible answers, available response time and test progress (e.g. question X of Y, where X is the current question and Y the total test questions), are visible on the upper part of the user interface. The user-selected answer is highlighted, with a thick white trimming. The provided answer was correct. Hence, answer background is green colored, to provide a successful task feedback to the user.

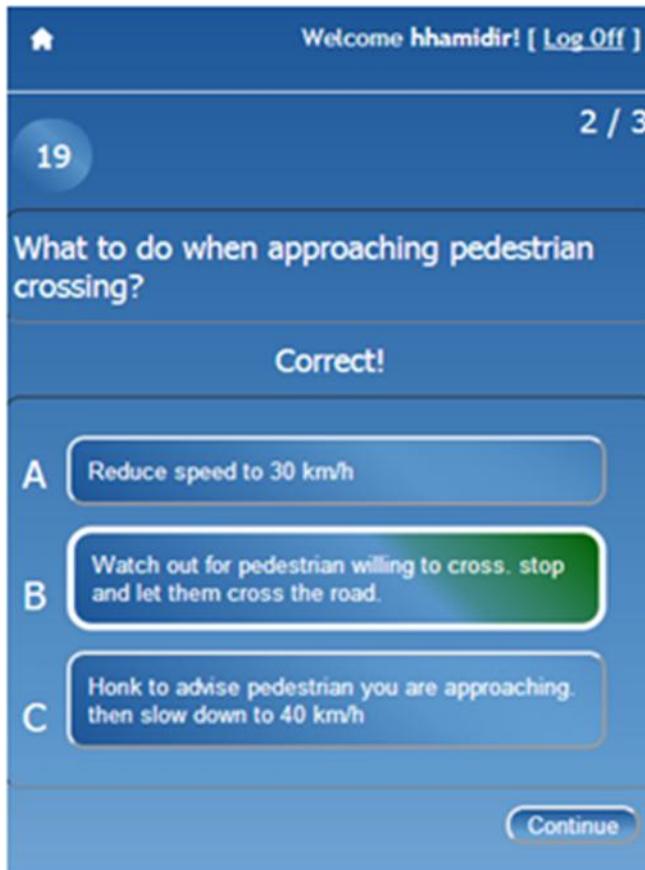


Figure 9. Theory test screenshot.

7.4 Instructor and administrator tools

The administrative and instructor tools are the most important part of the driving license application. With the help of the former instructors define the application content, self-evaluation test content and criteria for passing theory lessons. Furthermore, the instructor tool provides a unique report to the instructors about the students' performance on the theory part and in the practical driving sessions. Both tools are compatible with mobile and PCs, because instructors must be enabled to operate bot on the move using a mobile device, or in office working on a non-mobile device. Similar to the self-evaluation question UI, we have used color coding for instructions to check the report of each student instantly. The instructors are able to print out the students details learning history and driving session progress summary reports. The following screenshot (figure 10) presents the instructor tool main page.

The screenshot shows the 'MLearning' interface with a 'Trainees Performance' section. It contains a table with the following data:

Firstname	Lastname	Registered	Type	Exam Date	Performance	License	Driving Sessions Done	Theory Tests Done	Issues	Actions
Amir	Dirin	31.10.2012	Senior		<div style="width: 50%;"></div>		54 (30)	19 (50)	⚠️ ⚠️ ⚠️	🚗 📄
andrea	viisi	4.11.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
august	lammin	4.11.2012	Senior		<div style="width: 0%;"></div>		0 (20)	0 (40)		🚗 📄
david	l23	4.11.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
dec	winter	4.11.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
Dude	Guy	14.03.2015	Senior		<div style="width: 0%;"></div>		0 (20)	0 (40)		🚗 📄
Instruttore	Gulda	31.03.2015	Instructor		<div style="width: 50%;"></div>		23 (10)	8 (15)	⚠️	🚗 📄
Jauko	Kalliola	31.10.2012	Senior		<div style="width: 0%;"></div>		0 (20)	0 (40)		🚗 📄
Juhy	varinen	4.11.2012	Senior		<div style="width: 0%;"></div>		0 (20)	0 (40)		🚗 📄
Juho	suomaa	4.11.2012	Senior		<div style="width: 0%;"></div>		0 (20)	0 (40)		🚗 📄
Jussi	Virtanen	31.10.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
mark	second	4.11.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
Maurizio	Casarini	4.11.2012	Junior		<div style="width: 0%;"></div>		0 (30)	0 (50)		🚗 📄
Maurizio	Test	25.03.2015	Admin		<div style="width: 0%;"></div>		0 (10)	0 (15)		🚗 📄

Figure 10. Instructor tool's main page screenshot.

Furthermore, the system enables the instructor to manipulate user-specific information, by setting learning targets and access historical data. The former functionality is implemented by user's training target form. It allows setting of the desired amount and minimum accepted performance index, in relation to both theory test and driving sessions (figure 11).

