TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JYRI KÄRPIJOKI

PERSUASIVE TECHNOLOGY IN CROSS-PLATFORM SYSTEMS

Master's thesis

Examiner: Kaisa Väänänen-Vainio-Mattila
Thesis and topic approved by the faculty council of Computing and Electrical Engineering on 5 May 2014.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
KÄRPIJOKI JYRI: Persuasive technology in cross-platform systems
Diplomityö, 77 sivua, 25 liitesivua
Toukokuu 2014
Pääaine: UX
Tarkastaja: Kaisa Väänänen-Vainio-Mattila
Avainsanat: Suostutteleva teknologia, Käyttäjäkokemus, Mobiilisovellus, Järjestelmäriippumaton teknologi, Kokonaisvaltainen teknologia

Suostutteleva teknologia on ihmiskeskeisen teknologian osa-alue, jonka tarkoitus on muokata käyttäjän asennetta tai käytöstä teknologian avulla. Suostuttelevaan teknologiaan liittyy runsaasti muita aihealueita ihmiskeskeisestä teknologiasta, kuten käyttäjäkokemus, joka on suuressa osassa onnistuneessa suostuttelevassa järjestelmässä. Suostutteleva teknologia on erityisesti läsnä terveyteen liittyvissä sovelluksissa, jotka pyrkivät kannustamaan ja lisäämää käyttäjän motivaatiota. Tämä on mahdollista esimerkiksi sisällyttämällä järjestelmään sosiaaliseen kanssakäymiseen liittyviä ominaisuuksia.

Järjestelmäriippumaton suunnittelu helpottaa yksittäisen järjestelmän toteutusta usealle eri käyttöjärjestelmälle. Teoreettista taustaa järjestelmäriippumattomasta suunnittelusta voidaan hyödyntää käyttäjäkokemukseen ja teknisiin ratkaisuihin liittyvissä ongelmissa. Positiivinen järjestelmäriippumaton käyttäjäkokemus sisältää yhtenäisen käyttäjäkokemuksen järjestelmäosasta riippumatta. Tämä on mahdollista saavuttaa huomioimalla järjestelmäriippumattoman suunnittelun erityispiirteet ja pyrkimällä yhtenäiseen järjestelmämalliin. Teknisessä ratkaisussa tulee huomioida miten järjestelmä olisi mahdollista toteuttaa eri alustoille tehokkaasti, miten järjestelmän komponentit ovat yhteydessä toisiinsa ja miten järjestelmää olisi hyvä ylläpitää.

Tässä diplomityössä esitellään teoreettinen tausta sekä suostuttelevalle teknologialle että järjestelmäriippumattomalle suunnittelulle. Tässä työssä pyritään tutkimaan ja testaamaan esiteltyjä teoreettisia viitekehyksiä suostuttelevan järjestelmäriippumattoman järjestelmän kehityksessä. Fysioterapeuttinen prototyyppi suunniteltiin ja toteutettiin hyödyntäen esiteltävää teoreettista taustaa. Prototyypin toteutuksesta arvioitiin miten hyvin se täytti sille halutut ominaispiirteet ja ominaisuudet, sekä miten toteutettua prototyyppiä olisi mahdollista jatkokehittää. Lisäksi toteutuksessa hyödynnettyjen teoreettisten viitekehysten oleellisuudesta ja koetusta hyödystä keskusteltiin toteutuksen perusteella.

Toteutetulle prototyypille ei tehty käyttäjätestausta sen tarkoituksenmukaisuudesta tai miten hyvin valitut sekä toteutetut suostuttelevat menetelmät sopivat järjestelmään. Kuitenkin, valituista teoreettisista viitekehyksistä oli selkeää hyötyä prototyypin kehityksessä, ja ne erityisesti suoraviivaistivat ja helpottivat prototyypin toteutusta.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
KÄRPIJOKI JYRI: Persuasive technology in cross-platform systems
Master of Science Thesis, 77 pages, 25 Appendix pages
May 2014
Major: UX
Examiner: Kaisa Väänänen-Vainio-Mattila
Keywords: Persuasive technology, Cross Platform Design, UX, Mobile System, Pervasive Technology

Persuasive technology is a branch of human-centered science, which aims to modify user's attitude or behaviour with help of technology. Persuasive technology is related to vast amount of other topics from human-centered science, such as user experience which plays a major part in successful persuasion. Considerations of suited methods to achieve wanted persuasion is essential in persuasive technology and the consequences of the used methods on the user. Persuasive technology is especially present in health-related systems, which aims to improve the users' physical health. This can be done with various methods, such as by including social efficacy features to the system.

Cross-platform design is the answer for designing application for various platforms. Theoretical background of cross-platform design can be used to solve user experience and technical issues. Positive cross-platform user experience has positive coherent user experience regardless of the used system component. Positive coherent user experience can be reached by considering the specific characteristics of cross-platform systems and by aiming for coherent system design. Technical issues include how the implementation could be done efficiently, how the components of the system are connected to each other and how the system could be maintained.

This thesis represents theoretical background for both of these presented subjects. This thesis aims to explore and test the introduced theoretical frameworks for designing cross-platform persuasive system. By using this theoretical background, a prototype for physiotherapy was redesigned and implemented. The prototype's implementation was evaluated: how well it fulfilled its defined goals and how the prototype could be developed further. Additionally the relevancy in the development process of used theoretical frameworks and models will be discussed.

User studies for the prototype were not conducted in order to study how well the prototype suited for its purpose and context of use, or how effective and suited the selected persuasive system characteristics were for the system. However, the selected approaches for the design process including Model-Driven Development and Persuasive System Design model were found suited for this particular development of a prototype.

PREFACE

The development of the prototype in case study started in the fall of 2013. The concept of the system originally was designed by Hanna-Mari Nevala for her thesis “User Experience of Mobile Service for Physiotherapy: Case FysiApp” in Lapland University of Applied Science. The concept and its content was then reformed and developed to this seen form by Hanna-Mari Nevala and Mikko Luukkanen. The system content includes over 150 specific exercise instructions and over 400 instruction images.

According to the concept I implemented an early version of the prototype, which included the most essential technical functionalities. After I started writing this Master’s Thesis in the spring of 2014, I reformed the prototype to include the desired functionalities from persuasive technology and Model Driven Development was used to ease the cross-platform adaption for the implementation. Mikko Luukkanen is developing prototype from the presented form further and aims to study the benefits of the proposed concept in his Master’s Thesis: “Using mobile services to improve pilot fighter’s physical health” in National Defence University.

I would like to thank Kaisa Väänänen-Vainio-Mattila for supervising and examining my work. Additionally, for helping me in defining the structure of this thesis and for all the helpful pointers. I also appreciate all the comments and suggestions from friends who I asked to proofread this thesis. Lastly, I would like to thank various people for testing the developed prototypes and providing me essential feedback according to their experiences.

TABLE OF CONTENT

1 INTRODUCTION

Since the invention of modern day computers, developers have searched new ways to utilize them for various tasks. Computers have been proved to be a powerful tool for managing massive amounts of information and they have been utilized for various interaction methods for their users. Additionally, ubiquitous computing has become to a major factor in people's daily lives.

Unlike physical tools, computers have various ways for the users to be utilized and computers additionally provide feedback for the users. This gives rise to the field of human-computer interaction, which includes studying and developing the interaction between the users and computers. There are vast amount of commonly recognized theories of user behavior through the system's life-cycle and for designing the system's user interface. But, the behavior and attitudes of users cannot be exactly determined when they are adapting a new system in use.

The purpose of this thesis is to introduce and discuss methods of changing user behavior and attitude to a wanted direction by using persuasive technology. The object of persuasive technology is to change user behavior and attitude by using human-computer interaction. For persuasive technology to be successful and fulfill its intent, there are a few substantial problems before implementing such a system. How the user behavior or attitude can be changed, if their exact behavior or attitude cannot be known when designing the system? Is changing user behavior or attitude by using technology intellectually ethical?

As the usage of mobile devices has increased significantly, the human-computer interaction is not only based on desktop environments. The design for mobile systems is different for each mobile platform and in comparison to desktop environments. Consequently, when designing a persuasive system, the design of the system must be carefully designed for wanted results. The user experiences and features included in persuasive systems must have similar aspects in their designs regardless of the used platform. Thus, studies and theories for cross-platform design will be introduced from the human-computer interaction and technical points of view.

In this thesis, a case study of designing a persuasive cross-platform system is introduced to further explore and test the theoretical frameworks of persuasive technology and cross-platform design. The concept of the case study was originally developed by a physiotherapist student. The purpose is to offer physiotherapists a tool to communicate with their patients in real time and follow their level of performance in their designed training program. The described system provides additional information for the patients of the defined exercises. The original goal of the concept was to encourage the patients for interaction with their physiotherapists and increase their motivation in following their personal training program.

This master's thesis is divided into 8 chapters. The theoretical background for this thesis will be introduced in Chapter 2 and 3. We begin by introducing the background of persuasive technology, related human-computer interaction theories and related studies in Chapter 2. In Chapter 3 we will comprehensively introduce the basis of cross-platform design. These two first chapters additionally include developed frameworks and guidelines for designing described systems.

Chapters 4 to 7 are focused in the described case-study adapting the knowledge introduced in Chapters 2 and 3. First, in Chapter 4, the background of the system and its characteristics are introduced. In Chapter 5, we adapt a persuasive design model to the system design. The design model includes the system requirements of the final design. In Chapter 6, we present the design of the system. This includes the design models of the system, features of the system and the implemented system. Finally in Chapter 7, the system's design is evaluated in critical manner of how it could be improved and how its actual effect on users could be evaluated. The discussion in Chapter 8 concludes this thesis.

2 PERSUASIVE TECHNOLOGY

The objective of persuasive technology is to change human attitudes and/or behavior by using interactive computing system [1]. In this chapter, the focus is in software approach, especially in the mobile context of use. There are vast amount of studies in which persuasive technology has been used to change human behavior patterns to healthier and more active, which is the main goal of the presented case-study as well.

Fogg predicted that mobile devices will have a huge impact in persuasive technology [2]. The reasoning for this was divided into three factors: mobile devices are enjoyable to use, they are with and used by the users most of the time, and they have vast amount of capabilities [2].

Since Fogg's publication: "20 Perspectives on the Future of Behavior Change", so called smartphones have been raising their market share compared to feature phones significantly [2; 3]. In a study conducted in the US, the amount of smartphones have reached feature phone's market share, and their market share is predicted to reach 81% of all mobile phones by 2015 [3]. In addition, the smart phone users have their smart phones within their arm's reach 53% of the time, and within the same room 88% of the time [4]. Due to these facts, mobile applications provide high potential for persuasion, as they are accessible for most people and they are used, or near to the users, most of the time. Large amount of studies of persuasive technology in mobile context have been consistent with Fogg's prediction of mobile device being the tool for persuasion with technology.

2.1 Theoretical background

2.1.1 Background

Using specific methods to change human behavior or attitude is not a particularly novel idea. It has existed for hundreds of years in human to human communication and in media [5; 6]. The fundamental goal of persuasion is to change human behavior and/or attitude to a desired direction, which is the goal in media with advertising, political campaigns and even with propaganda [7, p. 5]. Thus it is not surprising that Fogg states in his publication that some of the responses for his first researches in persuasive technology were negative, e.g. calling persuasive research "immoral" [1, p. 5].

According to Fogg, some people saw the persuasive technology as a new potential method to grow business income [1]. Software, mobile applications, and web-sites offer the similar basic methods for persuasion as traditional media, including all from commercials to print magazines. The difference comes in when interaction is added to multimedia, which gives computer systems a strong advantage over traditional media in persuasion [1, p.6].

Fogg [1, pp. 6-11] offers the following advantages of persuasive technology compared to human communication:

- Persistency: Software is persistent to repeat certain wanted operations from the users, e.g. registering the software dialogues.
- Anonymity: The possibility of being anonymous provides the user with vast amount of possibilities. In particular anonymity is relevant for health-related information search or discussing private health-issues, which might be difficult without anonymity. Even though anonymity enables potential misuse from the user, the benefits, especially in social support, are more significant than the downsides [8; 9; 10].
- Data: Software can "store, access, and manipulate high volumes of data" [1, p.8]. This allows persuasive technology to access user's own and other user's data for persuasive feedback for the user. To add, human memory is limited, so registering and connecting all relevant information for persuasive feedback might not be as suited for the persuasion context.
- Modalities: The actual effect of feedback might not be the information, rather how the information is presented. In human-to-human communications, humans perceive three dimensions of information: feelings, non-verbal and verbal information [11]. Non-verbal communication and feelings are not always unambiguous for the receiver and they are culturally dependable [11]. To add non-verbal and feelings cannot be controlled in high level to achieve wanted persuasive effect [11]. Software systems provide only wanted information to achieve a wanted persuasive result. Also software systems are not limited to these three possible communication types; they also have the option to offer multimedia, recorded data, additional information and vast amount of other possibilities to adjust user behavior to wanted direction.
- Scaling: Even though a single individual can be very persuasive for listeners, it is not possible to duplicate the individual for all humans who would like the individual's help, e.g. motivational speaker. In contrast, software can be distributed for large amounts of people with ease.
- Ubiquitous: Ability to be accessible by the users in differentiating contexts with certain technological limitations.

2.1.2 Related theories in computer science

Even though computer science is technically oriented field of study, there has been large amount of studies related to users' attitudes and behavior in human-computer interaction. In result of these studies, there are generally established theories related to the reasons behind users' behavior and attitudes in using software systems. These studies are also essential in developing systems and theories for persuasive technology.

As human attitude, mood and behavior are not exactly technical issues, but the psychological field has also been included in the theories. In addition, explaining or predicting humans' behavior is a complex and difficult task [12]. Theories for users' attitude and behavior are not purely based on user studies, instead theories behind social psychology and cognitive psychology have been adapted to develop and justify theories in computer science [13]. Generally when the goal is to change, fix or modify anything, the information of current state is essential. Without knowing how the users likely interact with computer systems or their behavior when using the computer system, it is questionable to develop a theory to adjust their behavior or attitude. Thus, theories regarding humans' attitudes and behavior using computer systems and towards the computer systems are needed of which a few of the most relevant will be presented below.

- Theory of Reasoned Action (TRA): "Individual behavior is determined by behavioral intentions, i.e., an individual's attitude toward the behavior and subjective norms about the behavior." [13, p. 2] TRA is focused on the user having an intention, which is a plan or a likelihood of users behavior in specific situations and driven by what is believed. By belief, it is meant that the users have a trust or confidence in something, which is the reason behind the intention. However the theory does not explain the occasions of users not following their intentions with their actions. [14; 15]

Passos et al. studied applying TRA to agile software project teams and impacts of using it [14]. They identified in their first cycle of the research the factors which have an impact in team members' belief of behavior. By using recognized factors, e.g. organizational culture, they created a conceptual framework based on TRA. This was designed for mapping and analyzing the project team members' experiences (Figure 1). The data gathering for the analyzing was done in interviews, observations, document analysis, and focus group sessions [14]. Using the conceptual framework, the participants' beliefs main attributes were classified and categorized by frequency, source of belief, origin and context and related impacts. By using the model, conflicting beliefs, e.g. from the participant and organizational culture, were recognized, and the impact of the belief was measured.

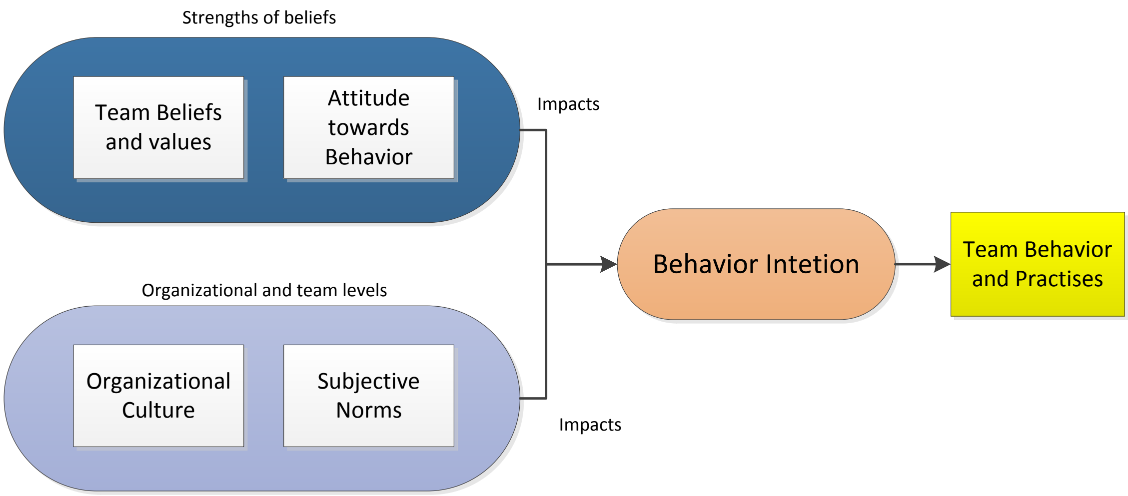

Figure 1: Conceptual framework for TRA [14, p. 4]

The conducted research suggests that past experiences and organizational culture has a strong impact on human behavior in described context [14]. Thus, it could be argued that in a similar way the social context has a strong impact on users beliefs whenever using any software system.

- Theory of planned behavior: "Individual's perception of the ease with which the behavior can be performed, i.e., behavioral control influences individual's behaviors." [13, p. 2] This theory does not differentiate highly from TRA or aim to disprove it. Theory of planned behavior is also based on users' intention to perform something. According to theory of planned behavior, the intentions are influenced by motivational factors, thus influencing the users' behavior. The behavior is a result of norms, attitude towards behavior and perceived behavioral control, which results into intention, which might result into actual behavior [12]. Especially attitude towards behavior and perceived behavioral control are essential on persuasive technology. If the user believes intention results into so called behavioral achievement, it increases the chance of user finishing the intention [12]. However, in the end, the user has the option to decide to do or not to finish the intention, even though all the factors resulting into intention would be fulfilled.

From these theories, the following theories and models have been developed with help of psychology:

- Technology Acceptance Model: "Perceived usefulness and perceived ease of use determine an individual's intention to use a system, which leads into actual system use, perceived ease of use impacts perceived usefulness, assumes that actors are free to act without limitations when they just have an intention to act, based on theory of reasoned action." [13, p. 2] Theoretically user behavior has a strong link to the perceived usefulness and ease of use of the system. Technology Acceptance Model is clarified in Figure 2. Thus they are essential, when the user is adapting to use a new system.

Davis conducted a study about the link between accepting technology and the perceived usefulness and ease of use of the technology. According to his study, usefulness has a stronger effect in accepting a new technology than perceived ease of use [16]. However, actual ease of use of the system provides better performance and usefulness of use (Figure 2). These will be result in total as cognitive response, which is a major determinant for intention of using the system [17]. Also, system's characteristics appear to influence behavior mainly through motivational variables and do not have additional direct effect on usage [17]. Technology Acceptance Model also provides a foundation for researching why users accept or reject technology and possibilities for improving system's design to avoid rejections.

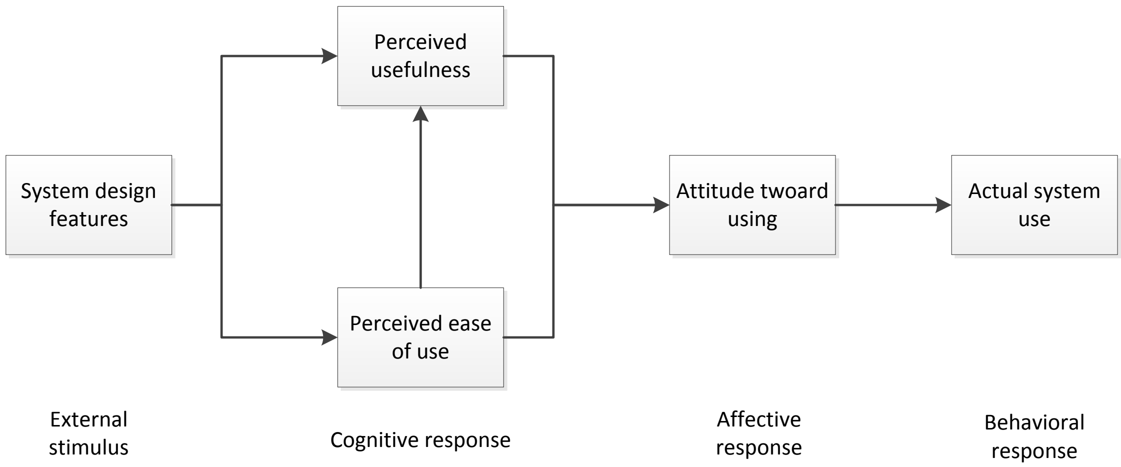

Figure 2: Technology Acceptance Model [17, p. 476]

- The Unified Theory of Acceptance and Use of Technology (UTAUT): "Performance expectancy, effort expectancy, social influence, and facilitating conditions determine the usage intention and usage behavior, whereas gender, age, experience, and voluntariness of use moderate this impact; extended from technology acceptance model." [7, p. 2] Venkatesh et al. formulated the theory of UTAUT and also contributed empirical support in favor of the theory [18]. According to UTAUT there are three direct determinants of intention to use; performance expectancy, effort expectancy, and social influence [18]. Two direct determinants of usage behavior; intention and facilitating conditions [18]. (Figure 3)

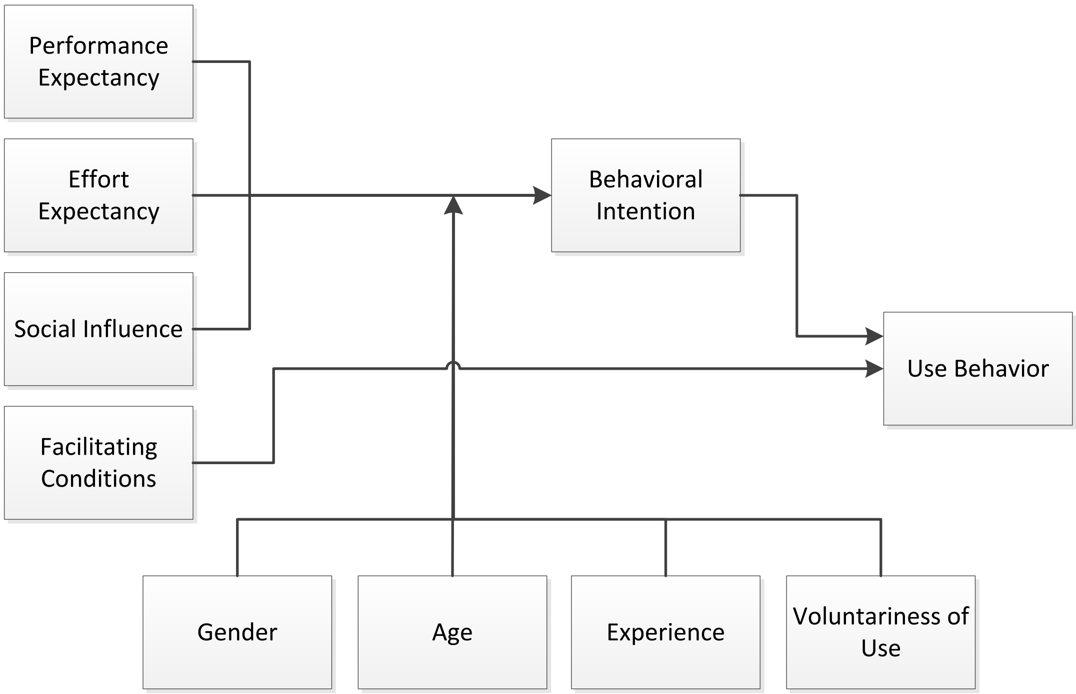

Figure 3: The Unified Theory of Acceptance and Use of Technology [18, p. 447]

Previously described theories must be kept in mind when studying the potential users and their context of use to achieve high possibility of persuasion. However, adapting and using the theories and models in designed systems features is a difficult task. For persuasive technology a few of the most commonly recognized and relevant theories and models for this study are presented below:

- Self-efficacy theory: Self-efficacy theory has been hypothesized that expectations of personal efficacy determine human behavior. Self-efficacy determines how much effort will be expected and for how long for behavior. Humans have personal efficacy expectations and outcome expectations, which will determine if behavior will be performed and in what level. Albert Bandura proposed in his publication that expectations of personal efficacy are derived from four sources of information: performance accomplishments, vicarious experience, verbal persuasion, and physiological states. These sources have an effect on each other as well, and human behavior can be changed by the modification of personal expectations. [19]
- Goal-Setting Theory of Motivation: Originally Locke and Latham introduced the theory of goal-setting theory of motivation, which emphasizes the correlation with goals and performance. "Research supports predictions that the most effective performance seems to result when goals are specific and challenging, when they are used to evaluate performance and linked to feedback on results, and create commitment and acceptance." [20 p. 1] The motivation is also effected by how realistic the goals are, thus the goals must be in possible to achieve with the individual's abilities. Goal-setting theory is also linked to self-efficacy, as the human's performance is related to the behavior and self-efficacy

determining it. Determining goals for groups and individuals does not have major difference in motivation, however if human perceives similarities between personal and group goals, it results into greater satisfaction and contribution to the team goals. [20]

- Social cognitive theory: Social cognitive theory tries to explain the human behavior and interaction in general level. Human activity is not purely based on environmental influences, or on their own thoughts of preferred behavior. Human's personal factors in the form of cognitive, affective and biological events, behavioral patterns, and environmental events determine human's activity and have an effect in each other. According to Albert Bandura, human activity is embedded from "self-organizing, proactive, self-reflective and self-regulative mechanisms" [21, p. 1]. Humans have an effect to social systems, and in some level, they are also products of social systems. [21]
- Transtheoretical Model: Transtheoretical model (TTM) is a dynamic process for modifying person's behavior to wanted direction, e.g. person wants to stop smoking. As the key is to change behavior, previously described theories must be kept on mind and they can be adapted in the models' stages. TTM is divided into five stages: precontemplation, contemplation, preparation, action and maintenance. The goal of the three first steps is for the person to adjust to the change and adjusting person's self-efficacy motivated for the change [19]. These three steps do not actually include actions towards the wanted behavioral change, but after the steps the person should be committed and motivated to change their behavior. The action stage's length depends on the set goal, e.g. for a smoker who would want to quit, it would take months to stop the habit. After the action stage, maintenance is quite self-explanatory, but the person should act similarly as in active-stage. Goal-setting theory plays a major part in TTM as well, as the person should be interested and motivated before the action step and maintaining the reached state. [22]

2.1.3 Methods for persuasion

Several studies have suggested that the key approaches to modify human behavior to desired direction is through persuasion, influence, nudge, coercion and possible deception [13; 23]. However using coercion or deception to achieve wanted behavior is questionable [24]. Berdichevsky and Neuenschwander introduced "Golden rule" for persuasive technology. The rile states that the system should not persuade in a way that the designers would not persuade themselves [24]. Therefore coercion and deception are not aimed to achieve in the case study.

As persuasion, influence and nudge are generally approved and effective methods for persuasion, their key characteristics are introduced below [23, pp. 1-2]:

- Persuasion: According to Eslambolchilar et al., the persuasion is characterized by three considerations. The use of digital tools in completing tasks and informing themselves about situations, "exploring and experiencing the outcomes of behavior" and using technological solutions in encouraging or by leveraging through social rules to achieve behavior change.
- Nudge: Selection in application which is aimed to direct users' behavior towards to the wanted behavior pattern. These selections should offer the user option of choice, and therefore they would not be coercive.
- Influence: Social influence, which can be achieved through six different approaches: reciprocation, consistency, social proof, liking, authority and scarcity

2.1.4 Persuasion context

As theories and methods for persuasion have been introduced, it is essential to acknowledge the elements which have an influence in persuasion. The persuasion context consists of three core-components: the intent, the event and the strategy [26].

The intent includes the persuader and change type. The persuader is the person(s) who is responsible of the designed persuasive system. Motives and reasoning for design of the system are essential for the persuader. The change type describes the goal of the persuasion and the transformation in behavior or attitude if the persuasion is successful. [26]

The event defines the actual elements related to the usage of the system: use context, user context and technology context. Use context describes the domain where the system is focused in. User context refers to the user and technology context to the technological context, including platform and technology related to the usage of the system. [26]

The strategy refers to the elements which compose the used strategy for persuasion: message and route. Message is a direct message, content or similar referred to the user, which is aimed to have persuasive effect on the user. Ideally these messages would be implemented by using the persuasive theory and methods to achieve wanted results. Route defines how the messages to the user are referred; directly, indirectly or by using both possibilities. [26]

2.2 Related work

2.2.1 Designing persuasive systems

Oinas-Kukkonen has conducted several studies of persuasive technology. According to his studies, the behavior can be changed by using the Persuasive Systems Design Model in the system design process [25; 26]. Transforming an idea from concept stage to functional persuasive system requires three necessary generic steps [25]:

1. Analysis of persuasion context and selection of persuasive design principles
2. Requirement definition for software qualities
3. Software implementation

The result of completing these steps properly gives the system possibility of changing potential users' behavior and/or attitude [25]. Especial focus in the first step is to discern the appropriate moments for interaction, e.g. via messages [26 p. 2]. Details of the persuasion context are presented in Figure 4. The software system characteristics are the results of the analysis from persuasion context.

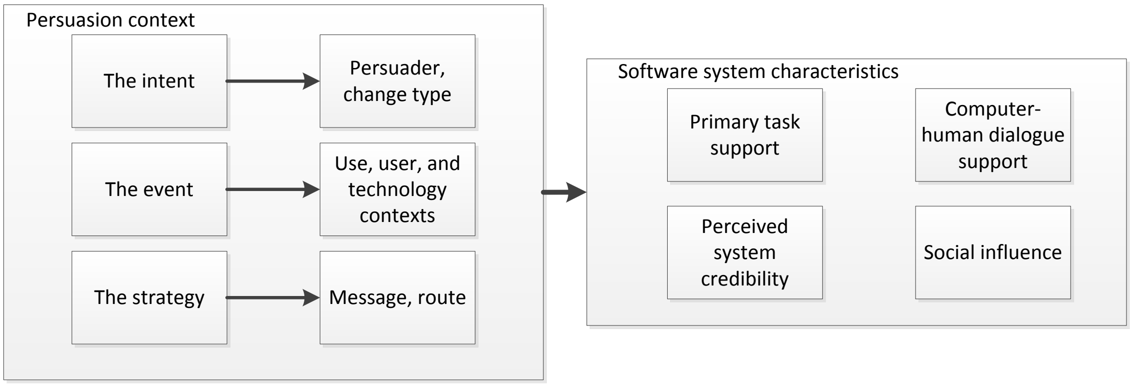

Figure 4: Persuasive System Design Model [26, p. 2]

For the end result to be persuasive, the system must fulfill fundamental quality requirements for software. Especial focus on quality goals is to reach positive user experience. Vast amount other usability related and technical qualities can be recognized for presented quality requirements; however user experience includes most of the essential ones. According to study conducted by Roto and Rautava, the most relevant elements for positive UX are [27]:

- Utility
- Usability
- Social value
- Enjoyment

These are the qualities relevant from the user's point of view, in result of that they also are the quality factors which have an impact to the persuasiveness of the system. However, to reach high utility and usability, the software's implementation must be flawless and suited for the context of use. Otherwise relevant factors for utility and usability, such as accessibility and reliability will be compromised, thus the positive user experience might not be experienced from the user's point of view. To add, the whole persuasion can fail if the navigation and interaction are not fluent due to technical limitations, even if the visual user interface is designed well [26].

As the user experience is a major factor in persuasion and according to Oinas-Kukkonen analysis of persuasion context is relevant step for developing persuasive systems. It is reasonable to suggest that user-centered design is one of the key principles in persuasive technology. User-centered design provides an answer and framework for design process for persuasive systems. By including the design principles of user-centered design to the development model Oinas-Kukkonen presented, reaching the highest potential of persuasion is possible.

Oinas-Kukkonen suggests in his study that persuasive system's characteristics can be categorized to "primary tasks, dialogue, system credibility, and social support" [25]. From these, the system credibility and social support are also major factors positive user experience [27]. To add, the content and principles included in these persuasive system's characteristics categories include relevant factors for positive user experience.

The primary tasks category includes elements which supports and possibly motivates the user to reach their goals and finishing their own or given tasks. Primary task support includes principles: reduction, tunneling, tailoring, personalization, self-monitoring and rehearsal [25].

Dialogue support category provides the elements for the actual interaction of the system and user. As Fogg stated, interaction is the tool which gives the significant benefit for software over media for persuasion [1]. To add, according to Oinas-Kukkonen, these principles are partly adopted from Fogg's ideas on social actors and media [25]. Dialogue support includes: praise, rewards, reminders, suggestion, similarity, liking and social role [25]. Reciprocity was excluded from the principles, because it is more a user characteristic than a system feature [25].

Systems creditability is also a factor in persuasion. Features which can be included by the developer are trustworthiness, expertise, surface credibility, real-world feel, authority, third-party endorsement and verifiability [25]. However, the developer's reputation and image are also factors, which the developers might improve or hurt by their actions, e.g. due to security breach, or improve by their other actions. These might have an effect to the users' opinion on the system's credibility as well, thus the credibility cannot be purely created with described system features. However, Oinas-Kukkonen states that presumed credibility and similar factors are excluded from the category, because they are not system features [25].

As it has been stated in several studies, e.g. Eslambolchilar et al. [23], Fogg [1] and Oinas-Kukkonen [13], social support plays a major role in persuasion [25]. The last category includes social support principles, which are: social learning, social comparison, normative influence, social facilitation, cooperation, competition and recognition [25].

Despite the categorization, the listed design principles effect is not just limited to their specified category. For motivational purposes from primary tasks and dialogue support; self-monitoring, simulation, rehearsal, praise and rewards have the same uplifting effect, thus improving the persuasion of the system. Tunneling, reduction, reminder and suggestion principles all guide the user using the system and help them to reach their goals. [25]

All the social support categorization principles and social role from dialogue support improves the social influence, which is also an element in the overall user experience of using the system. Tailoring, personalization, similarity and liking from task support also affect the user experience positively, which is a major factor in persuasion. System credibility support categories do not only improve the credibility, but also effects utility of the system and provides a link to real world. [25]

2.2.2 Evaluating persuasive effects

Oinas-Kukkonen introduced a model for measuring the success of persuasive technology applications [26]. The model is divided into five steps [26]:

1. Select the theoretical basis for research
2. Analyze the intent through the Outcome/Change (O/C) Matrix
3. Analyze the behavioral change support system (BCSS) through the PSD model.
4. Measure the behavior change
5. Explain the change through the theories, the O/C matrix and the PSD model

For the measurement to provide reliable and suitable information, the system’s features and aim in persuasion must be defined clearly and explicitly. Otherwise the possible results might not be comparable or usable to the domain, or for solving possible problems of the designed application. [26]

The O/C matrix includes three columns, which are the types of possible behavioral changes: change in an act of complying, a behavior change or an attitude change. The rows present possible successful outcomes of using the system, which are: formation, alteration, or reinforcement of attitudes, behaviors or complying. [26]

Table 1: Suggested O/C Matrix for persuasive technology [26, p. 2]

	Act of complying	A behavior change	Attitude Change
Formation	Forming an act of complying	Forming a behavior	Forming an attitude
Alteration	Altering an act of complying	Altering a behavior	Altering an attitude
Reinforcement	Reinforcing an act of complying	Reinforcing a behavior	Reinforcing an attitude

For the O/C Matrix the researchers should include the intended outcomes and the types of change [26]. The persuasion context and used strategies might differentiate highly from slot to slot in O/C Matrix (Table 1).

Once the goals have been set for O/C Matrix and the previously described the persuasive system design model has been adapted in the system design, the data for measuring the change will be gathered from the users [26]. Purely analyzing the system's persuasive effect by using theory would not be possible due to the fact that predicting users' behavior is not reliable [12]. However, even when the data has been gathered properly, actually measuring the effectiveness of persuasive application or pinpointing the cause of change is a complex task [26].

2.2.3 Ethical persuasive computing

As stated earlier, the persuasive technology has raised questions of its morality. Thus, to avoid questionable methods in persuasive applications, using ethical design guidelines and adapting ethical approach would be suggested. The content of this chapter is not just limited for persuasive computing, instead ethicality and responsibility for designing all human-computer systems is discussed. Convergence of ethics of persuasive technology is introduced in Figure 5. As the convergence suggests, the ethicality in technology is not just limited for persuasion, but it is also present in designing technology, and in computers, in wider broad.

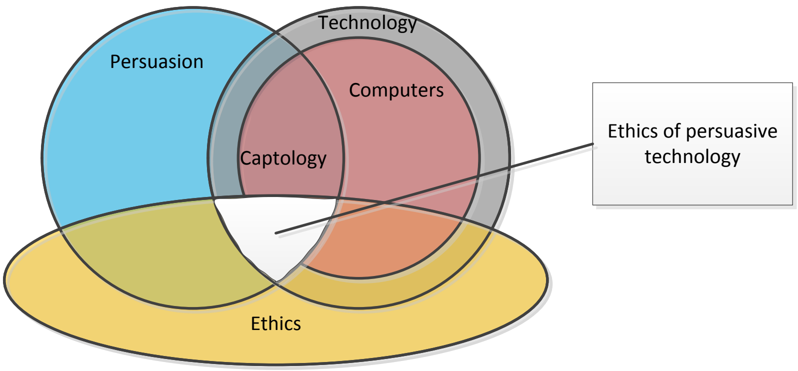

Figure 5: Ethics of persuasive technology [24, p. 53]

Davis states that just using principles and guidelines to avoid ethical issues in human-computer interaction is not enough to avoid ethical issues in designing systems. Instead, ethical issues should be considered in whole human-computer interaction field to help designer uncover and address ethical issues [28]. Good examples of systems which might have ethical issues are social sharing sites and freemium applications. Though, some would argue that the main-goal of freemium applications is to persuade the user to make in-game purchases. Especially emotionally and cognitively vulnerable users, e.g. children and elderly persons, should not be targeted with persuasion when designing ethically appropriate persuasive system [1].

As introduced in earlier chapter, the benefits for using computers in persuasion are superior compared to traditional media or human-to-human communication. Davis states that due to the qualities which provide the tools for high level persuasion for human-computer interaction, they are also the reason why the ethicality for using computers must be considered [28]. One of the main reasons is that computers' are considered as "intelligent and fair", the users might not be aware of the bias of the used system or perceive guidance by the system for wanted results [28]. In addition, the computers are persistent to guide the users for wanted selections [1]. E.g. every time when system is launched, the system launches dialogue for asking whether the user wants to receive advertisement from the system provider.

Davis points out a good example for persuasive technology compared to usability. For achieving good usability and high UX, there are vast amount of principles and design guidelines for achieving these goals. However for developing usable system, the design guidelines are not efficient enough – the users must be included in the design process. Observations and interviews provide essential information for prototypes, so the end-result of the product will fulfill the wanted requirements for the product. Similarly when persuasive system is designed, user studies are appropriate, so the designers will be aware if the system has wanted effect or if possibly the system has unintended consequences in users' behavior. Consequently, the persuasive technology has adapted user-centered design's development qualities in recent related studies. [28]

Berdichevsky and Neuenschwander pointed out in their early publication of persuasive technology the designer's responsibility in designing the system. According to their publication, the designer's intent and the outcome of the consequences of using the system determine if the designer is responsible for the consequences. However, the designers cannot be held for responsible if the outcome is highly unlikely, yet still they consider it unethical result [24]. The motivation of the designers' selections is also emphasized. The designer's responsibility in designing system is clarified in Figure 6.

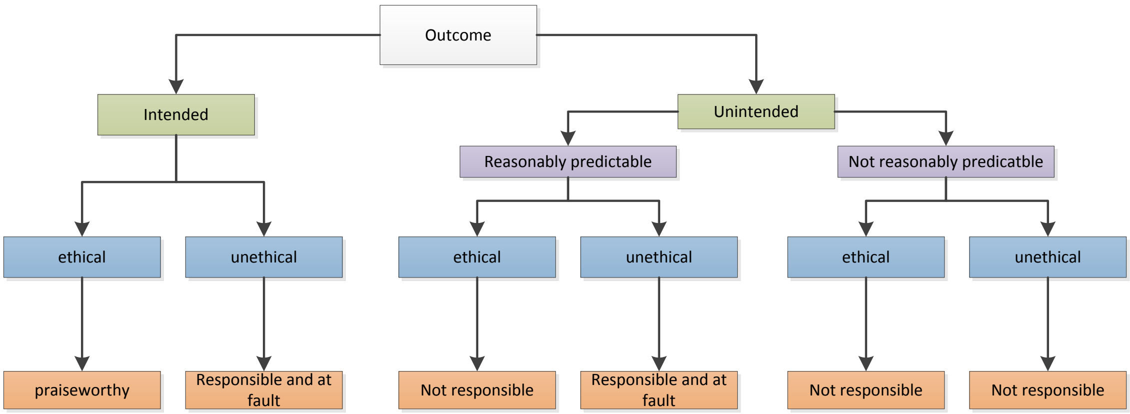

Figure 6: Designer's responsibility [24, p. 55]

Davis introduced two design guidelines are Value Sensitive Design and Participatory Design [28]. These design guidelines are not only limited for studying the persuasive effect on users, instead they also provide information for the systems stakeholders' beliefs of the system. These methods are explicitly to engage ethical issues.

Value Sensitive Design is based on legal and philosophical literature to recognize first five key requirements and on user studies which provide the sixth requirement for the framework. Value Sensitive design is developed by Friedman and it is a theoretical and methodological framework [28]. Value Sensitive Design emphasizes values of moral and ethical concerns in human-computer interaction. It has been used by vast amounts of researches in significant issues in computer systems, such as in delivering web browser cookies and in corporate knowledge-sharing systems [28].

"Key features of Value Sensitive Design include its interactional perspective, attention to both direct and indirect stakeholders, and a tripartite methodology." [28] Interactional perspective is an interactional theory, the users and social system have an effect on computer systems and computer systems have an effect on users and social systems. To add the computer systems have intended or unintended effect on the users, however the systems cannot force the desired change easily [28].

The second one considers all stakeholders to the system. As Berdichevsky et al. [24] stated, the responsibility for persuasive effect comes from the designers and their motivation for the made selections for the system and the intended and unintended consequences of the system for all stakeholders. In this the VSD's goal is to ask why the design might be considered harmful or beneficial. As ideally the persuasive system enhances the users' way of life, it is relevant to precisely consider the benefits and disadvantages of the system. The tripartite methodology is related to the second one. In tripartite methodology VSG helps the stakeholders the implications of using the systems and uncover and account the problems within the system usage. [28]

“Participatory Design (PD) is a family of theories and methods that involve potential users as full participants in design processes leading to the development of computer systems and computer-based activities.” [28] PD emphasizes the designers keeping persuasive and ethical theory in mind when designing computer systems [28]. The use of PD is not just limited for persuasive systems, but in optimal scenarios it would be adapted in all computer system projects. PD also emphasizes the users and developers communication similarly to UCD-approach, and thus helping the developers create innovative solutions, which also are designed well from the user’s point of view.

2.2.4 Persuasion in mobile context

As persuasion in general, persuasion in mobile context is not a novel concept either. Early as 2001, two researchers studying mobile applications found 72 different mobile applications dedicated for health. However they do not meet the set standards for today’s applications, but the principles did not differ significantly. Their conclusion of analyzing the applications was that most of the applications used one or more of following persuasive approaches: providing tracking, analysis and reference material for the users. [1, p. 186]

Principle of Kairos: Mobile devices are ideally suited to leverage the principle of *kairos* – offering suggestions at opportune moments – to increase the potential to persuade. Kairos is a principle for so called “opportune moment”, which is from five factors which mobile technology, can determine [1, p. 188]:

- Physical location
- Typical routine
- Time of day
- Your goals for the day
- Your current task

All the four first factors are possible to determine with current days mobile devices. However, determining the current task of the user cannot be determined with certainty. The user can provide additional information for the application of routine and scheduled tasks, but human’s random unexpected tasks cannot be scheduled or expected at defined time. [1]

2.3 Persuasive technology adapted in mobile context

In this chapter a few relevant studies conducted on mobile applications related to persuasive technology are introduced. These include applications which goal is to motivate users and applications which gather and process health related information in mobile context.

2.3.1 Persuasive applications

As stated earlier, the context of use plays a major part in persuasion. The new generation smart phones provide the option for accessing context information without additional hardware, such as GPS. Thus they provide the option for some level of knowledge of the user's context. Kimura et al. implemented and studied a persuasive application utilizing these possibilities called *iDetective* [29].

iDetective's goal is to encourage users to walk more in their daily lives [29]. When designing the application, they recognized and selected some key features and characteristics from other persuasive applications, e.g. *FourSquare*, and included them in *iDetective.* Chosen features were [29, p.2]:

- Application utilizes the smartphones hardware, e.g. GPS
- Application is appealing for potential users
- Application uses social comparison for persuasion
- Application does not bore the users

They implemented in the application features which fulfilled all these wanted characteristics of the application [29]. For the persuasion they chose three psychological persuasion techniques, which were [29, p.3]:

- Goal-Setting Theory: Goal-Setting Theory is based on that users are more motivated when they have challenging goals, compared to easy goals or no goals at all.
- Social Comparison: Social comparison was related to the goal-setting theory, that the users of *iDetective* could compare their achievements to other users.
- Transtheoretical Model: Transtheoretical model was implemented in *iDetective* that it uses user's context by the sensors provided on their devices. The information is then used by a feature called *Agent* in *iDetective,* which has dialogues with the user about their exercise-habits and compares user's answers to the results provided by the sensors. After a while of usage, the *Agent* provides information of benefits of exercises and notifications for exercising, thus possibly engaging the users using the system [19]. [29, p. 4]

Excluding encountered technical problems, e.g. measuring accurate walked distance, the main-issue with the *iDetective* was adapting transtheoretical model in high level to the application. According to Kimura et al. it is important for persuasive application to keep the user interested in the application, which was the goal of used *Agent*-feature for transtheoretical model. Thus, they plan to improve and increase possible interaction from the *Agent*, which is also related to the goal-setting for the user. For the social comparison, they did not encounter high possibility of privacy issues, because the application's shared information between the users was limited to estimated walking distance and gathered in-game points. [29]

Another approach for using persuasive technology in mobile context was implemented by Froehlich et al. [30]. Their goal was to motivate user for nature-friendly behavior [30]. The system was designed game-like to engage users use the application. The aim of the persuasion was to encourage users to be nature-friendlier in their daily activities. For the persuasion methods they used goals by emphasizing and rewarding users every time they make a “green choice”. Their goals and current progress was displayed repeatedly by changing the user’s wallpaper. Also, the designers tried to link concepts in user’s wallpapers, such as eco-friendly to saving money and exercising, which would be connected in user’s mind. This was based on findings from social psychology that repeating concepts together would become linked in human mind [30].

2.3.2 Information gathering and usage

The persuasive applications require relevant information for the users to be persuasive. To add the information gathering from the users should be completed with minimal effort from the users for the application to be persuasive [2]. As stated earlier, the information also should be reliable and relevant for the users. In result of this, harnessing the device’s possible sensors for the application would be optimal solution. However, gathering the data in for the application and presenting the information for the user in an understandable format requires a lot from the implementation.

Similar problem occurred in *iDetective*, in which the main-problem was linking gathered information and providing feedback based on the information. To begin with, the gathered GPS-data was inaccurate, and they had not provided enough possible interaction models based on gathered information. [29]

Cheng et al. managed to implement application for health monitoring of body motion in study conducted in University of Illinois. Their solution was tested on a midrange Android smartphone, which measured gait speed of patients. Measuring gait speed has been used in clinical application and it has a correlation with patient morality. Their root mean square error (9.98%) of gait speed estimation was even better than medical accelerometers (12%-15%) designed for this particular purpose. [31]

The solution can be used in monitoring severity of diseases, such as lung diseases and heart diseases [31]. As the solution is more accurate than medical accelerometers and it is accessible by owning a midrange smartphone, it provides a good option for people from suffering these diseases [31]. To add, it is a good example of how exact and essential information can be measured and analyzed for persuasive applications by using mobile devices.

If the required information is not accessible or available by harnessing the smartphone's device's hardware, multi-device usage will be required to fulfill the system's needs. Oinas-Kukkonen and Segerståhl state that computing products will be likely utilizing other devices, which will result into multi-device environments. These types of systems will be having distributed user experience, and the emphasis of designing the system will be understanding the overall user experience of the whole system. Additionally, understanding how the user perceives the interaction within the system and how the experience used persuasive strategies. [40]

In comparison to mobile systems which gather the information by using the smartphone's hardware, the user has few steps when utilizing a multi-device system. These might result in user being unable connect the devices to each other, due to their technical knowledge and experience or conflicts caused by human error or technical issue within the system. Thus, user acceptance is also emphasized for multi-device persuasive systems. Additionally, the multi-device systems face cross-platform issues, e.g. semantic coherence, which will be clarified in next chapter. [40]

3 CROSS-PLATFORM DESIGN

Cross-platform design stands for implementing software system for multiple computer platforms. The main focus in this chapter will be in unified user experience from platform to another. The importance of cross-platform design and reaching unified user experience has increased since tablets and smart phones have become popular. These also fulfill the technical requirements for running similar software systems as traditional desktop computers.

However, mobile devices are mainly limited to touch input and their operating systems have higher variation compared to traditional computers. Mobile operating systems also have suggested strict design guidelines, which provide the users better overall user experience for the device. Currently the most popular mobile operating systems are iOS, Android and Windows Phone 8 [32]. The iOS's and Android's design guidelines and operating system designs are unique and their design cannot be seen in traditional computers. However Windows Phone 8 includes similar concepts as Windows 8 deployed on laptop and desktop computers.

The need for cross-platform design is not limited for optimizing user experience for different operating systems. Vast amount of software systems are also based on web, accessible via web browser. The websites are mainly optimized for desktop environments. Providing high user experience for desktop and mobile based devices with same design would be a hard task. Thus, there are available mobile-optimized web-sites and mobile applications for the purpose, e.g. Facebook has individually designed mobile applications for iOS, Android and Windows Phone.

One relevant concept to cross-platform design is ubiquitous computing. Ubiquitous computing means that computers, their connections and data is available and perceived by humans. These allow the users to connect their devices and applications to each other at wanted time. Main problems in ubiquitous computing are related to using and sharing necessary data with various devices and the use of computer devices in various physical, cognitive and social contexts. The unified use and access of data is directly related to cross-platform device and cloud services, which enable the users accessing the data. [33]

Oulasvirta's opinions of developing ubiquitous computing and using multiple devices in work-environment are also adaptable in cross-platform design [33, p. 4]:

1. Minimizing overheads between activities
2. Remote resources for data connection
3. Propagating metadata on migration of data from device to device
4. Supporting device-specific resources, e.g. touch screen or keyboard
5. Using physical gestures for system features, e.g. data synchronization
6. Using appropriate materials depending on target context of use

Most of the listed potential goals are related to data usage and fluency of use [33]. Another key point is supporting device-specific resources, which is not always employed or used to maximal potential in cross-platform devices. The actual physical material of used devices will not be discussed in this chapter, as the focus will be in cross-platform systems software-implementation.