The screenshot shows the 'Amir Dirin's training target' form. It includes the following sections:

- Driving Sessions Amount:**
 - By Senior trainee: 20
 - User specific:
- Driving Target Performance:**
 - By Senior trainee:
 - User specific:
- Theory Tests Amount:**
 - By Senior trainee: 40
 - User specific:
- Theory Target Performance:**
 - By Senior trainee:
 - User specific:

Buttons: [Back to List](#)

Figure 11. Screenshot of user's training target level.

Historical driving data is displayed in a specifically designed webpage, listing all of the sessions taken by the user. Overall information is provided (e.g. session execution date and time, time and distance driven). Moreover, instructors are entitled to display both the submitted self-evaluation (figure 12) and GPS tracking data (figure 13). The interactive map shows the driven path, and it is implemented using OpenLayers JavaScript-based library.

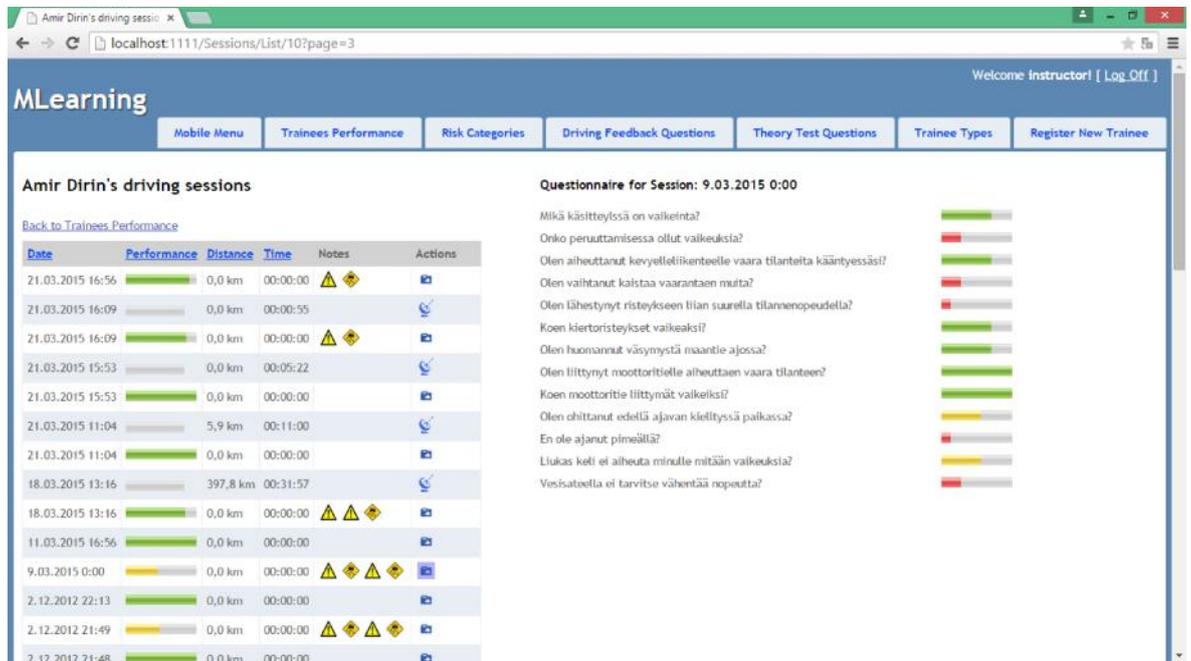


Figure 12. Screenshot of a driving session self-evaluation.

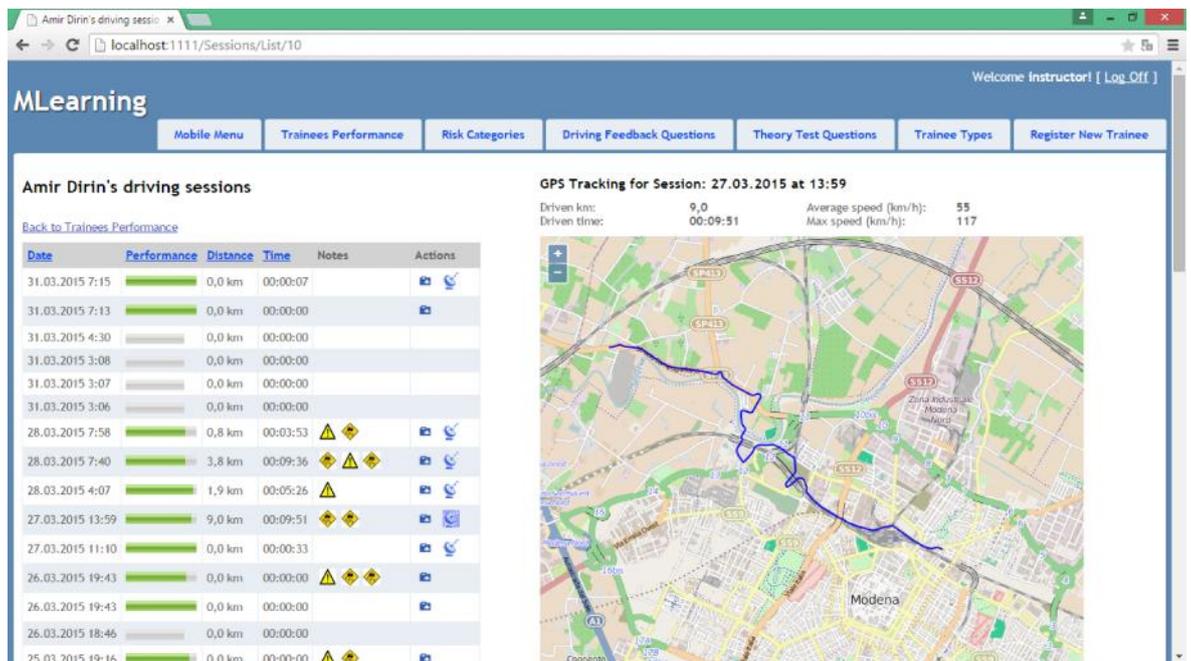


Figure 13. Driving session GPS data and interactive map.

7.5 Usability testing

This paragraph provides an overview of the application usability testing. More details can be found in the usability test plan, part of the thesis administration folder.

7.5.1 Test design

The goal of application testing was to assess the implemented functionalities, with particular focus on user experience and usability (Dirin and Nieminen 2015, 49). Accordingly, testing was usability centered. Objectives were, firstly to evaluate user interface (UI) consistency and efficacy, in relation to content readability and meaningfulness, efficient website navigation, reduced amount of actions required by human-application interaction, and lack of redundancy. Secondly, to collect users' qualitative feedback in relation to user experience (UX). Finally, to identify user-proposed UI and UX improvements, if any.

It was herein previously stated, how scenarios are particularly effective in m learning application design, because they provide users the same context they are accustomed to. Hence, ten testing scenarios were pre-design, emulating practical cases of application usage. Moreover, six driving license candidates were selected as test subjects, having age distribution between 19 and 25 years. A sample of the designed scenarios can be found in figure 14.

You are at school, and the hall is nearly empty: you were very early today. None of your schoolmates are here. Accordingly, you decide not to waste time and to practice theory, to get your first driving license. Luckily, you have access to the driving school mobile training website.

Accordingly, you browse the Web with your smartphone, access the driving school website home page, and log on with the given credentials. The system immediately displays the main menu, and you start the theory test function.

You read and answer the given questions, according to your knowledge of the driving regulations. **After the second answer**, you are approached by a friend and decide to chat a little bit. Hence, you navigate to the homepage, interrupting the test, but do not log out from the application. You will continue home, in the afternoon.

Figure 14. An example of the designed usability testing scenario.

Quantitative evaluation metrics were pre-defined, such as scenario completion rate and time, displayed number of pages and interactive element accessed. Accordingly, artefacts for data collection were designed, e.g. tables and forms. Moreover, it was planned to retrieve users' feedback via post-scenario structured questionnaires, in relation to application flow, UI manipulation elements, content quality, and overall UX (Figure 15).

<p>Rating scale: 1 = very poor, 2 = poor, 3 = average, 4 = good, 5 = very good</p> <p>Navigation related:</p> <ul style="list-style-type: none">- How easy was it to understand how to start a scenario, e.g. main menu buttons?- Were navigation controls easily visible, e.g. buttons and links?- How easy it was to understand how to proceed to next task/webpage?- How easy was it to understand if you could navigate back?- How easy it was to navigate back to the main menu page? <p>Controls related:</p> <ul style="list-style-type: none">- How easy it was to notice interaction elements in the webpages (e.g. buttons, links, calendar icon for date selection, sliders)?- How easy it was to understand the clickable element function?
--

Figure 15. A sample of the post scenario questionnaire.

In order to ensure adequate testing coverage and improve data quality, each and every scenario was executed by several users, and each and every user tested several scenarios. Moreover, the BYOD (bring your own device) policy was applied, to ensure test devices represented a reliable sample of hardware commonly available to the application target audience.

7.5.2 Test result

Test result produced raw data in two separate batches: from quantitative evaluation metrics (Appendix 4), and from post-scenario questionnaires. The latter is not herein included due to conciseness reasons. Nonetheless, it can be found in the *Usability Test Plan*, included in the thesis administration folder.

According to testing result, 78% of the scenarios were successfully completed, 11% completed but partially successful, and 11% was not completed (Appendix 6). The key issues

in scenarios completion were in relation to users' different cognitive ability, particularly on timed tasks (e.g. theory tests).

Overall scenario execution time ranged between 1' 20" and 7' 3", the most consistent variations depending on scenario complexity (Appendix 7). Minor timeframe differences were found in relation to each and every specific scenario, in the range of 5-25% for simple tasks, and 36-58% for complex scenarios (Appendix 7). Furthermore, they were in relation to user-specific capabilities, and hardware constraints. In fact, screen size variation was so consistent between devices to hinder content readability in some cases, despite UI adaptation (e.g. automatic element size scaling).