3.1 Theoretical background

3.1.1 User experience

The actual definition for user experience varies from one source to another. According to Hassenzahl [34], a product has certain features of which the individuals construct apparent product character. The features include content, presentation, functionality and interaction. The apparent product character includes pragmatic and hedonic attributes, which actually result into consequences of usage: the user experience. The model Hassenzahl proposes for elements of user experience is presented in Figure 7. [34]

Pragmatic attributes express the behavioral goals and achieving them, whereas hedonic attributes emphasize the psychological change in result of usage of the product [34, p. 35]. Pragmatic attributes include manipulation of the product, which means that the individual perceives the functionality of the product and is able to access this functionality. Hedonic attributes include stimulation, identification and evocation. These describe how the users can develop their skills, and identify and express themselves by using the product. To add, using the product can invoke positive or negative memories. [34]

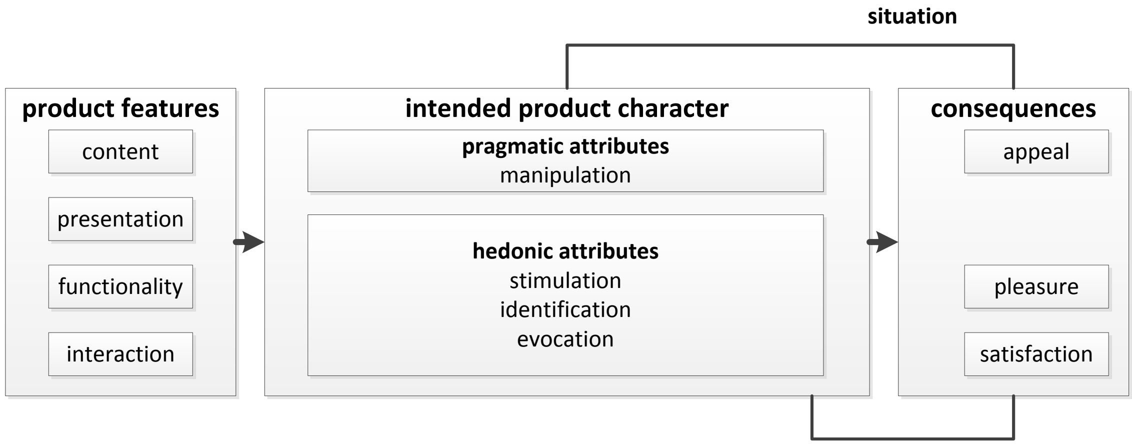

Figure 7: Key elements of the model of user experience [34, p. 32]

Even though usability and the system characteristics play a major part in user experience, actually reaching positive user experience is related to every contact point of the life cycle of the product [35]. This includes purchasing the product, taking it into use and other aspects of the product's life cycle [35]. Thus, Technology Acceptance Model and The Unified Theory of Acceptance and Use of Technology should also be considered when designing the product [16; 18].

Additionally to other aspects of user experience following values have been suggested for pleasant user experience; fun, joy, hedonic value, ludic value and enabling the user have flow in using the product [34; 35]. Consequently, the user's emotions, memories and values towards the product are crucial for pleasant user experience.

From the practical point of view when designing the system for the users the usage of the system has an obvious effect for the user experience. According to the organization for standardization the definition of usability is "effectiveness, efficiency and satisfaction in specified context of use" [36]. These fundamental qualities should be reached with the system definition for pleasant user experience, and they even could be described as a prerequisite before concentrating the other elements of the user experience.

As the main focus in this thesis is in mobile context, the key-elements in simplified form for mobile user experience could be: utility, usability, social value and enjoyment. These elements were recognized as most essential factors for mobile user experience by Roto and Rautava [27]. Utility and usability representing the pragmatic values. Utility includes perceived usefulness and reliability, and usability includes ease of use, efficiency and accessibility. Social value and enjoyment are then again representing the emotional side, i.e. hedonic values. [27]

3.1.2 Cross-platform mobile user experience

To start with, when designing mobile applications the native apps provide the richest user experience for the users [37]. Native apps have consistent look and feel, full access to platform hardware and data and highest performance [37]. The users have an expectation that they can run and use the application in a certain standard way – similarly as the other applications on their used device [37]. This would result into best user experience, but designing application for cross-platform usage has additional features which will have an impact on the user experience [37; 38].

Using the defined development environment by the operating system designer, the designer have the necessary tools for creating native applications with rich user experience. However these applications are just limited to the one specific operation system and the users are not exclusively using one platform for their tasks. For creating functional applications for all operating systems, there are few different approaches. Web, hybrid, interpreted and generated applications have the necessary functionality to be ran in cross-platform environment. However each of these have certain limitations, which makes these types of cross-platform applications less usable in all selected devices compared to native applications. [37]

With the knowledge of the nature of developing and implementing cross-platform applications, the users still have expectations of their preferred software systems to be available for each device and operating system. The users also expect that the applications available for cross-platform access are highly optimized for each device type and support each device's capabilities at high level. In addition, the applications are expected to have efficient flow of interaction and coherent user interface across the various platforms. [39]

The cross-platform applications are not only implemented for various platforms, but some of them are also interactive systems which are interconnected to each other. As they are interconnected, it creates challenges and limitations for the interaction designers. These interaction designers meet the requirements of consistency and continuity across the used platforms to ensure the fluent inter-usability of the system. [39]

According to recent studies, the user experience in cross-platform interactions are significantly influenced by prior and expected experiences using the system's features [38]. Thus, the user experience is not purely limited to particular component of the system, or to the current moment of usage. Wäljas et al. [38, p. 226] argue, that the central elements of cross-platform UX include:

- "Fit for cross-contextual activities: The structure of the service matches the user's activity, leading to effective fit for task in different contexts.

- Flow of interactions and content: Interaction with and across the service components are experienced as fluent and connected.
- Perceived service coherence: The service and its components are perceived as consistent and coherent; as parts of the same service"

The analysis for the central elements was done by using identified three key themes related for cross-platform design: composition, continuity and consistency [38]. The elements of cross-platform UX are related and affected by each other, and their effect on the user experience is result from how well and how these three conceptual key themes are implemented within the system [38]. However these particular introduced elements does not purely compose the user experience. Instead the elements related to user experience, e.g. usability and likeability, of the system have an effect as well. Overall user experience and elements related to it are introduced in figure 8. It takes in account the Hassenzahl's model for user experience, and the suggested elements and themes for designing positive cross-platform user experience by Wäljas et al. [34; 38].

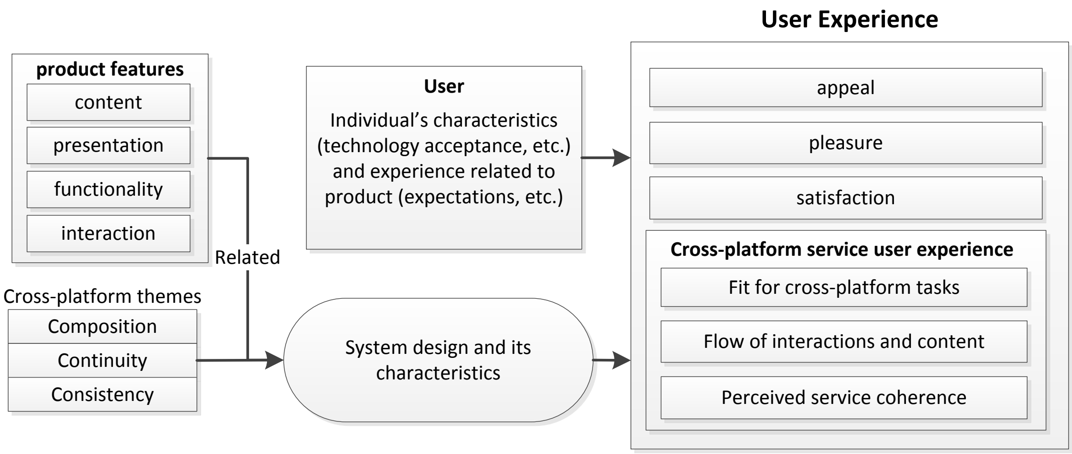

Figure 8: Cross-platform UX elements [34, p. 32; 48, p. 226]

Composition refers to how different platforms within a system relate to each other and how to functionality or modality of functionality has been divided for different platforms [33]. Continuity includes the support for interoperability, managing and migrating the used data and user's tasks for various platforms for the users [38; 39]. The continuity is established through synchronization of data and content, which is also one of the key-concepts Antti Oulasvirta recognized for ubiquitous computing [39]. Consistency is the most perceived by the users and heavily related to the usability of the system, thus to the overall UX of the cross-platform system. Wäljas et al. suggests in their framework for cross-platform UX design, that the consistency can be result of three factors in consistency: perceptual (look and feel), semantic (symbols and terminology) and syntactic (interaction logic) [38, p. 222].

In conclusion, the cross-platform UX results from various factors and the design process demands compromises in many cases. But the end-result of system's usage should result into positive coherent UX and not only limited for high utilization for single platform. This can be a result of designing the system as a whole, and identifying the suited interaction methods and characteristics for each platforms [40]. Additionally, it is essential to acknowledge that the designed features for each platform of the system influences the user's expectations and usage of other components [38].

3.1.3 Configuration of cross-platform system

The configurations for cross-platforms are designed fitting for the purpose of the system in question. In this chapter the two aspects of potential configurations for cross-platforms will be introduced: the way how devices are organized using the system and the way the system is used by variety of platforms.

Devices using the system can have three different roles: redundant, complementary and exclusive roles. Redundant devices have the same functionality and access to the data. When the cross-platform system has complementary devices, some of the devices provide access to data or functions, which are not available for all platforms. For exclusive approach, each device has specific purpose and they have individual access for data and functions, which other platforms do not have. [38]

For delivering and accessing the system data, i.e. provided service, the cross-platform system can have multichanneled services or they form crossmedia systems. Multichanneled services are used for systems, which enables the users access data and use functions with any device, i.e. the system is usable in full-mode with every device. In result of this, designing multichanneled services has vast amount of usability-related problems. Implementing full-scale website or desktop-application for mobile device including the functionalities in high usability and efficiency requires compromises and careful design. The crossmedia systems does not share the same problem. Their interaction and features for each platform are optimized for their specific characteristics and context of use. [38]

The service delivery types are related to employed device-roles as well. The multichanneled services are often used with redundant or complementary devices, when the crossmedia is used with exclusive devices. Finding and deciding correct approach in cross-platform configurations is one of the key-elements in developing high cross-platform UX.

3.2 Designing cross-platform system

We will introduce the design methods implemented particularly for cross-platform design. These will include theorized frameworks and models for implementing the user interface and technical side. These include methods for analyzing and creating an abstract interface, which is then converted for wanted platforms. The technical point of view is in developing mobile applications, because there are vast amount of development environments, which allow the developers create cross-platform systems for traditional desktop-based computers, e.g. Qt Creator [41].

3.2.1 User interface

When designing system with the goal of positive user experience, the design process of the user interface (UI) plays a major role. Sketching and low-fi paper prototypes are popularly used and their relevance for user-studies have been suggested just as informative in quantitative and qualitative researches as high-fi prototypes [42]. To add, 97% percent of UI-designers use sketching in designing interfaces and some of test users prefer low-fi prototypes over high-fi prototypes in user tests [42; 43]. However according to Antila and Lui [39], the challenges when designing interactive cross-platform system includes inefficiency of using low-fi prototypes and challenges of “seeing the big picture” in the design-process [39].

The reason for finding the low-fi prototypes inefficient, was because evaluating concepts without functional prototype was found difficult [39]. The actual difference in different platforms and interaction cannot be simulated or tested in a similar way as testing the usability of particular view of user interface. The other problem Antila and Lui found in their research was that the designers were designing part of an interconnected services, which resulted the designers purely focusing their own part. This again resulted into inconsistencies and discontinuities inside the application. [39]

Model-Based Design of User Interface has been suggested to be an efficient tool for designing user interface for cross-platform systems [44; 45]. Model-Based Design of User Interface frameworks provides an option to structure human-computer interaction to different levels of abstraction for designing the user interface [46]. The concept of these frameworks are similar to modeling databases or software architecture [46]. Aquino et al. introduced Cameleon Reference Framework, which has structured the user interface models into four levels of abstraction (Figure 9).

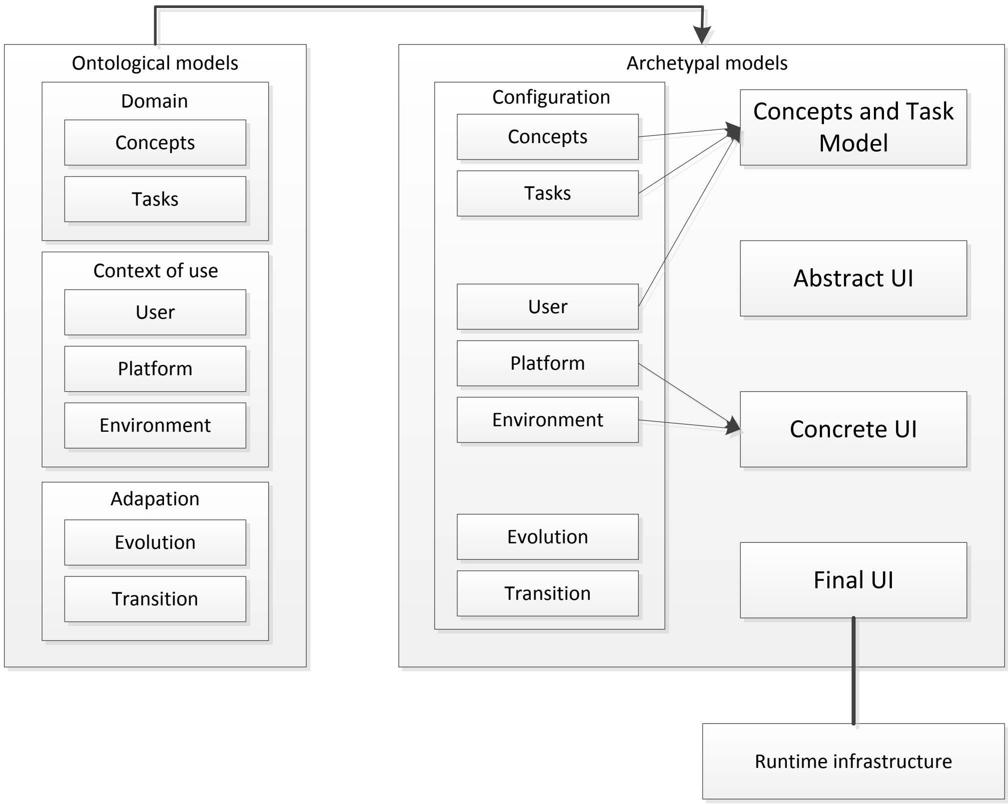

Figure 9: of Cameleon Reference Framework [45, p. 293]

According to the framework, the two first levels are the same for all possible targeted platforms using the system. These are used as a framework for modeling the two last steps which takes in account the types of platform and its qualities. As the concrete user interface model has been done, the final implementation for each platform can be implemented or generated. The definition of context of use and the domain is crucial for the two first models to be successfully implemented for the final stage designs. Additionally, the Cameleon Reference Framework takes in account if the domain or context use varies highly. In these cases, similar archetypal models can be used for differentiating ontological models, which uses the same runtime infrastructure. [46]

For the Cameleon Reference Framework to be successful, human-centered design methods and user studies are required to finish each model step properly and reaching positive UX. Identifying the user's actual needs, context of use and main goals of using the system by interviews. Applying user interface and user experience related theories in designing the user interface's graphics and interaction is also suggested, e.g. Jacob Nielsen's heuristics [47]. For the last step creating user interfaces based on the created models for each platforms acknowledging their specific qualities and possibilities. Especially taking in notice when designing for mobile platforms, which have comprehensive and strict design guidelines.

Balagtas-Fernendez [48] introduced similar approach for designing mobile application's user interfaces called: the model-driven development (MDD). MDD is particularly designed for human-computer interaction and for user-centered system design. "Model-driven development (MDD) is an approach to creating complex software systems by first creating a high-level, platform-independent model of the system, and then generating a specific code based on the model to the target platform." [48, p. 510] MDD would provide a method for designing and implementing cross-platform mobile applications with ease, especially for developers without high knowledge of each individual operating system's APIs and development problematic. [48]

Model driven Architecture (MDA) is model which applies the MDD and it can be used for MDD-approach. MDA is divided into four models (Figure 10):

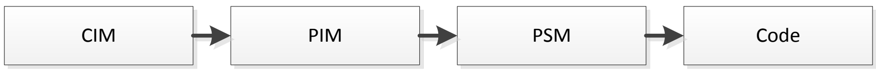

Figure 10: MDA [49, p. 295]

Similarly to the Cameleon Reference Model, MDA has two first models the same for each platform (CIM and PIM), which include the fundamental system design. Computational Independent Model (CIM) represents the system's activities for meeting the business objectives [49, p. 295]. Platform Independent Model (PIM) includes the system features and abstract user interface design, which is automatically converted for Platform Specific Model (PSM). Theoretically, the PSMs would be transformed to specific native codes [47; 49].

Diep et al. introduced a system model based on MDA for solving the cross-platform problem for MDA. Their suggested system model have similarities with MDA, and their correspondence between MDA has been introduced in table below. [49]

Table 2: System Model similarities with MDA [49, p. 296]

The system models	**MDA**
Task and Concept Model	Computational Independent Model (CIM)
Abstract User Interface (AUI) Model	Platform Independent Model (PIM)
Concrete User Interface (CUI) Model	Platform Independent Model (PIM)
Final User Interface (FUI) Model	Platform Specific Model (PSM).

Task and Concept Model has similar responsibilities as CIM; it describes specific definitions for the system, e.g. user's requirements and goals. The difference comes in PIM, where the introduced system model has divided it into two platform-independent steps: Abstract User Interface and Concrete User Interface. Abstract User Interface model is an abstraction of available controls for used platforms which have the same functionality. Concrete User Interface is the realization AUI Model and the system's user interface.

Final User Interface is the actual native application, which is generated from the Concrete User Interface Model with Transformation Module. [49]

Even though MDA or introduced system model cannot be used currently for cross-platform development with the full capabilities of it, the first steps can be used for creating and developing the cross-platform concept before implementing the native applications independently [37; 48]. Additionally, they support ideology behind the Camemeleon Reference Model [46; 49].

3.2.2 Technical approaches

The introduced technical approaches for creating cross-platform applications will be in mobile platforms. There are commonly known solutions for creating desktop software solutions with various cross-platform frameworks and development environments. Qt is a good example of this, which enables the developers develop and deploy their solutions for various operating systems; including Linux X/11, Microsoft Window (XP and later) and Apple Mac OS. Additionally, there is support for vast amount of embedded operating systems [50]. The same implemented solution can be built with platform specific toolkits and compilers for wanted platform. Thus, the focus will be in the problematic of implementing and deploying the same or similar implementations for new era mobile devices. [41]

As previously mentioned, there are designed methods for implementing cross-platform applications by using frameworks for cross-platform usage. These development approaches will be analyzed and introduced here. Xanthopoulos and Xinogalos made a comparative analysis between these approaches [37, p. 216], and the results of it will be introduced in Table 3:

Table 3: Comparative analysis of cross-platform development approaches [COMP, p. 216]

	Web	Hybrid	Interpreted	Generated
Marketplace deployment	No	Yes, but not guaranteed	Yes	Yes
Widespread technologies	Yes	Yes	Yes	No
Hardware and data access	Limited	Limited	Limited	Full access
User interface and look & feel	Simulated	Simulated	Native	Native
User-perceived performance	Low	Medium	Medium	High

As it can be seen from Table 3, when designing cross-platform applications, each of the approaches for designing applications has their cons and pros. According to the comparative analysis, the web-solution does not offer high possibilities for the application and their performance is poor compared to native applications (Table 3). However developing web apps application by using libraries, e.g. jQuery Mobile, does not demand high knowledge of mobile computing or time when developing prototype-stage applications. Obvious downside of web-approach is that they are mainly limited to business processes and they cannot access additional built-in device capabilities, e.g. GPS, which results them being useful for limited purposes [48]. [37]

Hybrid applications does not differentiate much from web-based applications. Their development methods is similar as web apps, but the solutions are compiled by using PhoneGap for creating application for application stores. However they share the same downsides as web apps; their performance is poor and they cannot access stored data in the device. Due to these facts some of the device manufacturers do not allow uploading hybrid applications to their application stores, e.g. Apple. [37]

Interpreted applications do offer native user interface and medium performance for the users. They also allow the applications to access some of the hardware and data by using specialized APIs. Interpreted apps performance is weakened, because of the application logic interpreting and the presence of specialized APIs required for the application to be functional. Generally interpreted applications can be shared and uploaded in application stores, given that they fulfill the application store guidelines. [37]

Generated would have high performance and they have native look and feel. However, non-commercial versions for generating the native code does not offer productive environment and they cannot be used for building anything more than very simple applications. There are currently various studies regarding the generation of cross-platform code through different approaches, but they have not found a well performing solution [48]. Thus, even though this approach is promising and would be potential approach for creating native applications, it is not yet in a stage for public use. [37]

In conclusion, platform-specific development designing applications provide major advantage over introduced approaches for designing applications. Platform-specific software development kits offer the optimized approach for developing applications without limitations and providing the look and feel with their included frameworks. However, platform-specific approach causes “fragmentation”, which increases the development time and maintenance costs for cross-platform applications. Consequently, if the designed cross-platform application does not demand high performance or significant amounts of using platform-specific resources, e.g. database usage, introduced options for cross-platform development can be the optimal solution for development process. [37]

3.2.3 UX design frameworks

In this chapter we will introduce a framework for designing positive cross-platform UX constructed from a field study in publication Cross-Platform Service User Experience: A Field Study and an Initial Framework by Wäljas et al. [38]. This framework has been constructed based on three main themes in designing cross platform UX (composition, continuity and consistency). According to these themes designable characteristics have been recognized, which will result into the final UX of the system. (Figure 11)

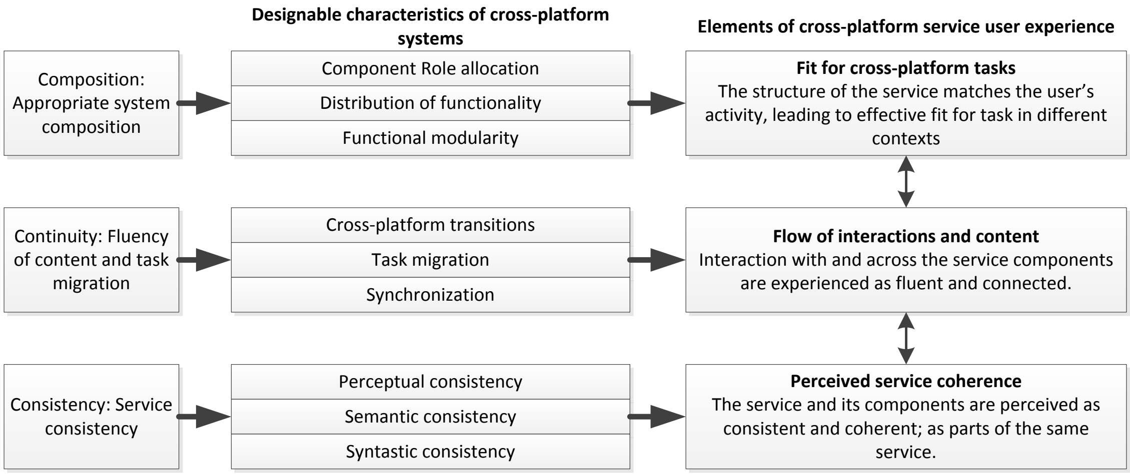

Figure 11: Framework for cross-platform service user experience [38, p. 226]

Appropriate system composition is achieved with component role allocation, distribution of functionality and functional modularity. Composition is related to the system's configuration and how the functionality is divided between the devices. Component role allocation refers how the users perceive the purpose of system's components and determines the expectations for each component. Users allocate the roles for each device, and it is crucial when designing a system whether using specific device is task or situation based. The users might selectively use specific device in certain situations, e.g. in a bus, or for specific tasks, e.g. complex operations on desktop work-environment. [38]

Distribution of functionality provides the option for designers distributing the functionality in a way that the system complements each device's individual strengths. If the component role allocation is perceived by the users task-based and the users mainly have the option for selecting the device, the distribution can be done with ease and reducing the complexity for each device. However the users might have expectations of functionalities for each device, which the system does not offer. Functional molarity is related to this issue; how each platform can be used in different situations if another is not available. This would result into limited use of the system, thus it is suggested to provide some degree of functional modularity for each device. [38]

Fluency of content task and migration is reached with cross-platform transitions, task migration and synchronization. Similarly to composition, continuity is highly related to the system's cross-platform configuration, but it is more of how well the technical aspects of the system have been implemented. [38]

Cross-platform transition occurs when the user switches from device to another using the same system. Cross-platform transition also includes the system adoption; how to transfer information and adopt the additional mobile version of the system. Additionally it is crucial for the user to understand the connection between devices, and how to use them alongside. [38]

Task migration and synchronization share the same technical aspects for them to be successfully implemented. When users switch devices whilst finishing tasks or simultaneously using the tasks, the synchronization of the system must be flawless. It also depends whether the system is multichanneled or crossmedial, to what extent the system provides the functionality. For multichanneled systems, the system must provide the same content and tasks for each device and synchronizing them across the provided platforms. Then again, crossmedial devices are optimized for specific tasks, thus the users might divide the work load for most suited platforms. In overall, the tasks should be divided in a logical way for the users use the system to its full extent with flawless and high performance synchronization. [38]

Service consistency is reached with perceptual consistency, semantic consistency and syntactic consistency. In contrary to previous themes, consistency is more related to the visual aspects of the system, as the previous two were highly regarding the functionality of the system. However, the users might perceive the problematic occurring with continuity for the system to be inconsistent.

The service consistency is for the system to share similar look and feel for each platform. Additionally same terminology and symbols should be used within the system, especially for same functionalities and notifications. The system should provide similar navigation scheme for each device, but mobile devices have limitations due to their screen size, inputs and operating system limitations, which results the navigation schemes to be fairly different. In conclusion, coherent system image is the main goal of service consistency**.**

3.3 Mobile design guidelines

Each of the major mobile operating system manufacturers argue that the best user experience for mobile applications designed for their operating system is reached by following their design guidelines [51; 52]. Additionally, their specific development environments include the design elements and elements for creating applications, which are fitted to their design guidelines. Implementing an application, which would not utilize these provided framework, would be inefficient for the designer, as it would demand high volumes of implemented code and icons. For the purpose of providing an example of differing guidelines and operating systems characteristic, we will introduce the guidelines for Windows Phone 8 and Android. These two operating systems are also used in the case study. Additionally, their conceptual interface design does not only differ in visual design, but for designing the functionalities and navigation for the user interface has also differing approaches [51; 52].

When creating native applications for cross-platform system, the mobile design guidelines do not have major effect on the composition or continuity of the cross-platform system [38]. The main difference comes in navigation, formation of selections and in icon design, which results into breaking either the consistency of cross-platform design or the suggested mobile guidelines. However, the operating system design guidelines emphasize the unity of the design of the applications for high user experience [53].

The operating system guidelines share a lot of common user-interface design principles, e.g. designing informative and understandable icons. These will not be individually presented; instead the focus will be overall user interface and made selections in the design guidelines, which needs to be followed for specific user experience.

3.3.1 Windows Phone 8 design principles

Microsoft's design principles for Microsoft design include: pride in craftsmanship, more with less, fast and fluid, authentically digital and win as one [53]. The content of these principles will be clarified in this chapter.