Post-scenario feedback delivered an overall appreciation of the application, in all of the main evaluation areas, by most users: 78% for application flow, 64% for interaction features, 74% for content, and 66% for overall likeability (Appendix 5). Most of the feedbacks were in relation to screen size constraints (Appendix 8). Furthermore, it was confirmed necessity to focus on UI and UX quality, in order to meet users' expectations. In fact, according to test subjects' feedbacks (appendix 8) they were respectively referred to with an occurrence rate of 78% (UI) and 100% (UX). Moreover, it was possible to assess the most critical application adaptability triggers (table 2) and features (table 3).

Table 2. Occurrence of adaptation triggers, according to test subjects' feedback.

Adaptation trigger	Occurrence
Screen size (smart device hardware constraint)	78%
User cognitive abilities	22%

Table 3. Occurrence of adaptation target/features, according to test subjects' feedback.

Adaptation target	Occurrence
UI layout	78%
UI element size	78%
Delivered content (in relation to user capabilities and preferences)	11%
Application behavior (in relation to user performance)	11%

Such a high users' request for UI and UX performance, hence usability, confirms validity of selecting the mLUX framework (Dirin & Nieminen 2015), as project implementation

methodology, due to its focus on user interface and user experience. Accordingly, test results confirmed the importance of both key requirements and users expectations assessed during implementation of the mLUX framework. Moreover, they demonstrated the implemented application to be easily usable and agile, due to high scenarios completion rate and low execution time. Furthermore, application flow likeability was high. Accordingly, the m learning application for driving schools can be positively rated.

Key quantitative and qualitative performance aspects proved to be user-centered (e.g. individual differences in cognitive and learning capabilities, task response time), UI focused (e.g. easy to understand and interact with, adaptability to screen resolution), content related (e.g. contextually delivery, clean design, emotional appeal, adaptability to network performance), and resulting in an overall pleasant and effortless experience. Accordingly adaption and context awareness proved to be necessary characteristics of m learning applications, as shown in table 2 and table 3.

8 Conclusion

This case study relied on a User Centred Design (UCD) framework specific for m-learning application development. The project strived to implement and assess a mobile learning application for driving schools and driving licenses' candidates. The whole software development life cycle was grounded on the UCD principle, and mLUX framework for m-learning application development, centred on user experience and user interface usability. Both methodologies proved successful in ensuring application users and role-players direct involvement, throughout the whole process.

The methodological framework resulted in an application fulfilling all the usability criteria. The prototype evaluation report indicated that the application is easy to use and provides the essential learning materials for driving school candidates. Users are particularly satisfied, because driving theory lessons are accessible at anytime and anyplace. Additionally, the UCD framework for mobile learning application helped to develop a mobile learning adaptive, context aware, interactive and easy to use application. The usability reports show that these features in mobile learning application provides positive user experience for test users.

The UCD framework resulted in this case study and dynamic application. System content and UI is customizable based on students' performance, preferences and instructor determined parameters, on theory lessons evaluation and practical driving session reports. Moreover, the application supports multi-formatted content. Students may select one or combinations of many formats based on the context and the content of the learning materials. The main focus of this project was to assess the UCD application development frameworks' efficiency, in order to deliver a fully functional prototype implementing adaptation and context awareness, according to requirements collected during user studies. The pedagogical considerations or evaluations were out of the scope of this study.

9 Challenges and possible future work

The challenges met during project executions were many, first and foremost the implementation of a highly dynamic application, with consistent requirements in terms of user experience and UI usability. Moreover, injecting adaptation and context awareness proved to be a complex task, requiring solid product vision, a considerable amount of different technologies at all system layers (front-end, back-end, database design), and IT expertise at all levels (e.g. concept design, development and testing). Moreover it required extensive background study, due to the limited author knowledge and experience in the key study fields upon which the project had to be grounded (e.g. UCD, application adaptability and context awareness). Moreover, despite all the project challenges, limited resources were available for implementation.

Finally, it would be valuable to further explore the increment of user experience and usability features, required by modern dynamic and rich modern web application. Among them the possibility to completely dynamically generate UI layout and outlook, due to the extremely high importance of usability in users' satisfaction, and to increase the role of users in determining application physiology and behaviour. The ultimate goal would be to further explore the plethora of adaptability and context awareness features theoretically implementable in m-learning applications, eventually adding pedagogical considerations and evaluations.

References

Abras, C., Maloney-Krichmar, D., Preece, J. 2004. User-Centered Design. In Bainbridge, W. S. Encyclopaedia of Human-Computer Interaction. Sage Publications. Thousand Oaks, California. USA.

Aroyo, L., Dolog, P., Houben, G., Kravcic, M., Naeve, A., Nilsson, M. & Wild, F. 2006. Interoperability in Personalized Adaptive Learning. Educational Technology and Society, 9, 2, pp. 4-18.

Balacheff, N. March 2006. 10 issues to think about the future of research on TEL. Les Cahiers Leibniz, 147.