The Windows Phone 8 design is designed in a simplified and clean manner [53]. The design guidelines suggest the designers only leave the most relevant elements, i.e. needed commands and functionality, on the screen (Figure 12). The principles for designing applications for Windows Phone 8 emphasize the essence of the structure of the user interface, that the user interface has hierarchy and balance. Hierarchy and balance will result into sense of structure, rhythm and completed look for the applications. [53]

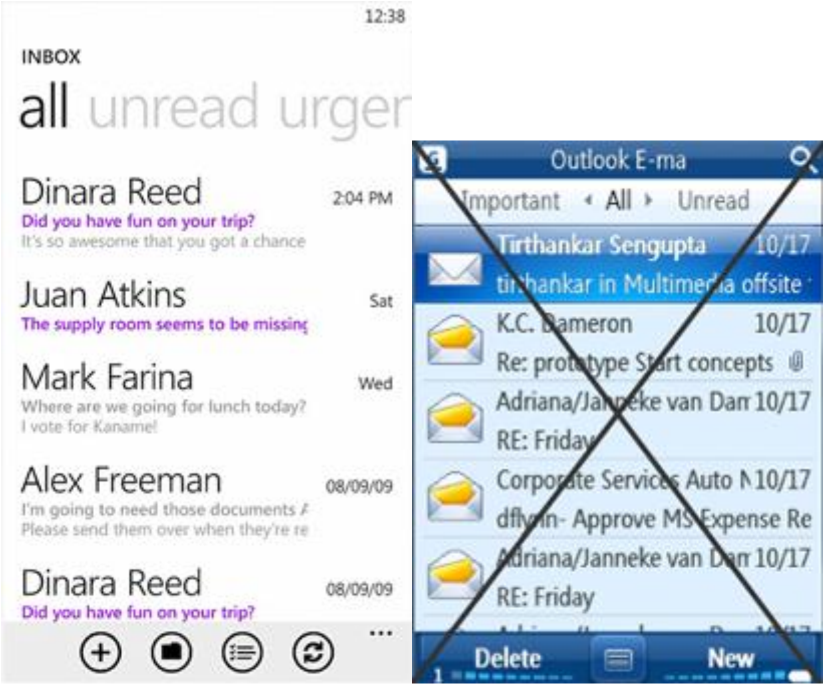

Figure 12: Windows Phone, Clear current design (Left) and Symbian design in comparison (Right) [53]

In contrast to the simplified and high level of structure, Windows Phone has interactive elements updating and changing over time. These are designed to provide the users additional essential info quickly for the users with a glance of the screen. As the Windows Phones navigation and content display differs from the other operating systems, e.g. Panoramic View, providing visual motion in navigation helps the users understand the interface's functionality and navigation (Figure 13).

Figure 13: Windows Phone, Still pictures of motion used in Windows Phone's applications [53]

Lastly, in the principles Microsoft emphasizes creating unified user interfaces and using available resources for highest performance for the users. This principle also includes the cross-platform design, as Windows 8 is available for desktop computers and tablet devices. The designed Windows based applications for cross-platform availability should follow the similar design, though optimized for selected platform. [53]

3.3.2 Android design principles

Similarly to Windows Phone design, Android has high-level principles for designing applications, which are: Enchant me, Simplify my life and Make Me Amazing [52]. Android design principles take in account more of the functionality of the system than the Windows Phones and emphasizes the actual usage of the system than the visual design.

When designing icons for the operating system or for individual applications, the icons are encouraged to be directly touched and manipulated objects. Additionally, they are not recommended to be simplified, but instead to be visually attractive and recognizable (Figure 14). Descriptions for information and selections should be kept brief, and irrelevant information hidden from the users. And if it is possible they should be replaced with pictures or icons. Pictures and icons for selections over words are recommended for two reasons: reducing the cognitive load and they are more efficient than words when used properly [52]

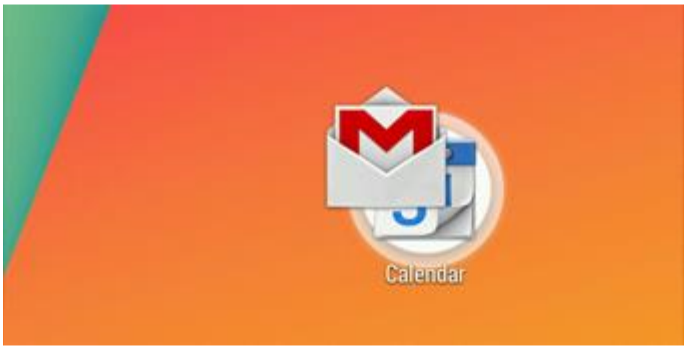

Figure 14: Android, Example of Android Icon [52]

Personalization is a major factor in Android applications and operating system. Modifying visual design, and providing the option for it is highly recommended. The users should be provided options for modifying haptic gestures according to their preferences and modifying user interface selections to possible extent.

Personalization comes in the emphasized efficiency of use for Android devices. Recommendations for enhancing the efficiency of use are either user-based or system-based. System-based efficiency functionalities include: remembering used functionalities and making sophisticated guesses for the users. For the first one the applications should collect information of the user's usage, and provide most used selections for the users when they are using feature of application. The second one is efficient when the possible selections are limited, and the provided guess of wanted functionality does not block the usage of current application and it can be undone, e.g. after taking a picture, providing an option for sharing it in social media. User-based efficiency related functionalities are creating shortcuts whenever possible and only showing essential information for the usage (Figure 15). [52]

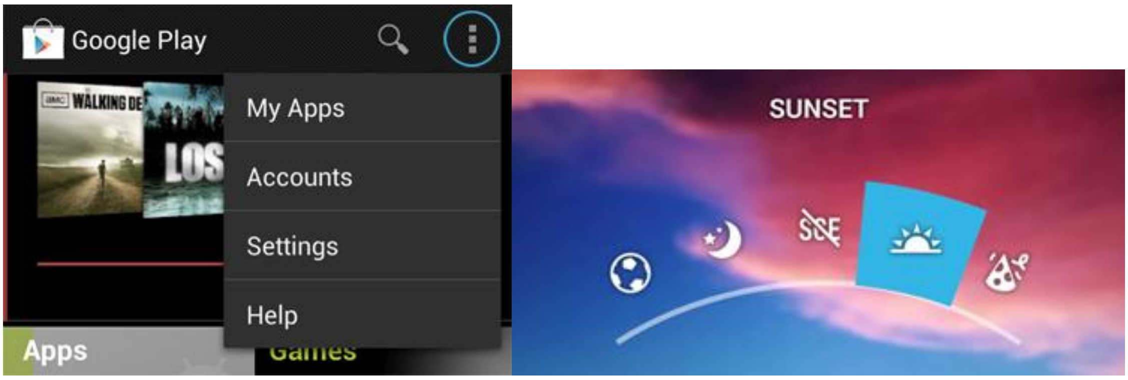

Figure 15: Android, Menu and Shortcuts [52]

3.3.3 Comparison of Android and Windows Phone designs

The main difference between Android and Windows Phone comes in the top of the view for the Android, called Action Bar. The action bar for Android application states the current view, option of navigating back and possible functionalities or navigations for the current view (Figure 16). The main action bar allows the user to perceive the application's hierarchy, whenever they can navigate back from the current view. The possible functionalities allow the user to access wanted data from the specific view without navigation within the system. [54]

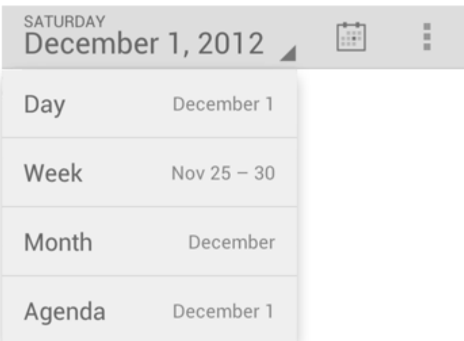

Figure 16: Android Action Bar [54]

Similar concept for Windows Phone is App Bar, which is in the bottom of the screen (Figure 17). However the App Bar does not state the current view or possibility of navigating back. Additionally, it is not available in most of the views for the Windows

Phone applications, and it can also be hidden (Figure 17). For some of the possible views the app bar is not recommended to use, i.e. Panorama View, thus, it is not as commonly seen or expected by the users. In most views, the possibility of navigating back is not available for the users. [51; 55]

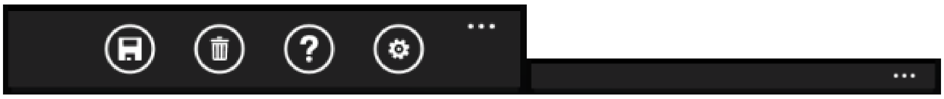

Figure 17: Windows Phone, App Bar [55]

The panorama view is one of the key-differences for the operating systems. It is only used by Windows, and is “a part of the native Windows Phone look and Feel” [51]. The ideology behind the panorama view, is to maximize the screen space for the content of the view (Figure 18). The panorama views include background art, which is below of the content to give “visually rich content presentation” [51]. Similar concept for the Android is Fixed Tabs (Figure 19), which allow the users to navigate similarly between related views, and by using the same swiping gestures as in Panorama View in Windows Phone [51; 56]. However, the content (e.g. images) should be emphasized in android applications and the background should not include distracting background art diminishing the content visually [56].

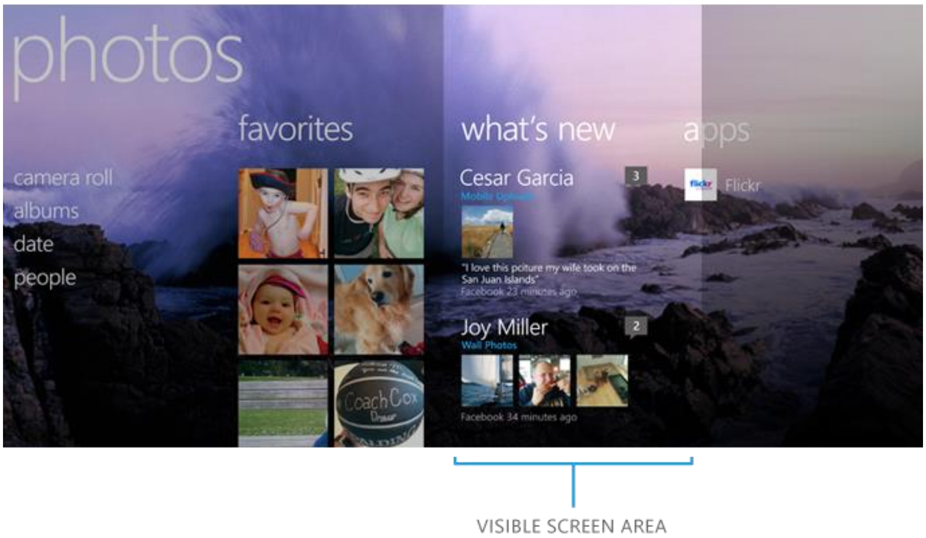

Figure 18: Windows Phone, Panorama View [51]

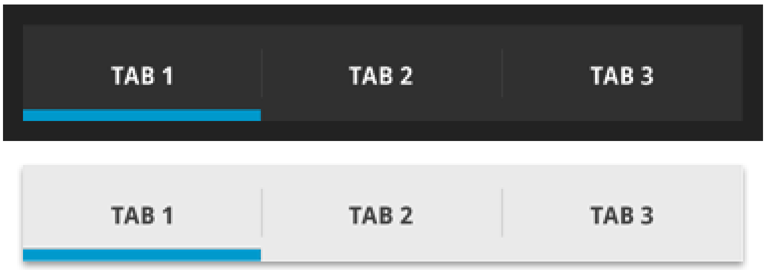

Figure 19: Android Fixed, Tabs [56]

Other essential differences, does not have similar impact in design, which include semantics of typography and using dialogs. For the Windows Phone the titles should always be in lowercase. However, this is an opposite of common design principles and not only differing from Android's design principles [57]. Dialogs are mainly used for Windows Phone for temporarily inform the user of something relevant of current moment, whereas in Android they are used for various purposes, e.g. additional information of selections [58; 59].

Additionally to these introduced conflicts in design principles, the principles include more minor differences. But it can be seen from the introduced differences, that using similar user interface design for Android and Windows Phone applications would not be convenient. These could be avoided by selecting similar types of views, e.g. Pivot-view from WP and Active tabs from Android. However, the operating system specific limitation, could make the implementation of forced similar designs difficult. Additionally, Google and Microsoft emphasized in their design guidelines that the designed user interfaces for applications should be designed by using the available resources to their maximal potential [52; 53].

4 CASE STUDY: FYSISYSTEM

The original concept of the case study was developed in Lapland University of Applied Science by Hanna-Mari Nevala in her thesis: "User Experience of Mobile Service for Physiotherapy: Case FysiApp". In the case study we will present a system designed for physiotherapists called *FysiSystem.* The original concept of *FysiSystem* was to provide the option for physiotherapists to create individual training schedules for their patients, follow their patient's progress and interact with their patient's by using the system. Additionally, one of the fundamental themes for the *FysiSystem* was to encourage the patients to interact with their physiotherapists. The *FysiSystem* consists of the physiotherapist's application (*FysiSystem Trainer)* and patient's application (*FysiApp).*

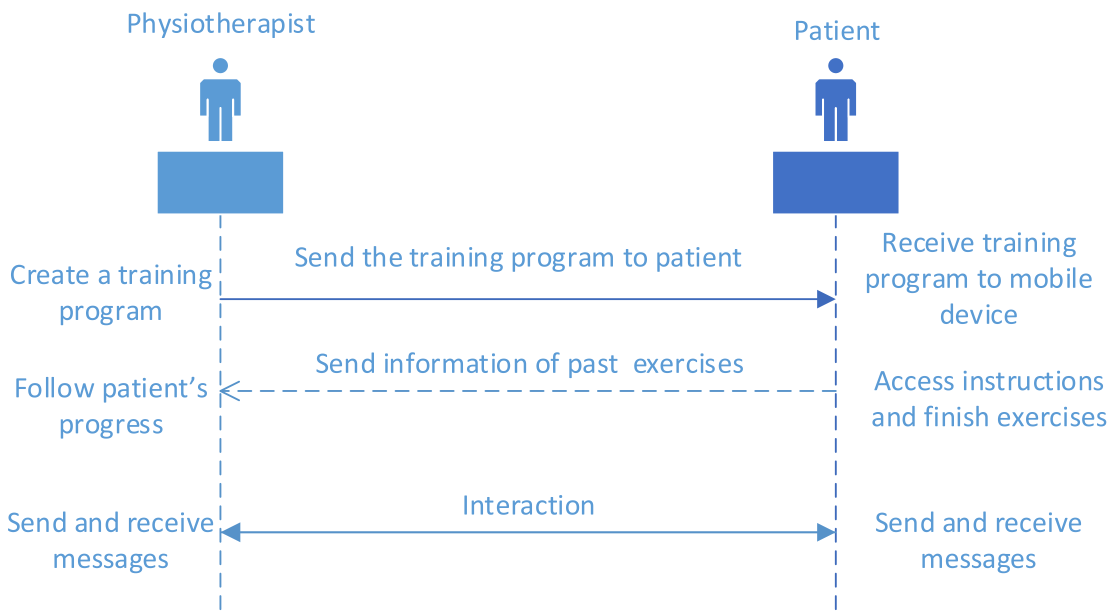

Figure 20: FysiSystem Concept

The *FysiSystem Trainer* is a cross-platform system, where the physiotherapist can access the system features with their desktop-computer or Windows Phone mobile device. In the design process of the *FysiSystem Trainer*, professional physiotherapists were interviewed of their daily tasks and they were asked for feedback of an early prototype of the system, which was created according to the described concept.

The patient's application, *FysiApp*, was designed for Windows Phone and Android mobile devices. *FysiApp* enables the users to access their personal training schedule, interact with their physiotherapists and access comprehensive exercise instructions. Additionally the *FysiApp* includes features for monitoring their performance. In theoretical basis, the *FysiApp* could enhance patient's motivation for finishing and following their training schedule. Theories and frameworks for designing persuasive application were adapted in the design of the *FysiApp*.

4.1 Design rationale

4.1.1 Health benefits

According to theoretical background, using technology for persuasion can be an effective method to adjust attitude to wanted direction [2; 19; 20]. The designed system does not aim to decrease personal interaction between the patients and physiotherapist, but instead one of the goals of the system is to increase interaction between the personal meetings and increase the motivation for the patients to follow their specific training schedule.

The essence of the motivation for physiotherapy has been studied from different aspects and approaches [60; 61]. The psychological aspects play a major part in the success of psychical training, which results into a certain level of performance in training and whether the physiotherapy is successful [60]. According to Tait, 25 to 40% of patients in physiotherapy have psychological disorder or psychological overlays to physical disorders [60]. Thus, these psychological aspects and disorders, which affect to the person's motivation, and to person's conscious and unconscious behavior, will be presented below.

Anxiety has major impact in the person's behavior depending on the degree of anxiety the person is experiencing. If the degree of anxiety is high, the person might not be prepared to put required effort for the physiotherapy to be successful. Depending the severity and nature of the disability, the disability might persist when the physiotherapy and/or the person's sick leave ends. Additionally, anxiety is related to fear of doing regular activities after the disability and anxiety includes range of different issues which might increase the experienced anxiety. Anxiety also increases in the tone of the body muscles, which commonly can result into added pain. [60]

Compared to anxiety, hysteria is deeper emotional conflict which is developed mainly through the person's entire life. Mild degrees of hysteria are common for humans and they do not cause a lot of problems. However, if the person experiences high degree of hysteria, they might lead incapacitating the person from easy tasks in life. To add, these persons have a need of retreating from difficult situations and they have tendency to illness. As the main cause of physiotherapy is a disability or an injury, patients experiencing high degree of hysteria might have difficulty adjusting to the situation. [60]

Obsessional neurosis is related to anxiety and it also increases anxiety. By obsessional neurosis it is meant that the person has a need for repeated acts and need to run according to specific system. Unlike anxiety or hysteria, in some cases obsessional neurosis can be beneficial on physiotherapy, as it can be provide additional motivation for the patients and they can carry out their physiotherapy without observation. [60]

Lastly, one of the most essential factors in physiotherapy is money. Money can be increasing or decreasing factor in motivation for the patients following their training schedule. As high amount of the patients have benefits paid from disability to work and some of the patients might seek invalid pensions after severe injuries, they might have no interest in physiotherapy and returning to work. However, according to Tait, from the people who are unwilling to return to work 50 % have financial considerations of lower income whilst they are on sick leave. [60]

Using previously introduced persuasive theories, such as self-efficacy and goal setting theory in physiotherapy, has had significant increase in motivation of the patients and their performance. These results from higher motivation have had significant effect in improvements in measured physical capabilities of the patients, in their self-perceived general health and in follow-up meetings with their physiotherapists. Thus, it is relevant to study new ways for increasing the motivation and possibly decreasing the effects of psychological disorders of the patients. [61]

4.1.2 Systems currently in use

These presented systems were used by the physiotherapist in the background study and are used for creating the training programs for the patients by the physiotherapists (Appendix A). The systems are mainly designed for traditional desktop environments purely for physiotherapists, and do not take in account interacting with the patients or accessing their training progress. The output of these systems is printed or emailed to the patients, which decreases the interaction and possible persuasive methods for achieving higher motivation for the patients.

One of the popularly used systems in physiotherapy is *PhysioTools*. It was used by all of the three physiotherapists interviewed for the background study of the physiotherapy. The *PhysioTools* is accessible via any device with internet access and internet browser. By using the tool, the physiotherapists can create and modify training programs for their patients and share them with their patients. However, they can only share the training information with printed copies or by sending it to their patients' emails. Additionally, the physiotherapists can share their exercises and patterns to each other and the system includes high amount of various exercises (over 2300). [62]

Another example of web-based physical therapy system is *PhysioFile*. It was used by one of the interviewed physiotherapists daily in work. This system is more for information and knowledge based system for the physiotherapists for accessing essential and up-to-date information with various formats including video, publications and images. However, this system does not offer any option for creating training schedules or sending the accessible information to their patients. [63]

Both of these systems provide vast amount of possible exercise guidelines including pictures and step-by-step guidance for doing wanted exercises properly. However, these systems are available and visible for the trainers. The users have only the actual output from the system, i.e. printed exercise-schedule, and the system does not take in account following the progress of training or encouraging the clients reach their set goals.

4.2 Users and context of use

4.2.1 Physiotherapist

The description of physiotherapists and their use of context are based on two part questionnaire conducted on two physiotherapists (Appendix A). The physiotherapists are from higher education background; bachelor’s degree or higher. The interviewed physiotherapists were using computers for their work assignments; creating exercise programs and saving relevant information of the patients. Their usage of computers in their free time was described to be low, approximately a few hours weekly.

The system’s target group is for similar groups, due to the nature of the system. The system provides the option for staying in contact with the patients with ease and sending the programs to the patients directly, however the prerequisite for the system is that the physiotherapist actually uses computer in their daily tasks. In other case it requires extra effort from the physiotherapists in using computers and likely learning the use of the system.

Their main context of use is at work, where they can access either desktop- or laptop computer. Using the system does not require higher knowledge than the previously described systems in use. Thus if the physiotherapist has basic knowledge of using computers, the system does not cause too high limitations for usage.

4.2.2 Patient

The patients vary highly in age, gender and in education background. There is not a specific pattern. However the system requires usage and owning a smart phone. Consequently, the users have some level of technical knowledge and experience before starting to use the described system.

The context of use varies from patient to another. Most of the exercises are designed to be done at home, but some of the training schedules require several exercises daily. This results some of the exercises to be done at school or at work. The system is available for common Windows Phone and Android devices, which have access to internet.

5 THE PSD MODEL ADAPTION FOR FYSISYSTEM

Oinas-Kukkonen presented a PSD model for designing and implementing persuasive systems [25]. The fundamental concept of the system was defined in last chapter. The PSD model is used to specify which characteristics would be beneficial to implement for the patient's application to increase patient's motivation in performing according to their training schedule. The PSD model includes the analysis of persuasion context and selection of persuasive design principles [25]. According to the PSD model adaption, the requirements of the system and features will be defined and presented in this chapter.

5.1 PSD Model

5.1.1 Persuasion context

The persuasion context includes essential factors for persuasion and it is discussed from the user's point of view. These include the intent, the event and the strategy of persuasion used to modify the user behavior or attitude.

The intent includes the persuader and the type of change. For the persuader in the *FysiSystem*, the persuaders are the following:

- Producers of the interactive technology (endogenous): implementer of the system (author of the thesis)
- Distributers of the interactive technology (exogenous): physiotherapists
- Person adopting or using the interactive technology (autogenous): patients of the physiotherapist

The aim of the persuasion change is a permanent change in behavior. The goal is to motivate the patients to finish and follow their personally designed training schedules for rehabilitation of their injuries.

The event includes the user, user context and technology context. In *FysiSystem* the use context is supported with features designed for the usage. The event is triggered with notification for timed training event, which then provides the training instructions. Additionally, whenever the user perceives the performance statistics or benefits of following their physiotherapy. The users can select how well they finished the training and view their accomplishments.

The user context varies, depending where the user is in their daily life. In preferable case the user is at their home or in a place suited for training, or in context where their cognitive load is low. However, the event is not as successful, if the user is in a place which is not suited for using the application, e.g. when driving a car. Technological context has few requirements for the *FysiApp;* the users must have battery on their smart phones and data access. However, as the system is currently only employed in Finnish physiotherapy clinic, it is likely the patients who have a smartphone have an access to internet [64].

The strategy for persuasion has been constructed from various points of view, both direct and indirect processes. The system triggers notification for each timed training even in schedule. This results the users viewing the training instructions, and finishing them in preferable scenario. The users can select how satisfied they were with their training by quickly selecting most suited emotion for proceeding from the selected training. Indirect persuasion methods include available additional information of their performance and accessing information how well other trainees have finished their training schedules. These users have been divided into groups with similar goals to enhance to motivation [20 p. 4].

In overall the strategy includes mainly following persuasive approaches for modifying the behavior: the goal-setting theory, social cognitive theory and transtheoretical model. Goal-setting theory can be seen from many views of the user interface. The users have a certain amount of training events in their schedule, and the system keeps in track of them. Feature which supports both, the goal-setting theory and transtheoretical model is events which occur weekly and after certain amount of finished scheduled exercises. The system provides visual and numeric information of the user’s progress, which is aimed to motivate the user to stay in the same performance level or even enhance the motivation.

The social cognitive theory has been implemented by creating a group-support feature, which allows the users view progress of other trainees and providing their own progress. Additionally, the users have interaction feature with their physiotherapist.

5.1.2 Software system characteristics

The principles aimed to reach with the *FysiApp* design have been categorized according to the model presented by Oinas-Kukkonen and Harjumaa [26]. The principles will be introduced under: Primary task support, Human-computer dialogue support, Perceived system credibility and Social influence.

Some of the introduced principles were excluded, because they were not found suited in this particular context. Additionally, some of the principles were excluded, so the system supports the introduced basic concept to its fullest potential, yet still keeping the system highly usable and providing positive user experience. In comparison to a system which would cause the user a high cognitive load by including all possible characteristics and features beneficial for persuasion.

The system requirements, i.e. implemented features, will be a result from selected desired system characteristics. From Primary Task Support following principles were chosen:

- Reduction: System reduces effort for performing their target behavior.
- Tunneling: System guides the users for target behavior.
- Tailoring: System provides tailored information for individual users and groups.
- Personalization: System provides personalized content for its user.
- Self-monitoring: System provides statistics and information of their performance and future schedule.
- Rehearsal: System provides the instructions for wanted behavior.

Reduction, tunneling and rehearsal are selected to ease the physiotherapy for the patients. The system should not be difficult or time consuming to use, instead to fulfill these characteristics, it should provide the information and needed features without effort. Other aspect of these characteristics is that the content and features of the system should be on track what the patient is required to do at current time and in the future. Tailoring, personalization and self-monitoring are selected, so the patient would have their system's content particularly relevant for themselves. Additionally, the system should contain only relevant content for the users, so these characteristics would also support the reduction, tunneling and rehearsal.

From Dialogue support following principles were chosen:

- Praise: System provides user feedback according to the user's performance in finishing tasks within the system.
- Reminders: System provides notifications for timed events.
- Liking: System's visual look should be pleasant for the users.