Billsus, D., Brunk, C. A., Evans, C., Gladish, B., & Pazzani, M. 2002. Adaptive interfaces for ubiquitous Web access. Communications of the ACM 45, 5, pp. 34-38.

Bossert, J.L. 1991. Quality Functional Deployment: A Practitioner's Approach. ASQC Quality Press. Milwaukee, Wisconsin. USA.

Brusilovski, P. 2001. Adaptive hypermedia. In User modelling and user-adapted interaction, 11, pp. 87-110.

Brusilovski, P., & Maybury, M. 2002. From adaptive hypermedia to the adaptive web. Communications of the ACM, 45, 5, pp. 21-24.

Casarini, M., Chawla, A., Rahman, S. & Rishling, Z. 2012. Usability. Needs and user analysis. Case: m-learning driving licenses application. Haaga Helia University of Applied Sciences. Helsinki.

Casarini, M. 10 Oct 2012. Haagan Autokoulu Töölö offices. Visit: Topeliuksenkatu 1, Helsinki. Finland.

Chadwick, J, Snyder, T. & Panda, H. 2012. Programming ASP.NET MVC4. O'Reilly Media. Sebastopol, California. USA.

Corlett, D., & Sharples, M. 2004. Tablet technology for informal collaboration in higher education. In J. Attewell & C. Savill-Smith (Eds.) Mobile learning anytime everywhere: Papers from Mlearn 2004 (pp 59-61). London. UK.

Dirin, A., Nieminen, M. & Kettunen, M. 2013. Student capabilities to utilize m-learning service in new smart devices. International Conference on Advanced ICT for Education (ICAICTE 2013) September 20-22, 2013 Hainan, China.

Dirin, A., & Nieminen, M. (2015). mLUX: usability and user experience development framework for m-learning. *iJIM, International Journal of Interactive Mobile Technologies*, 9, 3, pp. 37-51.

Dolog, P. 2008. Designing adaptive web applications. *SOFSEM 2008: Theory and Practice of Computer Science. Lectures notes on computer science*, 4910, pp. 23-33.

Freeman, A. 2012. *Pro ASP.NET MVC4*. 4th ed. Apress Media LLC, New York. USA.

Galloway, J., Haack, P., Wilson, B. & Allen K.S. 2012. *Professional ASP.NET MVC4*. John Wiley & Sons, Inc.. Indianapolis, IN 46256. USA.

Gartner Inc. 23 Oct 2012. Gartner Identifies the Top 10 Strategic Technology Trends for 2013. URL: <http://www.gartner.com/newsroom/id/2209615>. Accessed: 28 May 2013.

Gilliot, J. & Garlatti, S. 2009. An Adaptive and Context-Aware Architecture for Future Pervasive Learning Environments. Computer Science Department, TELECOM Bretagne. Cedex. France.

Google Inc. May 2012. Our mobile planet: Finland. URL: <http://www.thinkwith-google.com/insights/emea/library/studies/our-mobile-planet-Finland>. Accessed: 28 May 2013.

Google Inc. May 2015. Building for the next moment. URL: <http://adwords.blogspot.it/2015/05/building-for-next-moment.html>. Accessed: 19 September 2015.

Gould, J.D., Lewis, C. 1985. Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM*, 28, 3, pp. 300-311

Gould, J.D., Boies, S.J., & Ukelson, J. 1997. How to design usable systems. In Helander, M.G., Landauer, T.K. & Prasad, P.V. (editors). *Handbook of Human-Computer Interaction*, pp 231-255. Elsevier Science B.V. Amsterdam, Netherlands.

- Grosskurth, A. & Godfrey, M.W. 2003. A reference architecture for web browsers. *Journal of software maintenance and evolution: research and practice*, 00:1-7, pp. 1-19.
- Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. and Cajander, Å. 2003. Key principles for user-centred system design. *Behaviour and Information Technology*, 22, 6, pp. 397-409.
- Haagan autokoulu. Autokoulu Helsinki, Aina jonon kärjessä. URL: <http://www.haaganautokoulu.fi/>. Accessed: 27 May 2013.
- Haagan autokoulu 2011. Company Overview. URL: <https://www.facebook.com/pages/Haagan-autokoulu/222815791078207?id=222815791078207&sk=info>. Accessed: 27 May 2013.
- Jongyi, H., Eui-Ho, S., Junyoung, K. & SuYeon K. 2009. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36, pp. 7748-7757.
- Jäppinen, A., Ahonen, M., Vainio, T., & Tanhua-Piironen, E. 2005. Adaptive Mobile Learning Systems: The essential issues from the design perspective. In Attewell, J. & Savill-Smith C. (editors) *Mobile Learning Anytime Everywhere. A book of papers from MLEARN 2004*, pp. 109-111. Learning and Skills Development Agency. London. UK.
- Keinonen, T. 2003. Introduction: Mobile distinctions. In C. Lindholm, T. Keinonen. And H. Kiljander (Eds) *Mobile Usability: how Nokia Changed the face of the mobile phone*, pp. 1-8. McGraw-Hill, New York. USA.
- Kukulka-Hulme, A. 2007. Mobile Usability in Educational Contexts: What have we learnt? *International Review of Research in Open and Distance Learning*, Volume 8.
- Maged, N.K.B., Wheeler, S., Tavares, C., & Jones, R. 2011. How smartphones are changing the face of mobile and participatory healthcare an overview, with example from eCAA-LYX. *BioMedical Engineering Online*. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3080339/pdf/1475-925X-10-24.pdf>. Accessed: 18 June 2015.
- Mostakhdemin-Hosseini, A. 2008. Analysis of Pedagogical Considerations of M-learning in smart devices. *International Journal of Interactive Mobile Technologies (IJIM)* Volume 3, Special Issue 1.