Praise and reminders are similar to the selected primary task support principles. Praise is used to provide feedback of their progress and how much they have left in their physiotherapy. Reminders should guide the users to stay on track of their training, similarly to the goal of reduction, tunneling and rehearsal. Liking should be reached with the system design, so the users would not be annoyed of the system design and would enjoy using the system, Rewards, similarity, social role and suggestion were left out from the dialogue support. Rewards and suggestions were not found suited for this concept, as the patient's training program is designed by professionals, and the patient's should not be encouraged to depart from the program, e.g. by doing more exercises than suggested. Social role and similarity would not be suited either, as the system is developed for health related issues.

From System Credibility support following principles were chosen:

- Trustworthiness: The information provided is truthful and unbiased.
- Expertise: The information provided is created with especial knowledge and expertise.
- Surface credibility: System has a competent and finished look and feel.
- Real-world feel: System provides information of the creators of the content and system.

System credibility support is essential for the system. The patient's should feel that the content of the system is reliable, and that they would feel safe and motivated to follow the designed content to reach their goals. Authority, third-party endorsements and verifiability were left out from the system credibility. These were left out, as these principles would not have been related in implementing prototype.

From Social support following principles were chosen:

- Social learning: System provides means for observing other users behavior and outcomes.'
- Social comparison: System provides means for comparing own results to other users' results.
- Social facilitation: System provides the means of accessing information of other users performing the behavior.
- Normative influence: System provides the option of gathering users together who has the same target goals.

These principles were chosen, because social support is one of the most efficient methods for persuasion [21]. However, as the system is health related, the social support principles should not enable recognition of the patients or allow patients view other patient's personal information. Thus, the system provides only numeric statistics and the users can view other patient's statistics of their finished exercises, and share their own if they choose to. Cooperation, competition and recognition verifiability were left out from the social support. These were left out for the similar reasons as rewards and suggestions, as the physiotherapy is not a competition and the content of the training programs are personal information.

5.2 Requirement definition for software qualities

In previous chapter we defined desired system characteristics for the system. The purpose of this chapter is to categorize these characteristics to functional and non-functional requirements for the system implementation.

5.2.1 Non-functional requirements

Non-functional requirements are more related to content of the system and the effect of it to the users. Additionally, non-functional requirements include visual aspects of the system and its navigation related characteristics.

For the content, following principles should be fulfilled for the system: reduction, tunneling, tailoring and rehearsal. To achieve these goals, the system should enable the content to be created by physiotherapist and to be received by the patient. The content also should be credible for the patients, so the system should fulfill the trustworthiness, expertise and real-world feel principles. However, for the content to fulfill these principles, the content should be designed by physiotherapeutic professionals.

Lastly, the system should have pleasant and finished look. These are the liking and surface credibility principles which can be reached by designing the systems user interface carefully and by following design principles, e.g. Nielsen's heuristics [47].

5.2.2 Functional requirements

The system should have functionality to support reduction, tunneling, tailoring and personalization. This should be done for the physiotherapist's and the patient's system. Firstly, the physiotherapist should be able to create the tailored and personalized content, which is also non-functional requirement of the system. Additionally, patient's system should have the functionality to support the patients in their goals, e.g. staying on track of their progress and providing them their future training program. Other essential function would be providing notifications according to the created training program, which is the reminders principle.

The system should be on track of the patient's training and provide statistics of their performance. Additionally, to fulfill social affiliation principles, the statistics should be provided to other patient's group performance statistics anonymously. These would fulfill the self-monitoring, social support, social comparison and social affiliation principles. The patients should be allowed to be grouped according to their goals, which is studied to increase motivation, if patients perceive they share same goals within a group and it fulfills the normative influence principle [20]. However, as the social features are anonymous, the division to groups must be done with the physiotherapist's system.

Lastly, the system should provide feedback according the user's performance, which would be the praise and self-monitoring principles. The feedback should be relevant for the user and personal, so it would also support transtheoretical model. This function should be implemented to occur after certain amount of finished exercises or after certain amount of time.

5.2.3 Selected features

According to the functional requirements, following features were selected to fulfill these requirements. These features should be included in the system's functionality in the prototype.

The users are given an ID number, which is the key to the application. It is used as a single log-in account, and after the usage, the application updates its content frequently. Thus, the user does not have to register to the system or download its content separately. The goal of this is to ease the patient's cognitive load, reduce the effort for the usage and guide the users to the wanted behavior with lower effort. Additionally, the system is tailored for the patient's tasks and content.

The system should update the content for the patients and physiotherapists frequently. This results the system to update its content after each accomplishment and modification from the patient or physiotherapist to a cloud service. The patients have their own accomplishments available for personal usage and if wanted – for the other users. Thus the system always stays on track of the patient's and patient's group's performance, with an asynchronous updating within the mobile application. Because of this, the patient can view statistics and information of their performance at all times, and the progress of their physiotherapy. Additionally, this also affects principles within social support, as the users can view and compare their own results after each accomplishment to their peers in real time, relying the content being up to date.

Physiotherapist's application does not have to be asynchronous, because they manage vast amount of patients and their changes to the programs are result of careful design of the patient's training program. However the system's messages should provide a notification within the system without delay.

As the users can access their own and their peer's performance statistics, the system enables them to be grouped. This is designed specifically for the physiotherapists, so they can do the division according to the same goals. This ensures the anonymity within the group, and enhancing the influence, as the users are working together for the same target goals [20].

Notifications will play a major part in the system. The users can receive messages from physiotherapists and they have a strict training schedule, which they should follow. Notifications should be implemented to occur at wanted times and encouraging the users view the content of the source of the notification. This also helps the user staying tract on their training program, and helps the patient for using the system only when required and essential information or tasks are available.

The last feature is based on transtheoretical model and various principles Oinas-Kukkonen presented. The system provides feedback according to the patient's performance at wanted times. These user feedbacks should be encouraging and effected by how well the user has performed. Additionally, as similarly to the notification, they should only be provided when they have some relevant information for the users. These user specific feedbacks provide a good chance for the patient self-monitor their performance, and receive feedback (possible praise) on how well they have done. As Kimura et al. pointed out, these kind of transtheoretical model messages can reduce the boredom of using the system [29, p. 4]. To add, the notifications reduce the risk of the user using the system when nothing is available, the overall user experience should be interesting, beneficial and possibly stimulating for the user.

5.2.4 User experience goals

UX goals are mainly related to non-functional requirements. System's fundamental goal regarding UX is to fulfil standardized usability qualities and achieve pleasant UX. The usability characteristics are specific for the context of use instead of being universally the same for all products [36]. Additionally, listing all possible characteristics related to user experience would not be relevant for this study. Thus, the most essential and influential elements of the user experience related to persuasiveness in this particular context of use are presented in this chapter.

Non-functional requirements for the system were related to the visual look, system credibility and to support user with their tasks. The functional requirements included vast amount of social affiliation related requirements. Both of these also had requirements related to personalization in content and features. Thus, the selected themes for UX goals for the system are:

- Social affiliation: interaction, identification and support
- Utility: usefulness and reliability
- Usability: accessibility, efficiency and guidance
- Aesthetics: stimulation and credibility
- Personalization: content, feedback and support

Social affiliation plays a major part in the user experience. It has been suggested in many studies, that the social support increases motivation for finishing the given tasks. Identification the user for the system usage and other users also increases the credibility and reliability of the system. The interaction with other users also must be implemented well to increase the social factor of the system.

The utility of the system includes the usefulness and reliability of the system. The user should perceive that the system is useful and has additional value compared to printed instructions. Otherwise the motivation for using or accepting the system for usage is compromised. Related to the usefulness is the reliability of the usage. The usefulness of the system decreases significantly if the system does not perform in a high level, e.g. unable to access information.

Usability is mentioned in the themes for its qualities: accessibility, efficiency and guidance. These actually make the system useful for the user. The system's goal should be to guide the user to finish their tasks and the usage of the system. Additionally, the system should be efficient to use and not waste user's time for simple operations. Accessibility describes that the system should be available and its content from databases as well at all times.

Aesthetics is a major factor for two reasons. Firstly, the system should be pleasant and enjoyable from visual point of view. This results the users more likely to enjoy using the system and not annoyed by using it. The second reason is related to credibility, perceived finished look 'n feel increases the credibility of the system, even though it does not have a direct effect in how well the actual code is implemented or how professional is the content of the system [25].

The whole purpose of the system is for the users to access their personal content via the system. Thus the content of the system must be accessible and personal with ease for the user. Additionally the feedback provided by the system should be personal and relevant for the user according to the transtheoretical model. Lastly, the provided support, e.g. praise, from the system should have a personal touch for the user.

6 DESIGN OF FYSISYSTEM

The PSD model adaption added some requirements for the system. These were prioritized in the design process of the system. The cross-platform aspect of the system was notified in the selection of used frameworks for the system design. To simplify the cross-platform design process for the case study, Model-Based Design has been adapted in this chapter [38; 45]. This approach was chosen, as it was considered to be effective framework for cross-platform design in the theoretical analysis of cross-platform design approaches in Chapter 3 [38]. This adaption enables the design process to be top-down, starting from higher level of abstraction. This results the implementation to be efficient and clear operation, as the features and interface has been defined in higher abstraction level in models.

The models used are based on Cameleon Reference Framework with the exception of excluding the adaption models, which would define the system's behavior and modification to the current context of use [45]. This approach was chosen, because the adaption models have been excluded from other recent studies in designing cross-platform systems, which have no intention for multi-targeting the system for various unpredictable contexts [46; 49]. To add, they were mainly focused in designing the system's user interface and how most essential features will be included to the system's design similarly to this case study.

These models include defining the domain of the system (concept and tasks) and context of use (user, platform and environment). According to these definitions the abstract interface will be constructed, which is used for designing the concrete user interfaces. Lastly, the interaction with these two systems will be presented from user's point of view and how it was implemented from the technical point of view.

6.1 Physiotherapist's application

The physiotherapist's application was originally planned to be purely desktop-based application, which enables the physiotherapists to create the training schedules and communicate with the patients. High-fidelity prototype of the system was implemented to gain feedback from physiotherapists in user studies, to examine detailed information of their opinions of the system and its features. The high-fidelity was implemented according to the presented original concept. Additionally to the desktop-application, low-fidelity prototype of the mobile application was implemented to support user's needs for Windows Phone 8.

6.1.1 User study

User studies were conducted for two physiotherapists for insight of their daily tasks, relation to patients and preferences. The structure of the user studies is clarified in Appendix A. The interview included questions about the participant's background and their experience in physiotherapy. It also included specific questions of their work habits, systems they currently use and about their interaction with patients. After the interview they were asked to finish tasks with an early version of the system. The prototype and its content was developed and created according to the original concept of the system. During the tasks they were asked to explain what they were perceiving and trying to do, and after the tasks they were asked for their opinions and suggestions.

The test users were 30 and 36 year old male physiotherapists with bachelor's degree in physiotherapy, with work experience of 4 and 12 years from the physiotherapy. Their work habits did not differentiate highly from each other. Neither of the participants did spend high amount of time using computers in their free time, approximately 2 hours weekly. Additionally, their work-related usage was also low, approximately 8 hours weekly. This includes creating training schedules for the patients, maintaining patient information and answering to emails. They used different system for each given task.

Their interaction with the patient is mainly personally in the office, but scheduling meetings is via email or phone. The participants told that they have approximately one meeting per week with the patient, but it is highly dependent from the patient and patient's needs. Some of the patients require assistance or physical operations by the trainer, which results in more meetings. The meetings are usually one hour, and both of the participants used electronic systems whilst the patients are in the office for writing down essential patient related information.

The participants then create the training schedule according to the information by using electronic system, from which the schedule is printed or emailed to the patients. The patients do not usually participate in creating the training schedule, but it is often modified according to the patient feedback in the upcoming meetings.

The tasks for the participants included nine individual tasks. These tasks were designed to test individual essential features of the system and to observe how efficient and usable the system is. The tasks were ordered in a logical order to pilot a basic use case of the system, which included: adding a new patient to the system, creating and modifying a training program step by step, sending a message and accessing the patient's performance.

Observed problems in finishing the tasks:

- One user did not found easily the add exercise button to the training schedule. Instead the user tried to double click dates from the calendar. The view includes high amount of options for the users, which might lead to high cognitive load for the user. Additionally no icons were used in the options.
- Both of the users struggled with adding another exercise to created training program at a specific time. Instead, they created another program at the same time and added the exercise to the created program occurring at the same time. From the physiotherapist's point of view it does not differentiate highly.

Observed problems in general:

- The lack of notifications when the system was loading or sending data to database.
- Adding an exercise schedule was found inefficient. This was because, according to the participants the requirement of setting program related information for each exercise. These include the date, time, weights and repeats of the exercise.

Rest of the tasks did not cause problems and were finished with ease, which include interacting with the patient, monitoring the patient progress, modifying created training schedule and managing patients. Additionally to the observed tasks, the participants mentioned the benefits of their current system and their preference to it due to their experience in using it.

Excluding malfunctions and problems within the system, the physiotherapists suggested two features, which would be essential for the system to be efficient and beneficial for the physiotherapists. Firstly, they should be provided an option for adding additional instructions for created individual tasks. Additionally the instruction specific variables, including weight, repeats and series, should not be limited to numeric values. This was due to the nature of the exercises, as these instructing variables are in many cases described in a way which cannot be informed with numbers, e.g. “do as many as you can”.

Another major feature suggested was adding their own instructions for exercises, or at least modification the existing exercises. These include adding pictures and writing the instructions, or videos.

6.1.2 Cross-platform configuration

The system is designed for desktop environments and mobile devices. The roles of these devices are complementary. The service delivery for the physiotherapists system is multichanneled. Both of the devise access the data and content, but some of the system's features are extracted from the mobile device. This configuration was selected, to optimize the usage of the system and optimizing the usage for each device. Due to the device organization, each platform is using its functionalities and capabilities in a best way suited.

The desktop environment includes all the functionalities for the system usage. This includes accessing patient information and interacting with the patients, also more complex and demanding tasks including creating patient specific training schedules and modifying them. As the real estate for the desktop environments is significantly higher than for mobile devices, the functionalities available for the mobile devices are not excluded from the desktop environment. Additionally, the usage of the system's features will not be limited to specific features, if the user does not have these two types of devices available.

The mobile devices have the most suited features included to their systems; interacting with the patients and viewing their progress in real-time. Additionally, the mobile devices provide the possibility of notifications for the users, whenever they are carrying their mobile phones. The interviewed physiotherapists estimated that they use computers in their work approximately 8 hours of total weekly, thus unless they modify their work-habits, they would not perceive possible notifications most of the time without the mobile notifications. Thus, the physiotherapists would receive notifications of messages without having to be near computer or having the desktop version open at all times.

These described features are also easily accessible and usable with a mobile device. Especially considering if the physiotherapist has over 10 patients, the mobile device scales to the amount well with these limited features. In contrary if the functionalities would be increased for the mobile device, the usability would inevitably decrease.

In conclusion, the desktop version is the primary for using the system, and the mobile device being the tool for accessing information quickly and for interaction with the patients. The desktop version can be employed on modern Linux and modern Microsoft Windows operating systems (Windows XP and later). These versions share the same functionality and content. As the mobile system is limited to the designed native applications, which is only limited to Windows Phone with described high-fi prototype.

6.1.3 Ontological models

The ontological models are meta-models for identifying the system characteristics for the given problem. They are independent of any specific platform or its limitations. The ontological model includes domain, context and adaption models.

Domain models describe the concept of the system and user tasks. The top-level concept of the *FysiSystem Trainer* is to:

- Create patient-specific programs by using the content of the system

- Monitor the patient's performance following the training program
- Interact with the patient by using the system

From the simplified concept, the tasks within the system are presented in use case -diagram (Figure 21; 22). Meta task-models for single use cases are presented in Appendix B. The final use case diagram differentiates from the original concepts with single use case (creating new exercise instructions). Otherwise, the made changes to the prototype did not have an impact on task models.

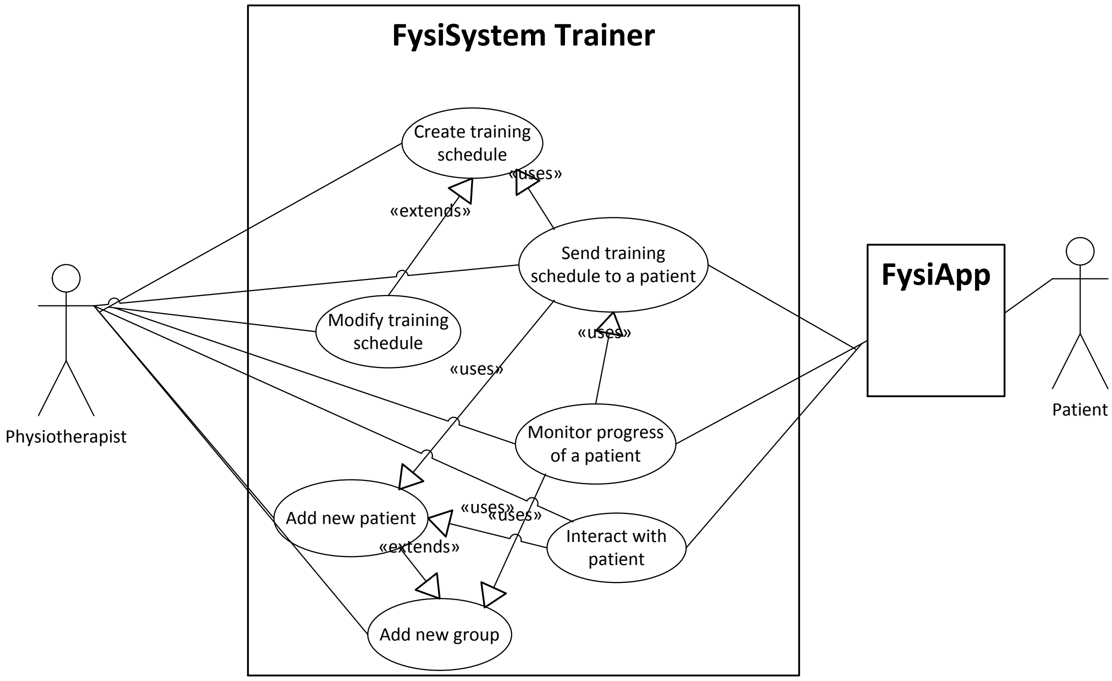

Figure 21: Desktop, Use Case Diagram

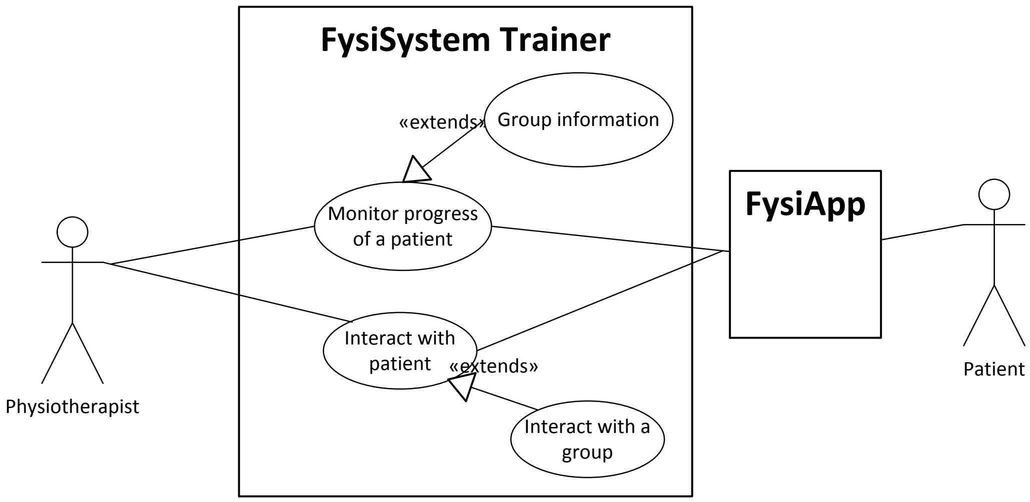

Figure 22: Mobile, Use Case Diagram

The context of use includes the user, platform and environment. The user and user's environment are related to the concepts and task models, which were studied in the presented user studies. These are taken in account in the task models.

The system is designed for desktop and mobile platforms. The desktop-platform solution does not differentiate from operating system to another in visual design or in the provided features. The visual design of the mobile environment varies from platform to another due to the suggested design guidelines by mobile device manufacturers. The functional design is similar from platform to another with some variations within the mobile devices.

6.1.4 Abstract interface

Abstract interface was formed according to the given concepts and task models. Additionally it presents the abstraction of the interface including the presentation of the functionalities and content, and the interaction with the system. Wireframes are used to define the abstract interface, which is used to create the implementation of the interface.

The wireframes of abstract main view of the system for desktop is presented in Figure 23 and for mobile devices in Figure 24.

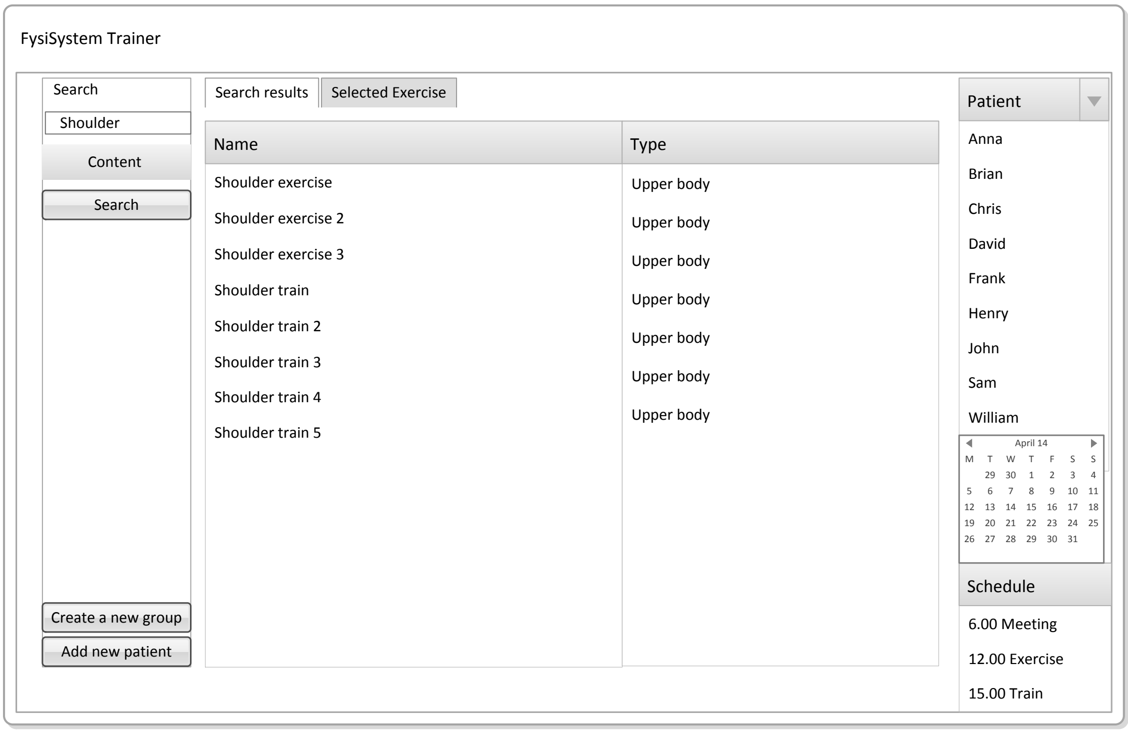

Figure 23: FysiSystem, Desktop

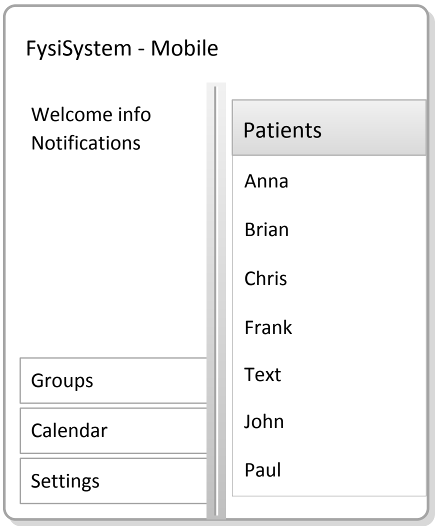

Figure 24: FysiSystem, Mobile

More detailed abstract interfaces for all functions are included in Appendix D and E. The mobile system design was done by using similar concepts as were used in the desktop abstract interface. Though, the designs of the tasks and navigations are not identical, due to the nature of mobile operating systems. The design for mobile device is designed to use the benefits of mobile devices, and concepts from the desktop solution were not forced if they were not suited for mobile device.

6.1.5 Concrete interface

The redesign of the system was done according to the abstract interface and ontological models. The concrete user interface is presented below for desktop environment (Figure 25). More specific interface images of all use cases are presented in Appendix G. The implemented prototype of the desktop application is designed and accessible for desktop and laptop computers, and its usage is most suited for it.

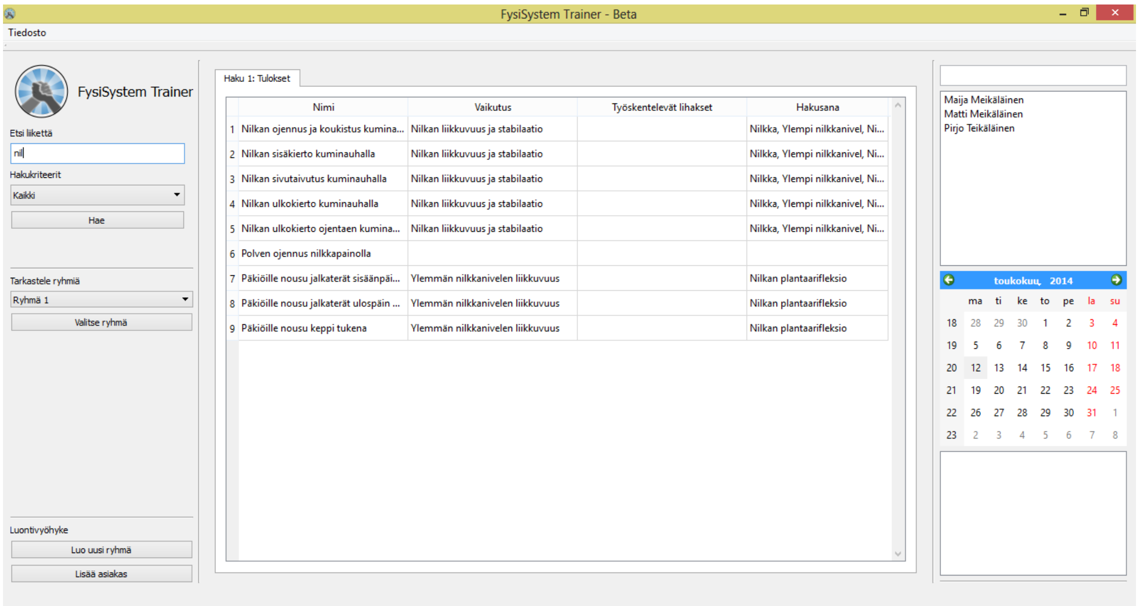

Figure 25: FysiSystem, Desktop version

The concrete user interface for mobile devices is presented below (Figure 26). More specific interface images of all use cases are presented in Appendix G. In contrary for the desktop design, the mobile device system is available in most of the possible contexts. It has high level of accessibility and efficiency for its specific tasks and additionally it provides notifications from interaction with the patients, which enables the user to receive information in typical work-environment when the user is not using the desktop system.

Figure 26: FysiSystem, Mobile version

6.1.6 Implementation

The desktop system was implemented with Qt Creator according to presented abstract interface and providing the necessary features. The mobile application was implemented with Microsoft Visual Express 2012 by using vast amount of provided Windows Phone 8 frameworks. For the mobile application code from the patient's implemented was re-used and modified to fit the purpose. This was possible due to similar tasks, for which the implementation was the same with minor modifications.

6.2 Patient's application

6.2.1 Cross-platform configuration

The patient's version of the system is specifically designed for mobile devices. The prototype was implemented for Windows Phone (high-fidelity) and Android (low-fidelity) operating system. But as their potential usage for the system does not differentiate highly, the cross-platform organization is designed identically for each device with no functional distribution. However, the system and its user-specific content is usable with various mobile devices at the same time and with differing operating systems, i.e. the system services are crossmedia systems.

6.2.2 Ontological models

The ontological models are meta-models for identifying the system characteristics for the given problem. They are independent of any domain or interactive system. The ontological model includes domain, context and adaption models.

Domain models describe the concept of the system and user tasks. Top-level concept of the *FysiApp* is to:

- Access training schedule and information of exercises
- Monitor performance
- Interact with the physiotherapist by using the application

As one of the main-goals of the system is to motivate the patient, the design and tasks are heavily affected by the system requirements presented earlier. Some of the tasks, e.g. monitoring performance, are due to the introduced system requirements. The tasks within the system are presented in use case -diagram (Figure 27):

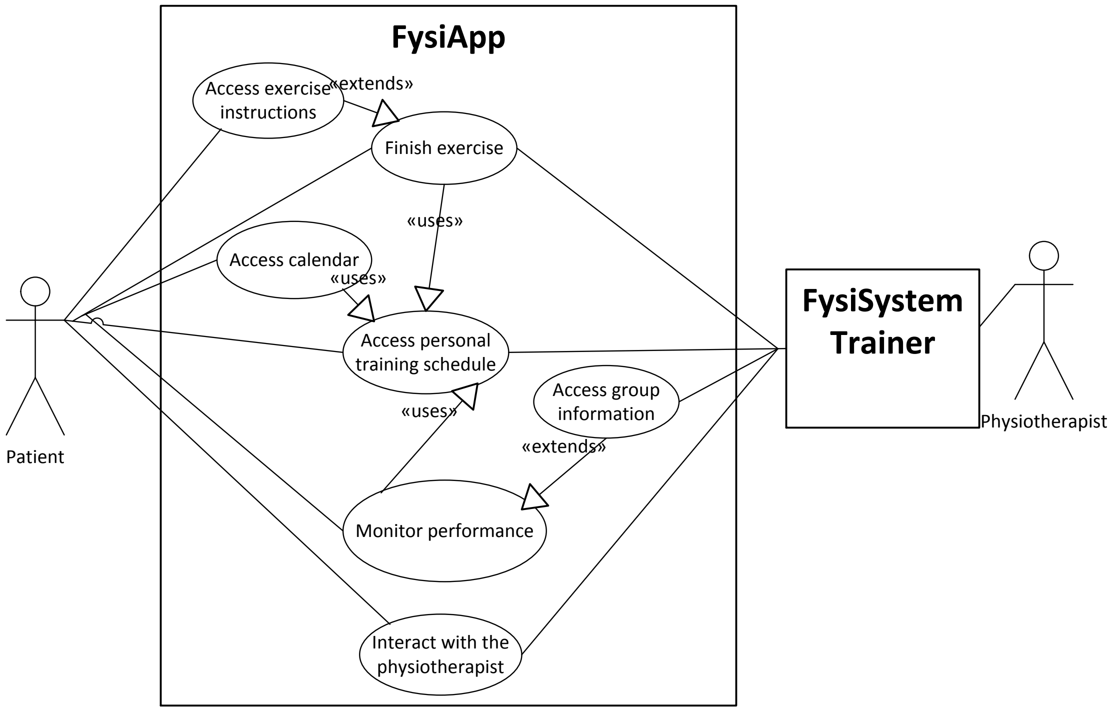

Figure 27: FysiApp, Use Case Diagram

The context of use includes the user, platform and environment. The user and user's environment are related to the concepts and task models, which were studied in the presented user studies. These are taken in account in the task models. All specific use cases are specified in Appendix C.

The system is designed for mobile platforms. The mobile environments visual design varies from platform to another. The functional design is similar from platform to another with some variations.

6.2.3 Abstract interface

Abstract interface was formed according to the given concepts and task models. Wireframes are used to define the abstract interface, which was used to create the implementation of the interface. Additionally, the abstract interface includes the given non-functional and functional requirements developed from the persuasive methods chosen for the system design. According to the abstract interface, the concrete interface is formed, which is environment and platform dependent.

Similarly to task model, the abstract interface does not highly concentrate in the content of the features. Thus, the listed non-functional or functional requirements for persuasive effect are not particularly addressed in the abstract interfaces. However, positive user experience is a key-factor in persuasive systems. The wireframes' design must fulfill the basic principles of user interface design and be intuitive for the users, so the actual content of the system features and information is perceived by the users in positive manner.

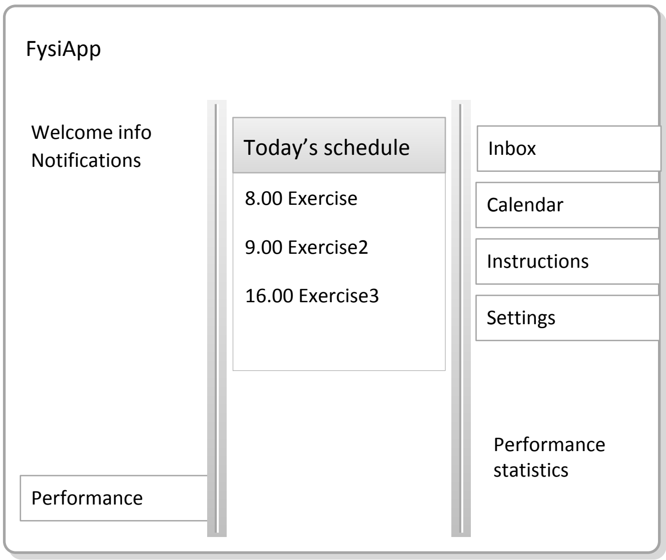

Figure 28: FysiApp, the main view

Main and the opening view of the system is presented in Figure 28. The detailed wireframes of all views of the system are available in Appendix F.

6.2.4 Concrete interface

The concrete interface of the implemented prototypes will be presented below in figures 29 and 30. All views are presented in Appendix H. The Windows Phone 8 prototype has high functionality and supports the described features, whereas the Android is low-fidelity prototype, with some of the features supported.

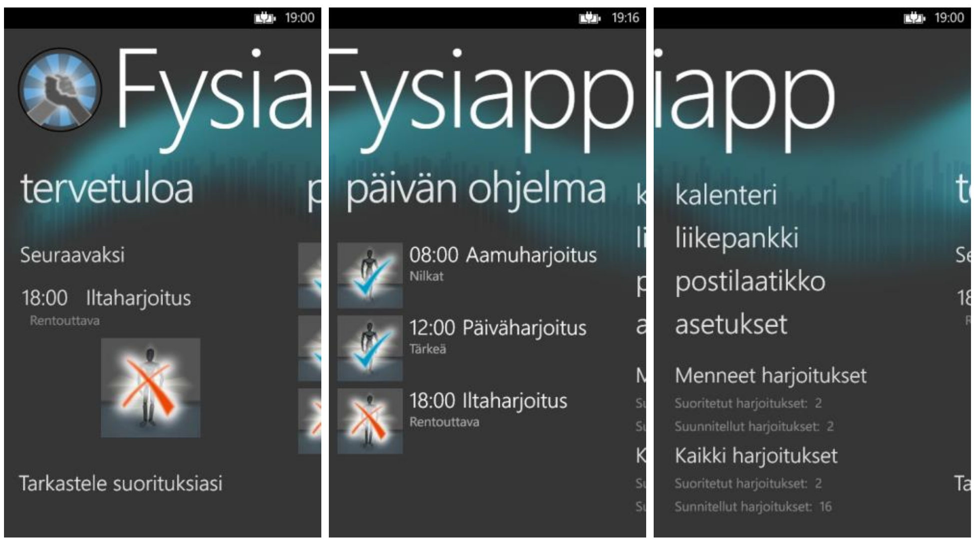

Figure 29: Mainview of the Windows Phone 8 prototype

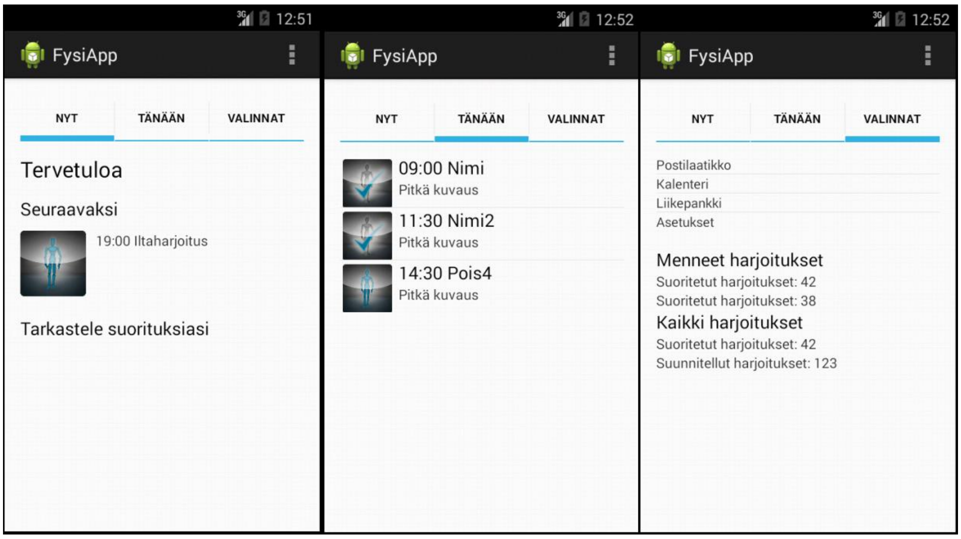

Figure 30: Main view of the Android prototype

Icons' fundamental design was the same for both of the prototypes. The icons had final design modifications, so they would fit to the overall scheme of the applications and fit to operating system's design guidelines. Panorama view (Windows Phone) and Active tabs (Android) give the possibility of fulfilling the design of the abstract interface, as they do not offer identical views for providing vast amount of features and information in one view. However, the design of the system has similar navigation model for both of these approaches because of this made selection.

The other views of the system follow similar concept, or they are just limited to one page. Only major difference is in the application bar used in Android application. The application bar provides information of possible navigations back and additional menu, which is used in this case for providing the information of the application.

6.2.5 Implementation

The Windows Phone 8 application was implemented with Microsoft Visual Express 2012 by using vast amount of provided Windows Phone 8 frameworks. The Android application was implemented with Eclipse development environment, extended with plug-in Android API libraries and developer tools from Android SDK. As the used database-storage was Windows Azure, additional libraries were used for the Android implementation for the connectivity.

On the abstract level, the program was developed by using model-view approach, which was usable in similar structure for both of these prototypes. As the views and models had similar designs, the classes implemented for the models and additional feature-classes had almost identical definition, excluding the language specific syntax required for C# and Java.

6.3 System connectivity

6.3.1 Technical approach

The runtime infrastructure is divided from the implemented systems and Windows Azure cloud storage. Accessing the data within the data storage is implemented by using simple SQL connectivity. Windows Azure offers especial method for mobile devices called *MobileServices*, which is used for Windows Phone devices. Similar approach would be possible for Android, by using the available libraries for accessing the Microsoft Azure backend cloud service [65]. However, the connectivity was not fully supported with this Android low-fidelity prototype, so the exact implementation will not be clarified in this chapter. For the desktop system simple SQL methods are used for accessing the data. [66]

Windows Phone applications for the patients and physiotherapists use asynchronous operations for staying up to date when the user navigates within the application or uses its functions, e.g. sends a message. In result of this, the users have the most recent information at all times when using the system with their mobile devices.

When the user navigates from the application, the application will be in *dormant*-state. Because of this, the application's threads will be stopped, but can be quickly reactivated from previous state. If the user closes the application or the user have too many applications running at the same time, the operating system sets the application to *tombstone*-state. In this scenario, the user cannot reactivate the system to previous state. However, after the launch, the user accesses identical data compared to *dormant*-state. [66]

For the desktop-application the data access is limited. This is due to the fact that during the implementation of the prototype and testing the prototype, there were performance-related issues, which caused the application pause or operate slowly when the database-access was used. Additionally, it is not crucial for the desktop-application to stay up to date to the progress individual patients in real-time. Thus, the system updates the patients' performance only when the system is launched. However, the messages update has been prioritized. Received messages will be updated within the system frequently, and whenever the user is using inbox-feature of the system.

6.3.2 Interaction

The interaction within the system with patients and physiotherapist is enabled by using the technical solution described. The interaction for patients and physiotherapists will be described individually for each other in this chapter.

The physiotherapists have the option of interacting with all their created patients by using their preferred platform. Additionally to sending and receiving messages from the patients, the physiotherapists have several features for adjusting the patient's system. The physiotherapists can access and monitor the progress of their patients and modify the training program in real time for their patients according to patients' preferences. The made changes will be available for the patient after the modifications. As the physiotherapist can access the full information of the patients, they can divide the patients into groups, which is recognized to be beneficial for increasing motivation.

The patients can directly interact with their personal physiotherapist and request changes for their training program, reserve meetings or ask for assistance via the application. The patients have the freedom of choosing whether they choose to join the groups created by their physiotherapist. If they choose to use the group support feature, they will be anonymous by using selected username within the system. The group support does not provide any personal information of the other group members, their disability or their training program. Instead, the members of each group will be ranked according how well they follow their personal training schedule by numeric information of their past exercises and success percent of finishing given exercises.

7 EVALUATION OF FYSISYSTEM

This chapter will present the desired behavioral changes of the system, which would be required to study, before developing the system further. The experienced persuasion would be essential to study comprehensively from the ethical point of view, as the experienced persuasion cannot be predicted [10]. Additionally, studying how well the experienced behavioral changes correspond the desired behavioral changes would be suited. This chapter also includes the limitations of the implemented prototype. This includes comparison to desired system characteristics and technical limitations of the prototype.

7.1 Persuasive evaluation

7.1.1 Outcome/Change Matrix

Table 4: O/C Matrix for FysiSystem [26, p. 2]

	Act of complying	A behavior change	Attitude Change
Formation	Takes the system in use.	Studying given exercise instructions and exercising.	Acknowledging the benefits of physiotherapy.
Alteration	Accepts the used system and given instructions from the physiotherapists.	Finishing scheduled exercises available from the system.	Increasing interest in physical health and ways in improving it.
Reinforcement	Shows initiate in the process, e.g. interacts with the physiotherapist with the system.	Does not require supervision for following training schedule.	High interest in staying in good physical length and additional exercising.

The act of complying includes accepting the technology. According to presented theories, some users have difficulty in accepting new technology, especially when it might be seen out of place. Additionally, if the user decides to take the system in use and uses it in daily life, it shows interest in overall for the wanted behavioral change. Once the user has taken the system in use, the last act of complying is that the user shows initiate towards the physiotherapist.

The formation and alteration of behavioral change is divided into two steps. Firstly the user is interested in give training schedule and studies it. Next, the user starts actually finishing the given exercises. Attitude change formation and alteration are from broader aspect. They take in account users' attitude to overall health related issues, which are essential for the person to stay in good physical fit and to not be disabled from the work-life. However, these attitude changes cannot be guaranteed and they are very subjective to each person.

The behavioral and attitude reinforcement outcomes have some prerequisites from the actual physiotherapy. In order the reinforcement to be successful, it is likely to predict that the user should perceive results from the training.

Measuring the persuasiveness of the system and how these goals have been reached would not be possible with the implemented prototype of the system. The users (physiotherapists or patients) might not accept the technology in first place, which would result in the failure of the actual study. Additionally, the actual measurement of each individual goal would demand extensive study in the specific area and knowledge of physiotherapy in general.

7.1.2 Fulfilment of the persuasive principles

This chapter is focused how the introduced persuasive principles were reached and in what level in the implemented prototypes. The characteristics are evaluated under the principles they belong according to the PSD model introduced [26]. The system design is presented in Appendixes G and H.

Primary Task Support:

- Reduction: The implemented prototype provides their specific schedule in real-time updated with their specific instructions to their exercises.
- Tunneling: The implemented prototype guiding to the target behavior is limited. This is due to the implementation of the notifications was not flawless, thus the users might not be following or acknowledging their current schedule.
- Tailoring: The implemented prototype does provide the group-specific information and their tailored training schedule.
- Personalization: The implemented prototype does provide the personalized content for their users. However, the system does not allow the users to customize the user interface or the features according to their preferences.
- Self-monitoring: The implemented prototype does provide statistics and information of their performance and future schedule.
- Rehearsal: The implemented prototype does provide the instructions for wanted behavior.

Dialogue:

- Praise: The implemented prototype does offer user feedback according to the user's performance within the system. However, the feedback is quite limited, and might not reach the full potential which praise might offer.
- Reminders: The implemented prototype does not provide notifications for timed events.
- Liking: The implemented prototype's visual look is subjective, and cannot be evaluated at this point.

System Credibility:

- Trustworthiness: The implemented prototype's information is limited to statistics of the user's performance, and the content is created by physiotherapist student. Thus, it could be described as trustworthy and unbiased.
- Expertise: The implemented prototype's content is created by physiotherapist student. Thus, the information is designed with especial knowledge.
- Surface credibility: The implemented prototype's visual look is subjective, and cannot be evaluated at this point.
- Real-world feel: The implemented prototype's provides information of the creators of the content and system

Social support

- Social learning: The implemented prototype's does provide means for observing other user's progress.
- Social comparison: The implemented prototype's does provide means for observing other user's progress and comparing own performance to theirs.
- Social facilitation: The implemented prototype's does provide means for observing other user's progress.
- Normative influence: The implemented prototype's does provide the option for creating groups according to their goals and interests.

7.2 Limitations of the system

This chapter introduced the most crucial limitations of the created prototype of the system. Originally, it was defined that it would be essential for the system to provide notifications. However, the implementation of the system does not provide notifications for the users mobile applications.

Other major limitation was the request from the physiotherapists in the user study that they should be able to add their own instructions. The implementation of the described feature would have demanded high volumes of modifications to the database and to the patient's mobile application. Additionally, the implementation of the system had more

crucial issues, which were prioritized over this particular feature, e.g. flawless asynchronous updating.
The patient's system is purely based on theoretical background, so the persuasion of the mobile system is currently undefined. Thus, it would be unethical to use the system for commercial purposes before comprehensive analysis of the system [10]. The user experience of the system is also unclear, as pragmatic values of the user experience are perceived and experienced by the users. Consequently, as the user experience is a major factor in persuasion, it would be essential to study and improve user experience related issues as well [34].

8 CONCLUSIONS

8.1 Summary

In this thesis, we introduced theoretical background of persuasive technology and cross-platform design. We adapted some of the introduced frameworks for designing persuasive and cross-platform systems in a case study. The concept of the system for the case study was predefined, and the goal of the case study was to implement a high-fidelity prototype of the system.

The adaption was done in two phases. First, we defined the system characteristics by using a PSD model originally introduced by Oinas-Kukkonen [26]. Furthermore, in the definition of the characteristics, the results of publications studying persuasive systems were considered in the definition of the used persuasive methods [29; 40]. We aimed to reach the desired system characteristics by defining non-functional and functional requirements. The system requirements led to individual features and user experience related themes for the implementation of the prototype

As the system consisted of several components and user groups, a cross-platform design framework was utilized for the implementation of the system. The selected approach was a Model-Driven Development loosely based on the Cameleon Reference Framework for the implementation. The user experience initial framework for cross-platform design by Wäljas et al. [38] was considered in the design process to tackle cross-platform related problems.

Lastly, the implemented prototype was presented and discussed. These included the definition of what would be the desired persuasive effects of the system and how these could be studied. Additionally, the limitations of the system were clarified. This included comparison to the predefined persuasive characteristics: how they were implemented and what of the characteristics were not included in the prototype. However, the evaluation of cross-platform user experience or the persuasive effect was not possible, because they are based on how the users perceive and experience the system, as user studies for the mobile application were not conducted.

8.2 Discussion

The used frameworks proved to be an effective asset in developing the case study. The persuasive framework for designing the system provided the desired characteristics and reasoning for the included system characteristics. Other beneficial factors included simplifying the design process of the system. Even though the benefits of using the PSD model are significant and reduce the resources required for designing the features of the system, following it explicitly would not be effective for implementing a prototype.

Some of the features included in the PSD model adaption would have demanded significantly higher effort to fulfill, compared to a simplified version of the system. This also can be seen from the developed prototype, as some of the most essential features according to the analysis were excluded due to their problematic implementation. One of these is notifications, which would have been utilized for received messages and timed events, i.e. scheduled exercise. Additionally, the feedback for the patients was limited to occur according to statistics and their design should be more personalized to fulfill the transtheoretical model's potential for persuasion.

The used Model-Driven Development (MDD) proved to be an effective tool for designing the system. The introduced cross-platform user experience framework was used to divide the functionality logically for components, which eased to design process of the system [38]. The first steps of the MDD were found clarifying in designing the overall structure for both of the system components (Patient's and Physiotherapist's system). For the physiotherapist's system MDD proved to be well suited for designing the system for two significantly differing platforms.

Even though the Android and Windows Phone design principles have significant differences, the created Abstract Interface suited both of the mobile operating systems. The Abstract Interface design was purposefully created to support both their potential and available navigation models. However, it could be said that under the visual components they have a vast amount of similar concepts used in their operating system design, i.e. in the wireframe of the user interface.

Another essential aspect of mobile devices' implementation was the similarity of their code structure. As the user interface and features were implemented according to the same designed Abstract Interface and User Task Model, their implementation in used class structure and models were almost identical, excluding the difference in syntax for Java and C#, and used adapters for setting the views. A major reason for this is that both of the development environments enable the developers to use a similar model-view approach for the implementation. However, the benefits of MDD for implementing native code without generation was not emphasized or studied in the used publications in this thesis.

Including users in the development process of persuasive system was identified as one of the most essential elements for designing persuasive systems. The user's reaction towards the system or their behavioral change cannot be determined purely using a theoretical background. Thus, user studies of the fundamental concept, i.e. interactive physiotherapeutic system, would be beneficial, before implementing the system to its final form. User studies were not done in this thesis, due to limited resources and as the originally arranged physiotherapist clinic did not find potential patients willing to participate in the user study in reasonable time. One of the reasons for this was the high-fi prototype was only available for Windows Phone 8. Studying the persuasion would not have been possible without limiting the user group to physiotherapy patients, as the experienced physiotherapy was defined to be an essential factor in desired behavioral change.

REFERENCES

[1] Fogg B.J. Persuasive Technology: Using Computers to Change What We Think and Do.US, 2002, Morgan Kaufmann. 283p.

[2] Fogg B.J. & Eckles D. Mobile Persuasion: 20 Perspectives on the Future of Behavior Change. Stanford, California US 2007, Stanford Captology Media. 166p.

[3] POS Goes Digital: Evolution of the In-Store Shopping Experience. [WWW]. [accessed 20.2.2014]. Available: http://www.gridleyco.com/wordpress/wp-content/uploads/2012/03/POS-Goes-Digital-Evolution-of-the-In-Store-Shopping-Experience1.pdf

[4] Key A., Wac K., Ferreira D., Tassini K., Hong J. & Ramos J. Getting Closer: An Empirical Investigation of the Proximity of User to Their Smart Phones. UbiComp'11. Beijing, China.

[5] Franz M. & Ridout T. Political Advertising and Persuasion in the 2004 and 2008 Presidential Elections. 2010, Sage Publications. 38p.

[6] Chang C. The Influence of Ambivalence Toward a Communication Source: Media Context Priming and Persuasion Polarization. 2013, Sage Publications. 16.

[7] Anne-Marie Brady. Mass Persuasion as a Means of Legitimation and China's Popular Authoritarianism. 2009, Sage Publications. 25p.

[8] Singley K., Lai J., Lei K. & Jung-Mu T. BlueReach: Harnessing Synchronous Chat to Support Expertise Sharing in a Large Organization. Florence, Italy 2008, CHI 2008. 8p.

[9] Juhlin O. & Östergren M. Time to Meet Face-to-Face and Device-to-Device. pp. 77-80.

[10] Davis J. Design Methods for Ethical Persuasive Computing. California, USA, 2009, ACM, pp. 1-8.

[11] Scholl W. The socio-emotional basis of human interaction and communication: How we construct our social world. 2013, Sage publications. 33p.

[12] Ajzen I. The theory of planned behavior. Organ Behav Hum Decis Process. pp. 179–211.

[13] Oinas-Kukkonen H. A foundation for the study of behavior change support systems.

[14] Passos C., Cruzes D. & Mendonca M. Applying Theory of Reasoned Action in the Context of Software Development Practices Insights into Team Intention and Behavior.