Mostakhdemin-Hosseini, A. 2009. Usability Considerations of Mobile Learning Applications. *International Journal of Interactive Mobile Technologies (iJIM)* Volume 3, Special Issue 1.

Nieminen, M. 2004. Information support for user-oriented development organization. Considerations based on the construction and evaluation of knowledge storage. Doctorate's thesis. Helsinki University of Technology, Department of Computer Sciences and Engineering. Espoo, Finland. URL: <http://lib.tkk.fi/Diss/2004/isbn951227308X/isbn951227308X.pdf>. Accessed: 30 August 2015.

Norman, D A. & Draper, S. W. 1986. *User-Centred System Design, New Perspective on Human-Computer Interaction*. Lawrence Erlbaum Associates Inc. Hillsdale. New Jersey. USA.

Rosson, M.B. & Carroll, J.M. 2002. *Usability engineering. Scenario-based development of human-computer interaction*. Morgan Kaufmann. San Francisco. USA

Seong, S. K. D. 2006. Usability guidelines for designing mobile learning portals. *Mobile 06 Proceedings of the 3rd international conference on Mobile technology Applications & Systems* ACM New York, NY, USA, 2006 ISBN: 1-59593-519-3. [http://nexus.hs-bremerhaven.de/Library.nsf/0946dbe6a3c341e8c12570860044165f/a6ab89074ac24f8bc1257394003df005/\\$FILE/a25-seong.pdf](http://nexus.hs-bremerhaven.de/Library.nsf/0946dbe6a3c341e8c12570860044165f/a6ab89074ac24f8bc1257394003df005/$FILE/a25-seong.pdf).

Sharp, H., Rogers, Y. & Preece, J. 2006. *Interaction design: beyond human-computer interaction*. John Wiley & Sons Inc. New York. USA

Sharples, M., Beale, R., Hoppe, U., Issroff, K., Jones, A., Milrad, M., Scanlon, E., Taylor, J., Walker, K. & Winters, N. 2007. *Big Issues in Mobile Learning*. Report of a workshop by the Kaleidoscope Network of Excellence in Mobile Learning Initiative. Kaleidoscope. University of Nottingham, Nottingham. URL: <http://matchsz.inf.elte.hu/tt/docs/Sharples-20062.pdf>. Accessed: 17 May 2013.

Statista 2015. Number of free and paid mobile app store downloads worldwide from 2011 to 2017 (in billions). URL: <http://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>. Accessed: 19 September 2015.

The International Organization for Standardization 2010. ISO 9241-210. Ergonomics for human-system interaction. Part 210: human-centred design for interactive systems. ISO. Geneva. Switzerland.

The Standish Group 2014. The Standish Group report chaos. URL: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>. Accessed: 15 September 2015.

Tilastokeskus 7 Nov 2012. Two thirds of Finns are online shoppers. Prevalence of Internet usage and certain purposes of use in 2012. URL: http://www.tilastokeskus.fi/til/sutivi/2012/sutivi_2012_2012-11-07_tie_001_en.html. Accessed: 29 May 2013.

Trinder, J. Magill, J., & Roy, S. 2005. Expect the Unexpected: Practicalities and problems of PDA project. In A Kukulska-Hulme & J. Traxler (Eds.) *Mobile Learning: A handbook for educators and trainers* (pp.92-98). London. UK.

Usability Research Group. 2002. *User Centred Design Methods*. Indiana University. URL: http://www.indiana.edu/~usable/presentations/ucd_methods.pdf. Accessed: 1 February 2014.

Wallach, D. & Scholz, S.C. 2012. User-Centered Design: why and how to put users first in software development. In *Software for People. Management for professional series*, pp. 11-38. Springer-Verlag. Berlin. Germany.

Zhang, D. and Adipat, B. 2005. Challenges, methodologies, and issues in the usability testing of mobile application. *International Journal of Human-Computer Interaction*, 18, 3, pp 293-308. Taylor & Francis, Abingdon Oxfordshire. UK.