[15] Fishbein & Ajzen: http://people.umass.edu/aizen/f&a1975.html (Fishbein M, Ajzen I (1975) Belief, attitude, intention, and behavior: an introduction to theory and research. Addison-Wesley, Reading)

[16] Davis FD. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quart 13. pp. 319–339.

[17] Davis FD. User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. 1993 MIS Quart 27. pp. 425–478.

[18] Venkatesh V., Morris M., Davis GB. & Davis FD. User acceptance of information technology: toward a unified view. 2003 MIS Quart 27. pp. 425–478.
[19] Bandura A. Self-efficacy: Toward a Unifying Theory of Behavioral Change. Standford University, US. 1977 Psychological Review, Vol. 84, No. 2, pp. 191-215.
[20] Lunenburg F. Goal-Setting Theory of Motivation. International Journal of Management, Business, and Administration. Volume 15, Number 1, 2011.
[21] Bandura A. Social cognitive theory: An agentic perspective. Asian Journal of Social Psychology, 199. Volume 2. pp. 21-41.
[22] Ferron M. & Massa P. Transtheoretical Model for Designing Technologies Supporting an Active Lifestyle. CHIItaly ’13, Trento, Italy 2013. ACM.
[23] Eslambolchilar P., Wilson M., Oakley I. & Dey A. PINC: Persuasion, Influence, Nudge and Coercion Through Mobile Devices. 2011 Canada.
[24] Berdichevsky D. & Neuenschwander E. Towards an ethics of persuasive technology. Communications of the ACM. Volume 42, Issue 5, May 1999. New York NY, USA ACM. pp. 51-58.
[25] Oinas-Kukkonen H. & Harjumaa M. Communications of the Association for Information Systems: Persuasive Systems Design: Key Issues, Process Model, and System features. 2009.
[26] Oinas-Kukkonen H. Requirements for Measuring the Success of Persuasive Technology Applications.
[27] Roto V. & Rautava M. User Experience Elements and Brand Promise. 2008. [WWW]. [accessed 25.2.2014] Available: http://research.nokia.com/files/UXelements-v2.pdf.
[28] Davis J. Design Methods for Ethical Persuasive Computing. California, USA, 2009, ACM, pp. 1-8.
[29] Kimura H., Ebisui J., Funabashi Y., Yoshii A. & Nakajima T. iDetective: A Persuasive Application to Motivate Healthier Behavior Using SmartPhone. Waseda University. Okubo, Shinjuku-ku, Tokyo, Japan.
[30] Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., and Landay, J. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In Proceedings of the 27th international conference on Human factors in computing systems, ACM Press, New York, NY, 2009, pp. 1043 – 1052.
[31] Cheng Q., Juen J., Li Y., Prieto-Centurion V., Krishnan J. & Schatz B. GaitTrack: Health Monitoring of Body Motion from Spatio-Temporal Parameters of Simple Smart Phones.
[32] Press Release: Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC. IDC Corporate. [WWW] [accessed 1.4.2014] http://www.idc.com/getdoc.jsp?containerId=prUS24676414
[33] Oulasvirta A. When Users “Do” the Ubicomp. University of California at Berkley. pp. 4.

[34] Hassenzahl, M. 2004. The thing and I: Understanding the relationship between user and product. In Funology, from Usability to Enjoyment, pp.31-42.

[35] Väänänen-Vainio-Mattila, K., Väätäjä, H., Vainio, T. 2008. Opportunities and Challenges of Designing the Service User eXperience (SUX) in Web 2.0.In Saariluoma, P. and Isomäki, H. (Eds.), Future Interaction Design II, Springer.

[36] ISO-9421-11: Guidance on Usability. [WWW]. [Accessed 2.5.2014] Available: http://www.usabilitynet.org/tools/r_international.htm#9241-11.

[37] Xanthopoulos S. & Xinogalos S. A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. pp. 213-220.

[38] Wäljas M., Segerståhl K., Väänänen-Vainio-Mattila K. & Oinas-Kukkonen H. MobileCHI'10 Septermber 7-10, 2010, Lisboa, Portugal ACM. pp. 219-228.

[39] Antila V. & Lui A. Challenges in Designing Inter-usable Systems. INTERACT 2011, Part 1, LNCS 6946. pp. 396-403.

[40] Segerståhl, K. & Oinas-Kukkonen, H.: Distributed User Experience in Persuasive Technology Environments. In: Y. de Kort et al. (Eds.), LNCS 4744, Persuasive 2007, Springer-Verlag, Berlin Heidelberg, Germany (2007).

[41] Qt: Supported Platforms. [WWW]. [Accessed 17.2.2014] Available: http://qt-project.org/wiki/Category:Qt-Supported-Platforms

[42] Kieffer S., Coyette A. & Vanderdonckt J. User Interface Design by Sketching: A Complexity Analysis of Widget Representations. EICS '10 Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems. New York, NY, USA 2010, ACM pp. 57-66.

[43] Sangiorgi U., Beuvens F. & Vanderdonckt J. User Interface Design by Collaborative Sketching. DIS '12 Proceedings of the Designing Interactive Systems Conference. New York, NY, USA 2012, ACM pp. 378-387.

[44] Motti V., Raggett D., Van Cauwelaert S. & Vanderdonckt. Simplifying the Development of Cross-Platform Web User Interfaces by Collaborative Model-based Design. SIGDOC '13, September 30- October 1, 2013, Greenville, North Carolina, USA. pp. 55-64.

[45] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Inter. with Comp.* 15, 3 (2003), pp. 289–308.

[46] Aquino, N., Vanderdonckt, J., Panach, I., and Pastor, O. Conceptual Modelling of Interaction. In Handbook of Conceptual Modelling: Theory, Practice, and Research Challenges, Embley, D. Thalheim, B. (eds.), Chapter 3. Springer-Verlag, Berlin (2011), pp. 335–358.

[47] Nielsen J. & Molich R. Imrpoving a human-computer dialogue. Magazine, Communications of the ACM, Volume 33 Issue 3. New York, USA 1990. pp. 338-348. http://dl.acm.org/citation.cfm?id=77486

[48] Balagtas-Fernandez F. "Model-Driven Development of Mobile Applications", IEEE 2008, pp. 509-512.

[49] Diep C-K., Tran Q-N. & Tran M-T. Online Model-driven IDE to Design GUIs For Cross-platform Mobile Applications. SoICT ’13, December 5-6 2013, Danang, Vit Nam. ACM. pp. 294-300.

[50] Qt. [WWW]. [Accessed 17.2.2014] Available: https://qt-project.org/doc/qt-5/supported-platforms.html

[51] Microsoft: Panorama control design guidelines for Windows Phone. [WWW]. [Accessed 17.2.2014] Available: http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202912(v=vs.105).aspx

[52] Android Developers: Android Design. [WWW]. [Accessed 17.2.2014] Available: https://developer.android.com/design/index.html

[53] Microsoft: Panorama control design guidelines for Windows Phone. [WWW]. [Accessed 17.2.2014] Available: https://dev.windowsphone.com/en-us/design/principles.

[54] Android Developers: Pure Android. [WWW]. [Accessed 1.5.2014] Available: https://developer.android.com/design/patterns/pure-android.html

[55] Microsoft: App bar for Windows Phone. [WWW]. [Accessed 1.5.2014] Available: http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff431813(v=vs.105).aspx

[56] Android Developers: App Structure. [WWW]. [Accessed 1.5.2014] Available: https://developer.android.com/design/patterns/app-structure.html

[57] Dina Helmy: UX/UI Guidelines for Windows Phone 8. http://blogs.msdn.com/b/africaapps/archive/2014/03/08/ux-guidelines-for-windows-phone-8.aspx

[58] Microsoft: Displaying popups. [WWW]. [Accessed 1.5.2014] Available: http://msdn.microsoft.com/en-us/library/windows/apps/hh738362.aspx

[59] Android Developers: Dialogs. [WWW]. [Accessed 1.5.2014] Available: http://developer.android.com/design/building-blocks/dialogs.html

[60] Tait H. Motivation and successful physiotherapy. Australian Journal of Physiotherapy. Volume 6, Issue 3, November 1960, Pages 107–114.

[61] Vong S., Cheing G., Chan F., So E. & Chan C. Archives of Physical Medicine and Rehabilitation Volume 92, Issue 2, February 2011, Pages 176–183.

[62] PhysioTools. [WWW]. [accessed 5.4.2014]. http://www.physiotools.com/

[63] Physiofile. [WWW]. [accessed 5.4.2014]. http://www.physiofile.fi/

[64] Official Statistics of Finland (OSF): Use of information and communications technology by individuals [e-publication]. ISSN=2341-8710. Helsinki: Statistics Finland [accessed: 23.4.2014]. Available: http://www.stat.fi/til/sutivi/2013/index_en.html.

[65] Microsoft Azure: Get started with Mobile Services. [WWW]. [accessed 5.4.2014]. http://azure.microsoft.com/en-us/documentation/articles/mobile-services-android-get-started/?rnd=1

[66] Windows Phone: App activation and deactivation for Windows Phone 8. [WWW]. [accessed 5.4.2014]. http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff817008(v=vs.105).aspx

APPENDIX A: USER STUDY STRUCTURE (PHYSIOTHERAPIST)

The user study was done in finnish according to presented structure.

Interview: Part 1

1. Age
2. Education
3. Profession
4. How long have you worked as a physiotherapist
5. Do you meet all new patients personally?
6. Does the patient provide their medical information before first meeting?
7. How often do you have meetings with patients?
8. How long does a single meeting approximately last?
9. How do you gather notes in patient meetings? With pen or with electroni device?
10. How do you create a training program for a new customer?
11. Do you plan training program before the first meeting?
12. Does the patient participate in the creating process of training program?
13. In what form the patient receives the training program? Via email, as a print?
14. How do you try to motivate patients to follow their training program?
15. Do you use any system for managing the patient information?
16. Do you use any system for following patient's progress and for what purposes?
17. Do you use any system for creating training program? What, and for what purposes?

Tasks:

1. Create a new patient to the system
2. Create following training program for created patient:
    a. Predefined 2 exercises, series, weights and repeats
    b. Program is repeated every day for a week at 9.00
3. Modify the created training program's Monday's and Tuesday's weight to 4 kg.
4. Add new exercise to the training program
5. Add another exercise to be repeated every day
6. Send a message to the created patient
7. Send created training program to the patient
8. Search the patient's next week's Wednesday's schedule
9. Verify if the patient has finished any exercises

Interview: Part 2

1. Would you prefer creating training program patterns to be copied for various patients?

2. Do you think it would be nesessary to modify the exercise instruction's images or current information?
3. Do you think it would be nesessary to copy and paste the whole training program from one to another.
4. What did you think about naming training program items?
5. Is system for just desktop environment enough for your needs?
6. Would you think additional system for mobile devices would be useful for interacting with the patients?
7. Would you like to modify the system's current method for creating training programs?
8. What would you change from the current prototype?
9. What was poorly implemented in the prototype?
10. What was well implemented in the prototype?

APPENDIX B: SYSTEM USE CASE DIAGRAMS (PHYSIOTHERAPIST)

The detailed use case diagrams for physiotherapist will be presented this appendix.

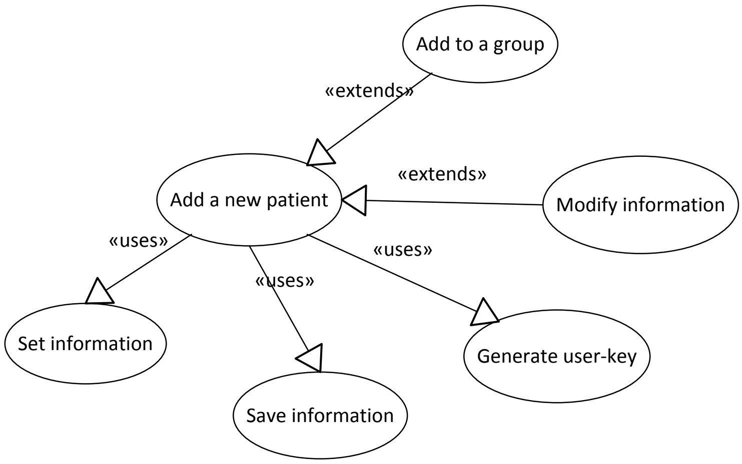

Figure B. 1: Add new patient to the system

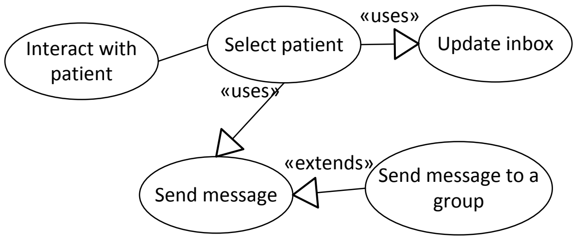

Figure B. 2: Interact with patient

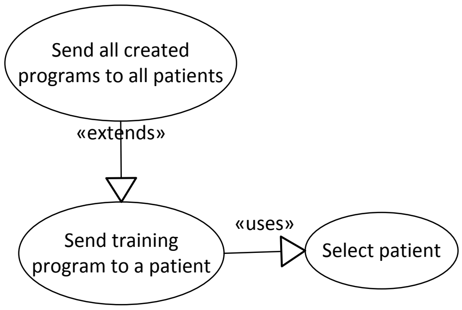

Figure B. 3: Sending training program to a patient

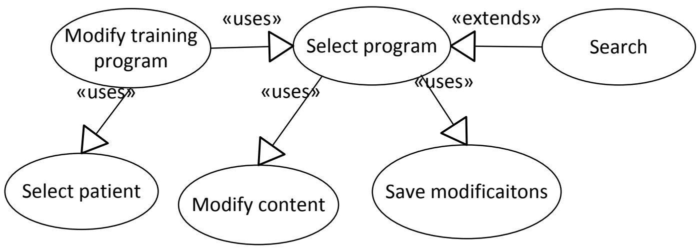

Figure B. 4: Modify training program

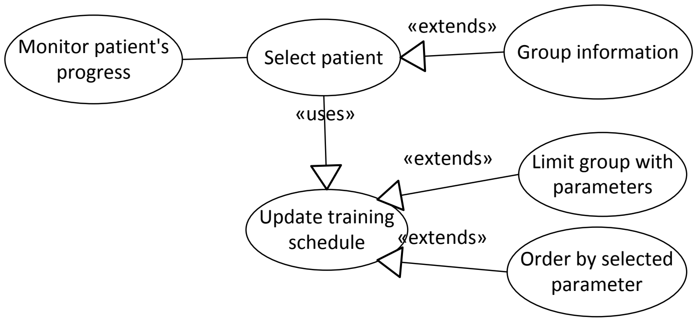

Figure B. 5: Monitor patient's progress

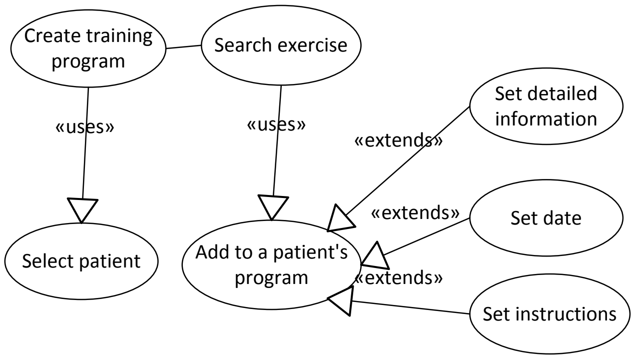

Figure B. 6: Create training program

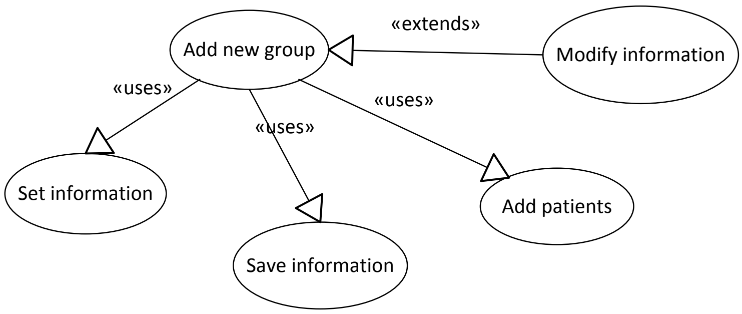

Figure B. 7: Create groups for patients

APPENDIX C: SYSTEM USE CASE DIAGRAMS (PATIENT)

The detailed use case diagrams for patient will be presented this appendix.

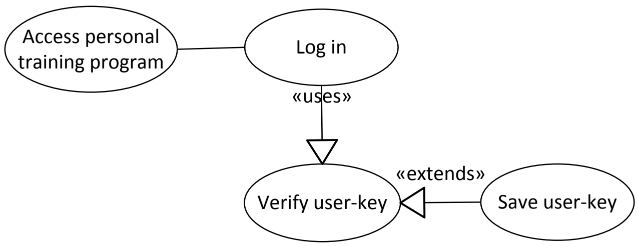

Figure C. 1: Access personal training program

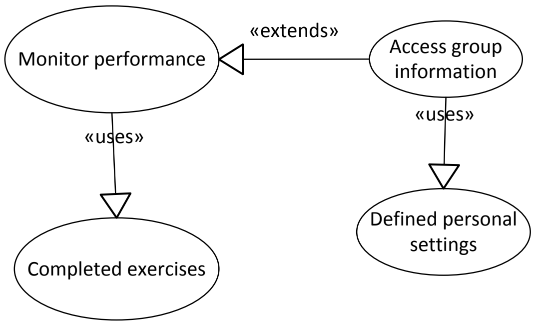

Figure C. 2: Monitor performance

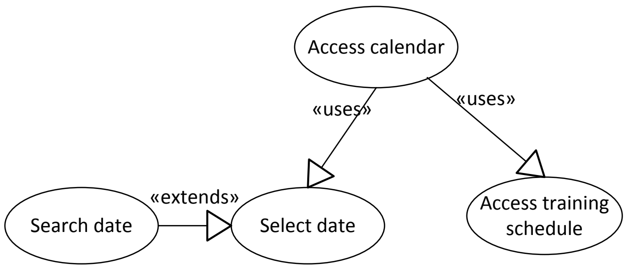

Figure C. 3: Access calendar

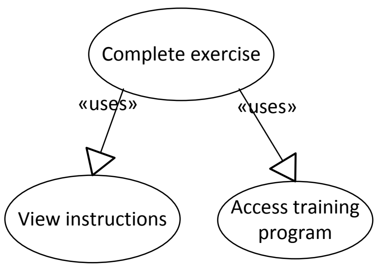

Figure C. 4: Complete exercise

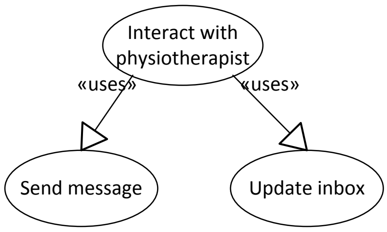

Figure C. 5: Interact with physiotherapist

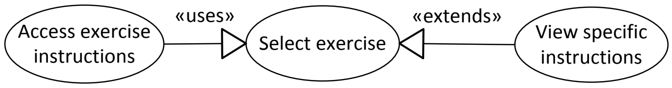

Figure C. 6: Select specific exercise

APPENDIX D: DETAILED DESKTOP USER INTERFACE WIREFRAMES (PHYSIOTHERAPIST)

Detailed wireframes from all desktop applications views will be presented in this chapter.