Appendices

Appendix 1. Adaptivity algorithm: determine the most relevant content category

```
public Categories getTopPriorityCategory(DrivingSchoolApplicationEntities db)
{
    try
    {
        int totalSubmittedRating = this.Questionnaires.Sum(q => (int)q.rating);
        Categories topCategory = null;

        // Rating submitted: top category from session questionnaires
        if (totalSubmittedRating > 0)
        {
            var prioritizedCat = this.Questionnaires
                .Where(q => q.rating > 0)
                .GroupBy(c => c.Questions.Categories)
                .Select(i => new { category = i.Key,
                                rating = i.Sum(c => c.rating)
                            })
                .OrderByDescending(i => i.rating).First();
            topCategory = prioritizedCat.category;
        }
        // Top Category from session issues
        else if (this.SessionIssues.Any())
        {
            var prioritizedCat = this.SessionIssues
                .Where(si => si.recurrenceCount > 0)
                .GroupBy(i => i.Categories)
                .Select(i => new { category = i.Key,
                                occurrence =
                                i.Sum(c => c.recurrenceCount)
                            })
                .OrderByDescending(i => i.occurrence).First();
            topCategory = prioritizedCat.category;
        }
        // if no rating in questionnaires and no issues then returns random
        else
        {
            // Select a random active category having multimedia resources.
            topCategory = Categories.getRandom(db);
        }
        return topCategory;
    }
    catch (Exception)
    {
        return null;
    }
}
```

A code snippet determining the most relevant multimedia resource category (e.g. subject) to be displayed in the user interface. The Entity Framework grants ORM capabilities for database access.

Appendix 2. Adaptivity algorithm: fetch a list of multimedia resources by category

```
public List<MultimediaResources>
    getMultimediaResourcesList(DrivingSchoolApplicationEntities db)
{
    try
    {
        // Get active multimedia training resources for selected category
        List<MultimediaResources> resources =
            db.CategoryMultimediaResources
                .Where(m => m.active == true &&
                    m.categoryID == this.categoryID)
                .Select(r => r.MultimediaResources)
                .ToList();

        // If no category specific resources are available => get one / type
        if (resources == null || !resources.Any())
        {
            resources = db.MultimediaResources
                .Where(m => m.active == true &&
                    m.MultimediaResourceTypes.active == true)
                .GroupBy(m => m.typeID)
                .SelectMany(r => r.OrderBy(m => m.typeID).Take(1))
                .ToList();
        }
        return resources;
    }
    catch (Exception)
    {
        return null;
    }
}
```

A code snippet fetching a list of multimedia reference objects from database, in relation to the previously selected subject category.

Appendix 3. Adaptive algorithm: sort multimedia objects and format for UI layer

```
public MultimediaResourcePrioritizedList(Sessions session,
                                         List<MultimediaResources> multimedias)
{
    try
    {
        int theoryIndex = (int)session.Trainees.TheoryTests
            .Average(t => t.performanceIndex);
        int drivingIndex = (int) session.performanceIndex;
        this.resources = new List<MultimediaResourcePrioritized>();
        if (multimedias != null)
        {
            foreach (var item in multimedias)
            {
                MultimediaResourcePrioritized r =
                    new MultimediaResourcePrioritized();
                r.resource = item;
                r.priority =
                    this.getMultimediaResourcePriorityByHtmlElement(
                        item.MultimediaResourceTypes.htmlElement,
                        theoryIndex, drivingIndex, session.Trainees
                    );
                this.resources.Add(r);
            }
            // Generic Sort() using the CompareTo method added to the class
            // MultimediaResourcePrioritized, implementing IComparable
            this.resources.Sort();
        }
    }
    catch (Exception)
    {
        this.resources = new List<MultimediaResourcePrioritized>();
    }
}
```

A code snippet sorting multimedia in-memory objects list, and formats data to match UI layer data manipulation needs.

Appendix 4. Detailed usability testing scenario results by test subject

Subject	Scenario	Successful (1)	Completion Time	Total webpages	Notes / Suggested improvements
A1	1	Y	3:19	Unavailable	<p>Issues:</p> <ul style="list-style-type: none"> – Calendar icon is too small to be interacted with. – Displayed link content is not clearly visible, out of the initial window frame <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Improve overall readability. – Increase interaction element size. – Increase form size.
	7	Y	4:00	Unavailable	<p>Issues:</p> <ul style="list-style-type: none"> – Major difficulties in selecting answer, e.g. answer button size. – The given answering time is insufficient. <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Increase interaction elements size, in particular answer selection button in theory test.

Subject	Scenario	Successful (1)	Completion Time	Total webpages	Notes / Suggested improvements
					<ul style="list-style-type: none"> – Increase given answering time (add: greater initial time, then adapt time according to user performance level?).
A2	1	P	3:43	Unavailable	<p>Issues:</p> <ul style="list-style-type: none"> – Generic problems in opening content links in Sessions/Result webpage (e.g. size to be increased?). <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Increase URLs size (font size).
	3	N	4:03	Unavailable	<p>Issues:</p> <ul style="list-style-type: none"> – Missing geolocation map in Sessions/Results (e.g. no GPS data available because test subject stood still during test execution. Fault in designing test scenario: user not instructed to move around the building to fetch GPS data.). <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Increase interaction elements size (e.g. home icon, “back” and “next” URL size in self-evaluation form).
A3	9	Y	1:31	Unavailable	<p>Issues:</p>

Subject	Scenario	Successful (1)	Completion Time	Total webpages	Notes / Suggested improvements
					<ul style="list-style-type: none"> – Box appearing while answer was processed (developer note: it is the Ajax execution feedback frame. It is due to longer than usual response time). – It is not clearly understandable that the user must touch the desired answer to input selection. <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Increase answer button size. – Provide indications that the chosen answer must be clicked (developer note: instructional windows before providing the first question?).
	10	Y	1:20	Unavailable	<p>Issues:</p> <ul style="list-style-type: none"> – Answer button difficult to select, e.g. too small <p>Suggested improvements:</p> <ul style="list-style-type: none"> – Increase answer button size. – Provide clear indication that the chosen question must be clicked/touched.
H1	1	Y	3:24	14	/

Subject	Scenario	Successful (1)	Completion Time	Total webpages	Notes / Suggested improvements
	2	Y	3:52	11	/
	3	Y	3:38	13	/
	4	Y	4:58	14	/
	5	Y	5:09	13	/
H2	4	Y	3:40	8	/
	5	Y	4:30	9	/
	6	N	7:03	12	Issues: <ul style="list-style-type: none"> – An empty result page is displayed. User returns to menu with no result content (developer's note: the system was unable to fetch adaptive content. The algorithm returns no data in case of application error, e.g. exception. The system provides a message, under the informational links session displaying "No Multimedia Resources Available").
H3	7	N	6:31	9	User selected answers other than those required by scenario.

Subject	Scenario	Successful (1)	Completion Time	Total webpages	Notes / Suggested improvements
	8	Y	3:46	12	/
	9	P	2:21	9	Issues: <ul style="list-style-type: none"> – Given answering time is insufficient. – Answer button is too small. Hence, the browser displays a zooming window, confusing the test subject.
	10	Y	3:36	9	/

Notes:

1) Y = successful, P = partially successful (e.g. test scenario completed, with impediments), N = not successful.

Appendix 5. Overall questionnaire ratings

Note: six questionnaires were responded, and all of them provided one rating per question.

Question	Domain	Total	Min	Max	Average
How easy was it to understand how to start a scenario?	Navigation	31	3	5	4,4
Were navigation controls easily visible?	Navigation	28	2	5	4,0
How easy it was to understand how to proceed to next task/webpage?	Navigation	25	3	5	3,6
How easy was it to understand if you could navigate back?	Navigation	30	4	5	4,3
How easy it was to go back to the main menu page?	Navigation	26	1	5	3,7
Average (Navigation)					3,9
How easy it was to notice interaction elements in the webpages?	Controls	28	3	5	4,0
How easy it was to understand the clickable element function?	Controls	25	3	4	3,6
How easy it was to use interaction elements?	Controls	18	0	4	2,6
How appropriate were the controls selected in relation to their function?	Controls	19	1	4	2,7
Average (Controls)					3,2

Question	Domain	Total	Min	Max	Average
Did you receive feedback messages when expected?	Content	27	1	5	3,8
How easy it was to understand feedback messages?	Content	28	3	5	4,0
How helpful were feedback messages?	Content	21	0	5	3,0
Were the webpages too full of information?	Content	24	1	5	3,4
How easy it was to understand webpages content?	Content	30	4	5	4,3
How easy it was to read titles and element labels?	Content	28	2	5	4
		Average (Content)			3,7
How easy it was to use the website?	General	23	0	5	3,3

Appendix 6. Scenario completion rate

Outcome	Count	Share
Successful	14	78%
Successful, partially (e.g. completed with impediments)	2	11%
Unsuccessful	2	11%
Total	18	100%

Appendix 7. Scenario completion time

Completion time (e.g. min, max and average) is computed in seconds.

Scenario	Min	Max	Avg	Delta (%)
1	204	214	209	5
2 ⁽¹⁾	232	232	232	0
3	218	238	228	8
4	220	298	259	26
5	270	309	289,5	13
6	199	423	311	53
7	249	391	320	36
8 ⁽¹⁾	226	226	226	0
9	80	141	110,5	43
10	91	216	153,5	58
Overall	80	423	234	27

Notes:

- 1) Scenarios 2 and 8 data is statistically subject to bias, because they were executed one time only, by one user only.

Appendix 8. Adaptation, UI and UX key features by user feedback

Statistic resulting from post-scenario questionnaire feedback analysis.

Key application area	Users	Feedbacks	UI	UX	Adaptation trigger	Adaptation Target
Content readability	4	6	X	X	Hardware screen size	UI layout UI element size
Content easy to understand	1	1		X	User cognitive abilities	Delivered content
Adaptable answer time (timed task)	1	1		X	User cognitive abilities	App behaviour
Smooth navigation	1	1	X	X	Hardware screen size	UI layout UI element size
Total feedbacks		9	7 (78%)	9 (100%)		