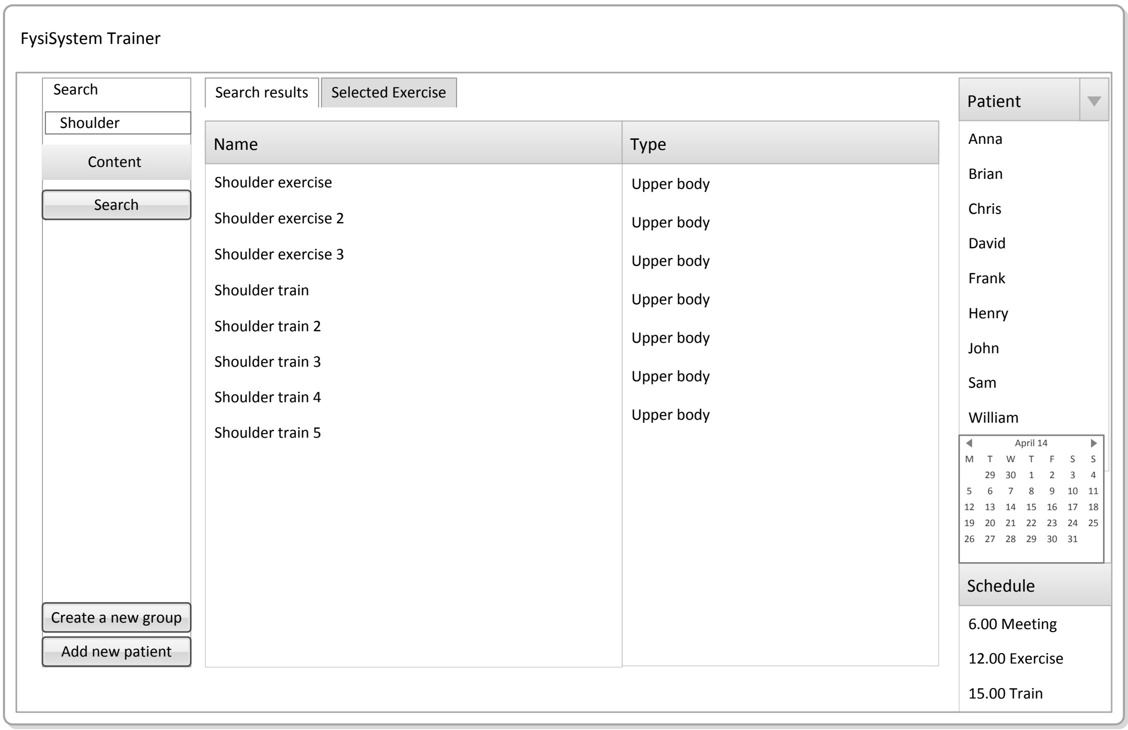

Figure D. 1: Search

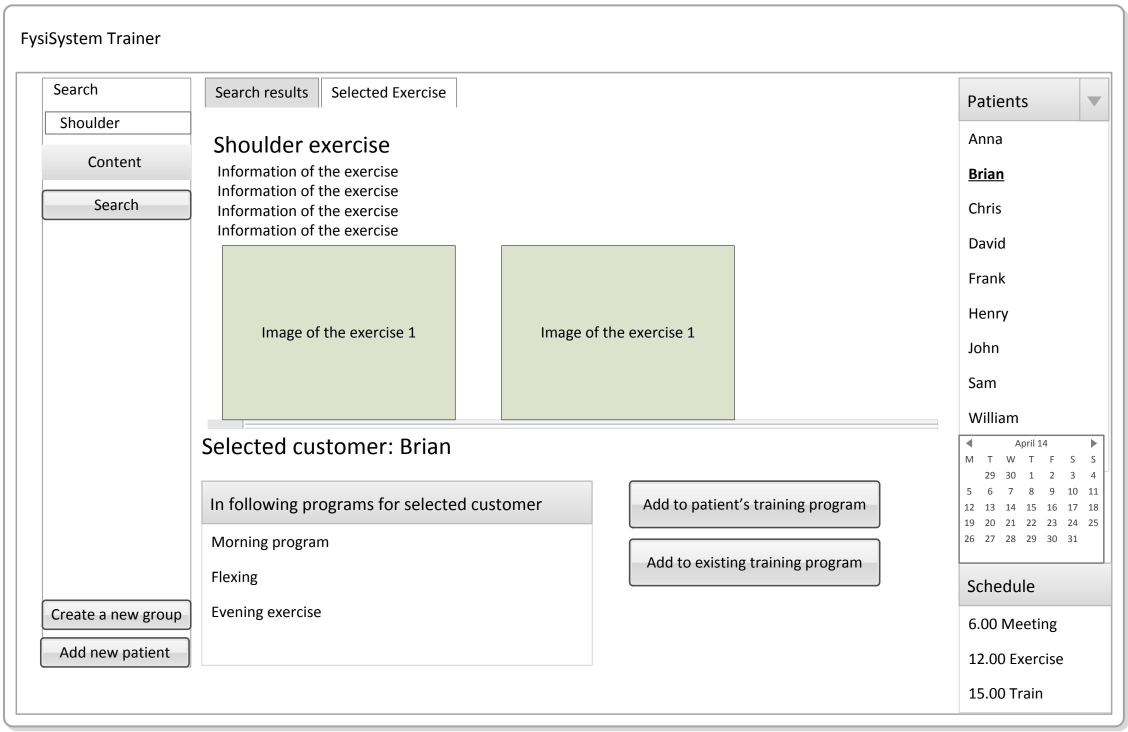

Figure D. 2: Exercise Info

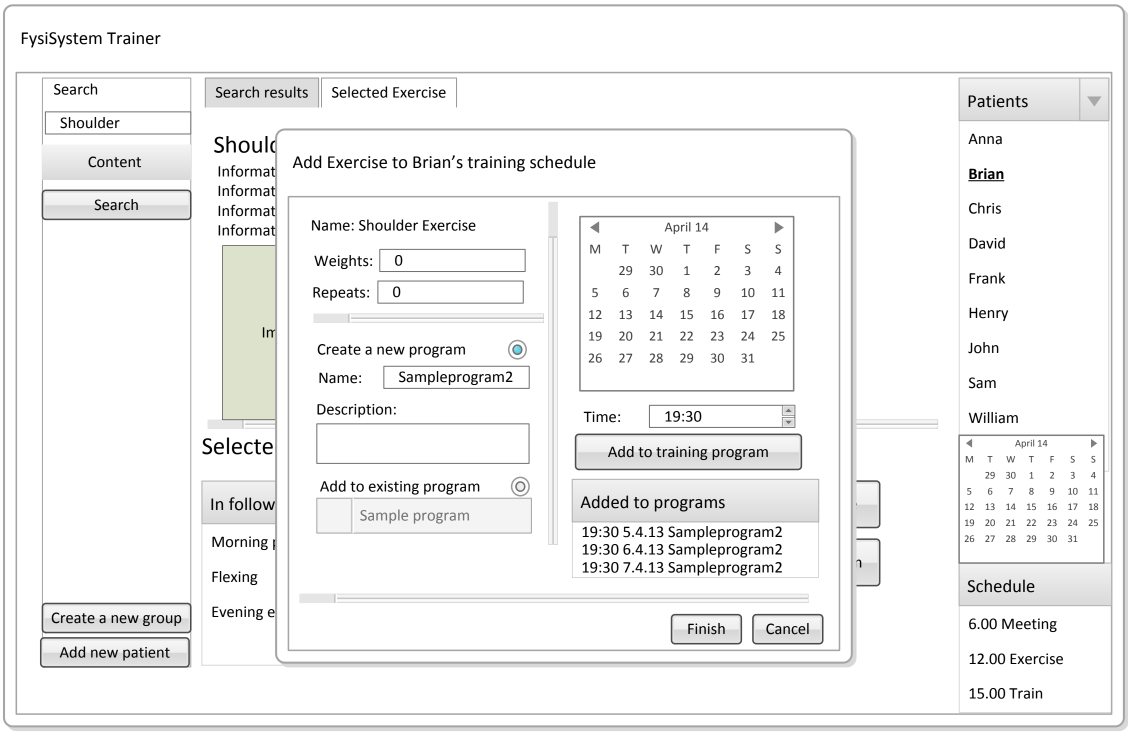

Figure D. 3: Add exercise to training program

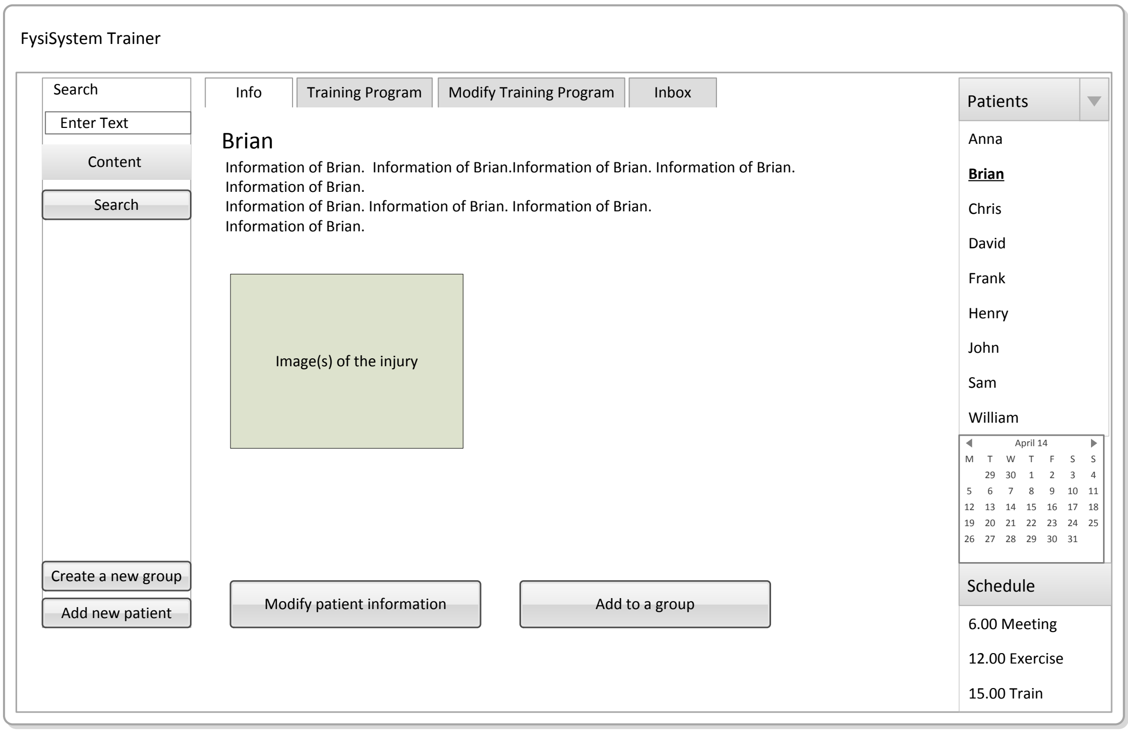

Figure D. 4: Patient's info

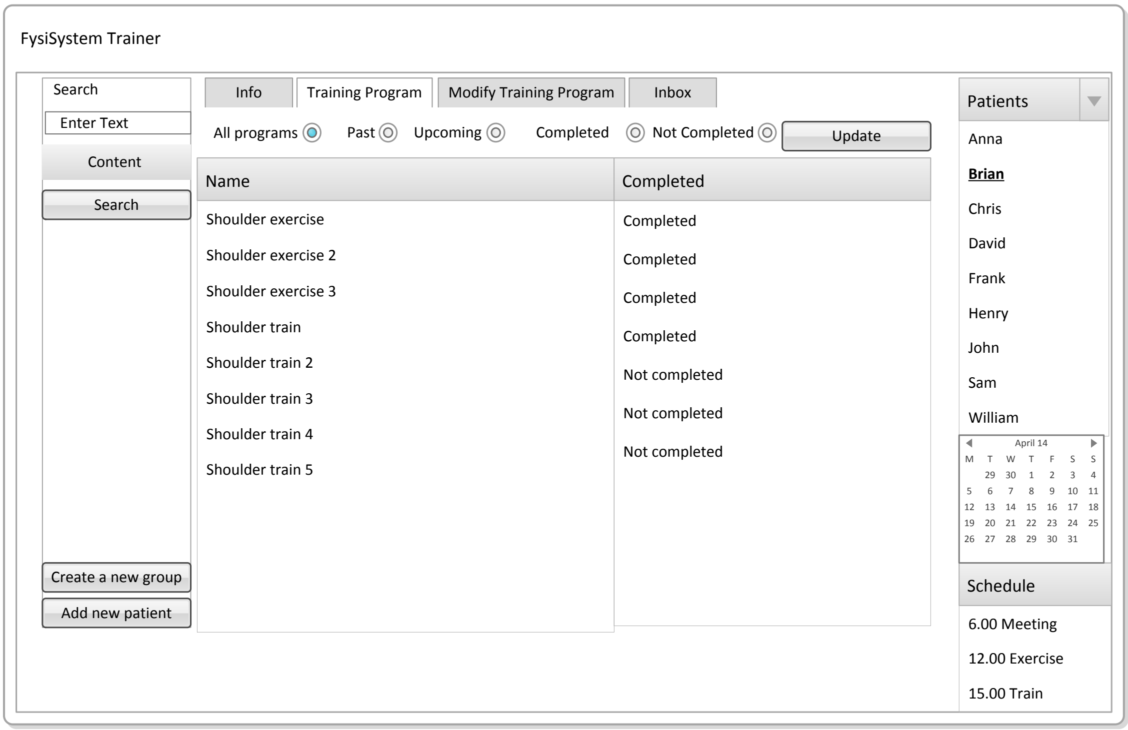

Figure D. 5: Patient's training program

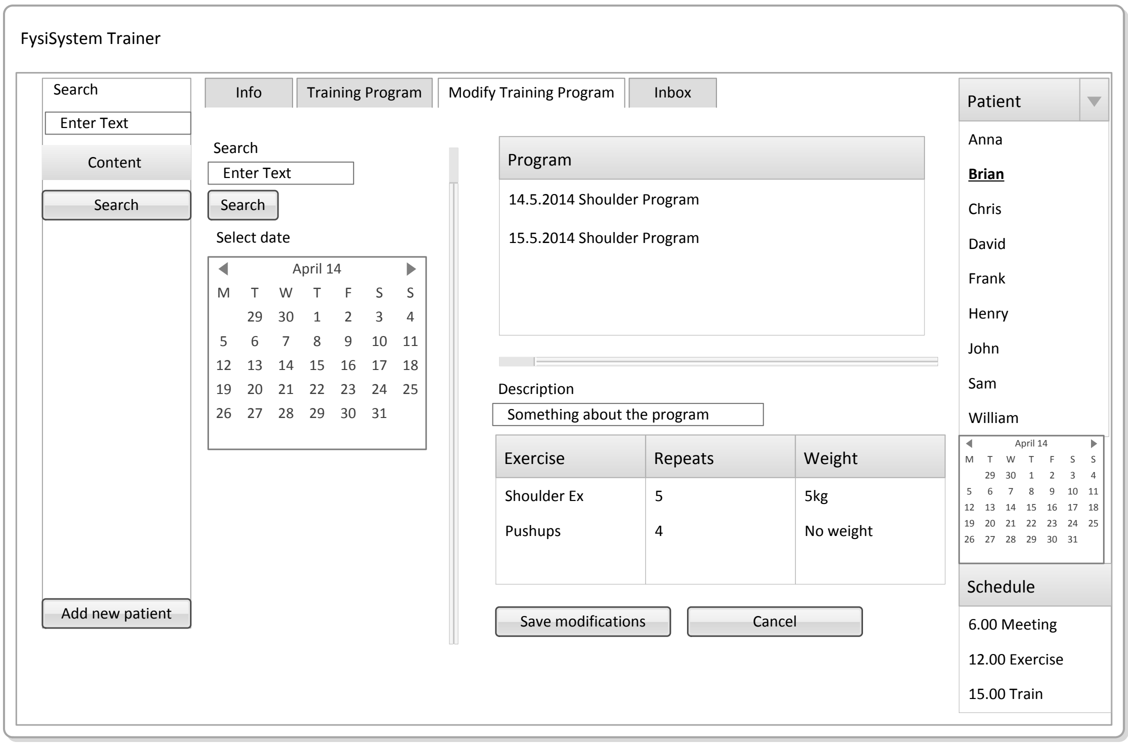

Figure D. 6: Modify patient's training program

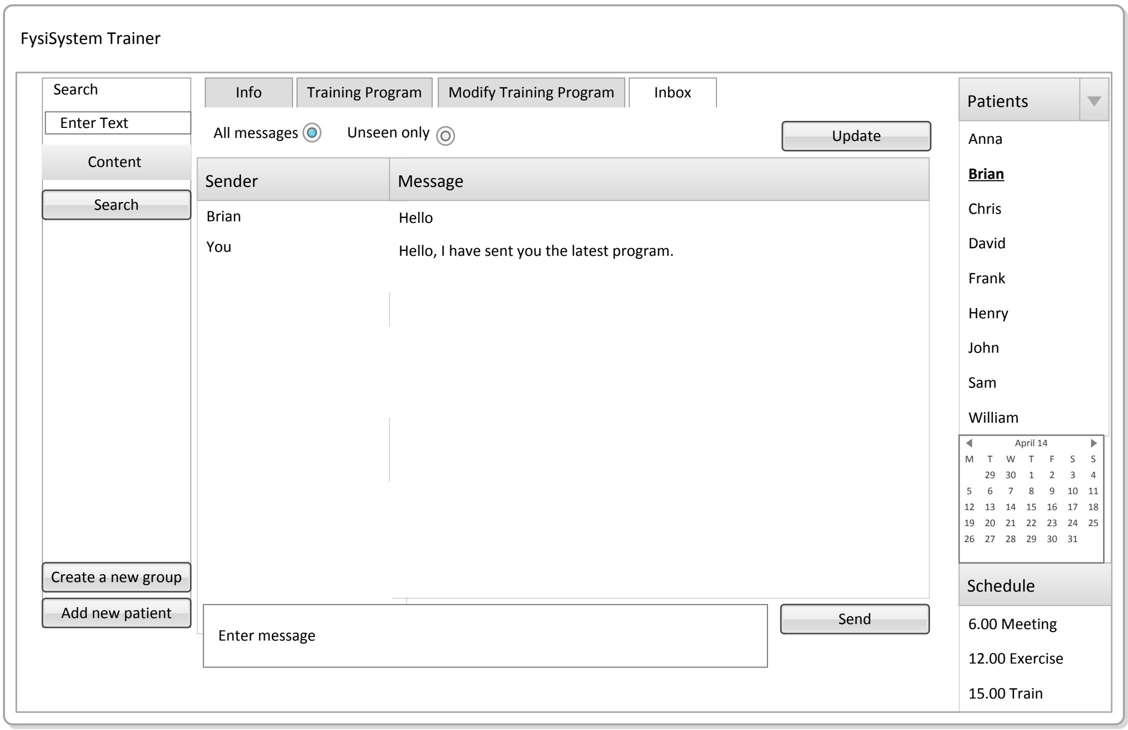

Figure D. 7: Interaction with selected patient

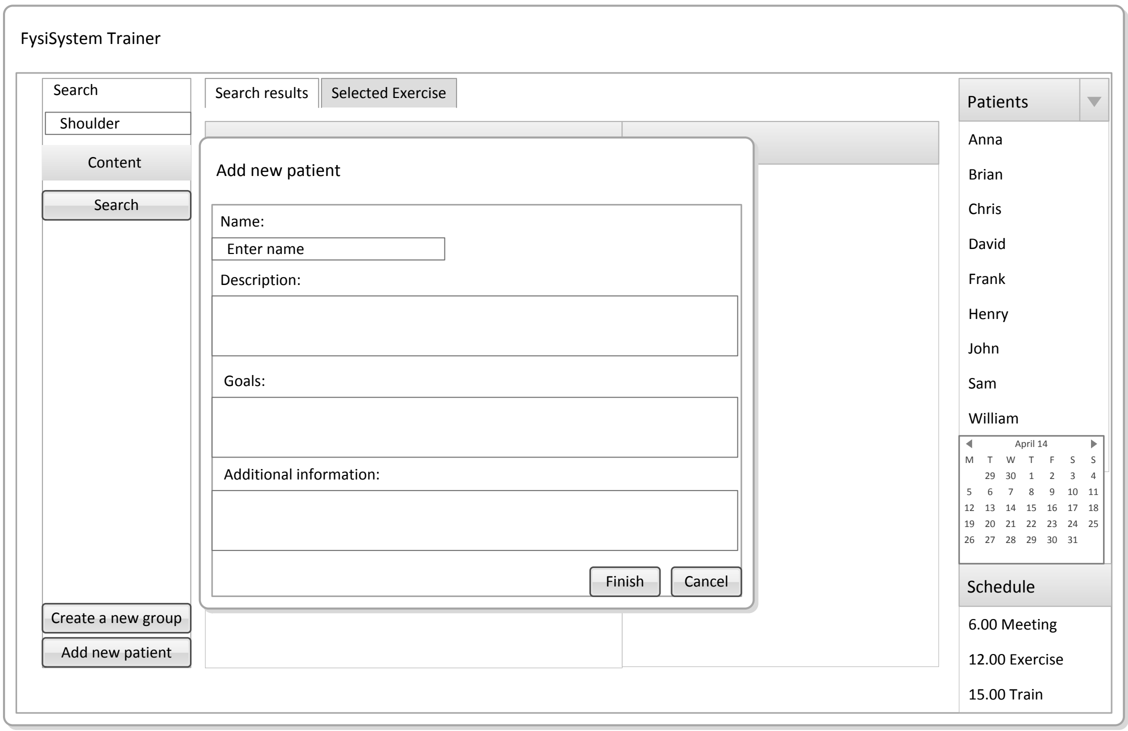

Figure D. 8: Add new patient

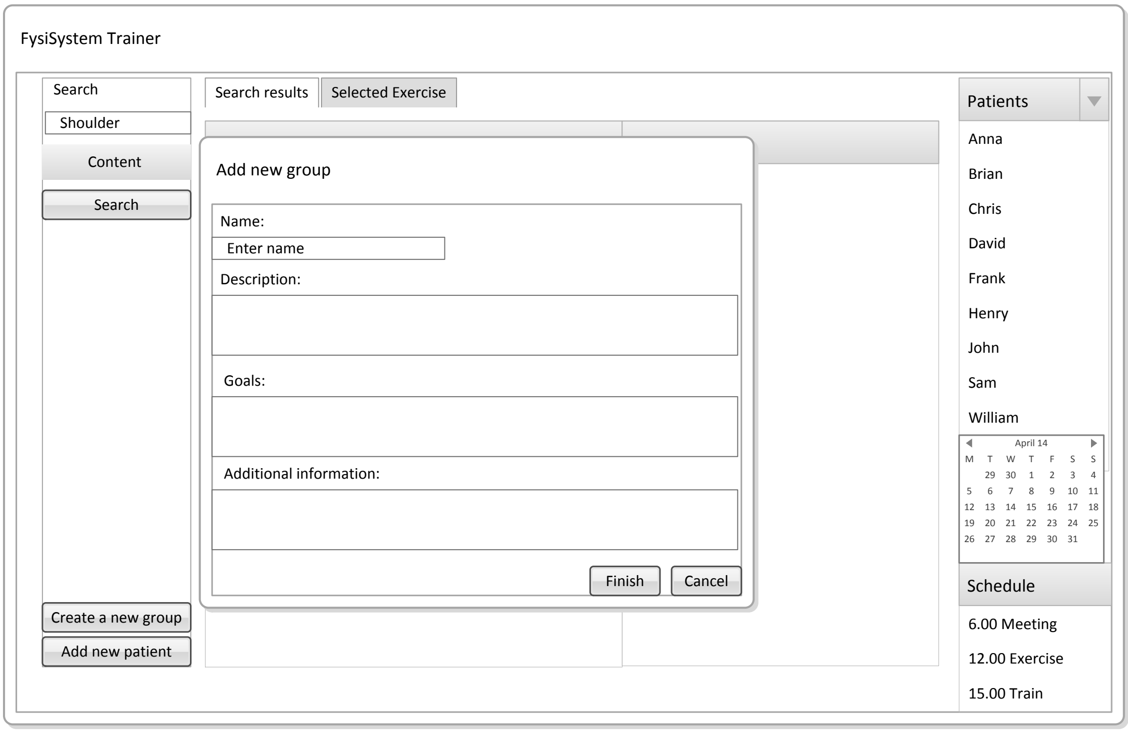

Figure D. 9: Add new group

Figure D. 10: Group interaction

APPENDIX E: DETAILED MOBILE USER INTERFACE WIREFRAMES (PHYSIOTHERAPIST)

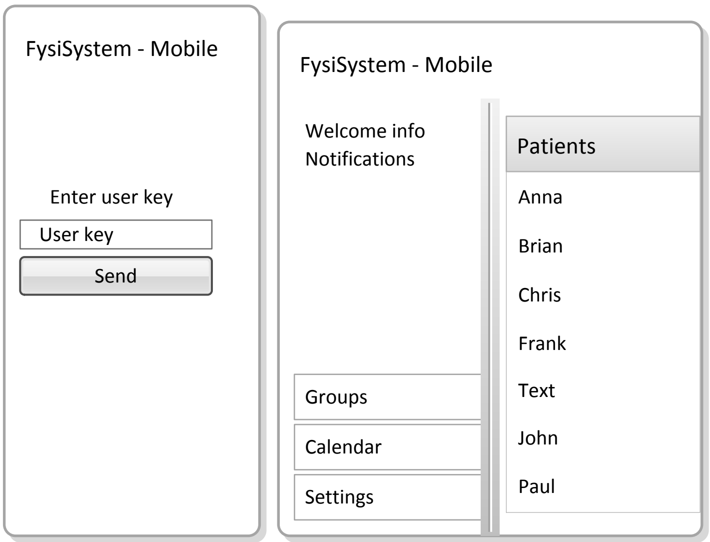

Figure E. 1: Login (Left) and Main View (Right)

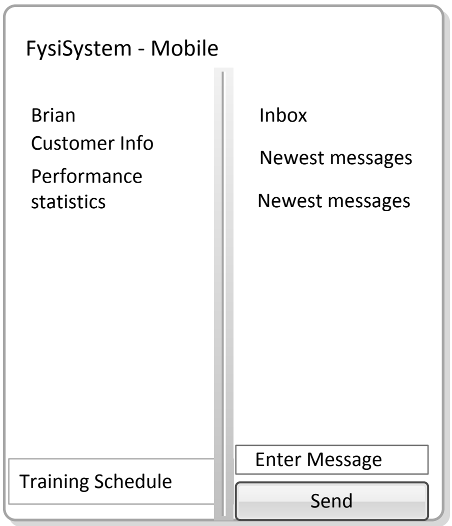

Figure E. 2: Patient info

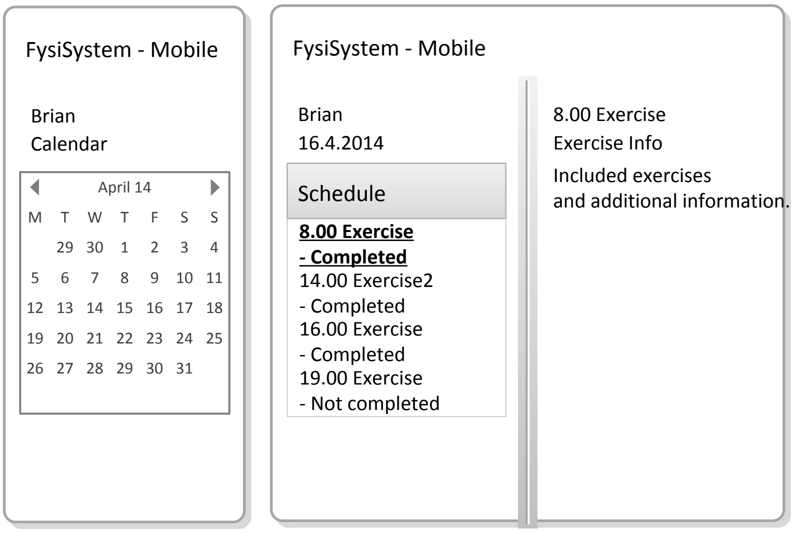

Figure E. 3: Selection of the date on calendar (Left) and patient's schedule on specific date (Right)

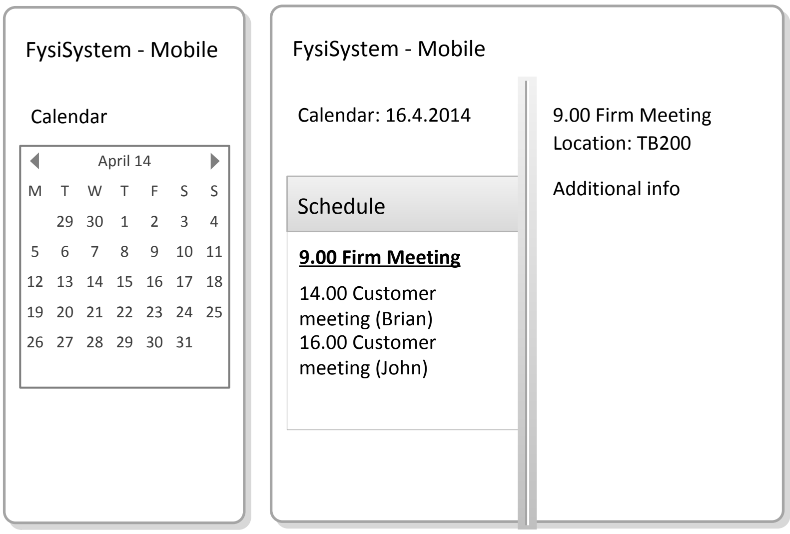

Figure E. 4: Selection of the date on calendar (Left) and personal schedule on specific date (Right)

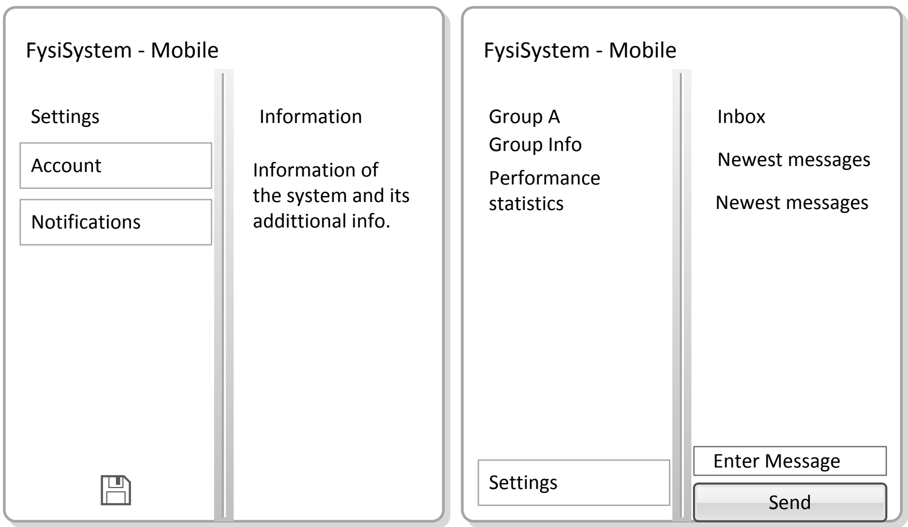

Figure E. 5: Settings (Left) and Group info (Right)

APPENDIX F: DETAILED USER INTERFACE WIREFRAMES (PATIENT)

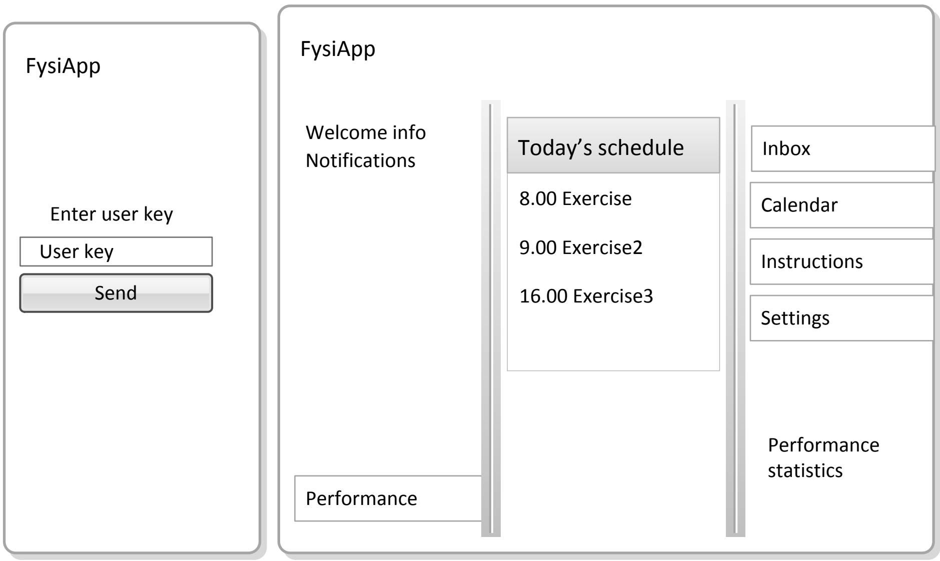

Figure F. 1: Login (Left) and Main View (Right)

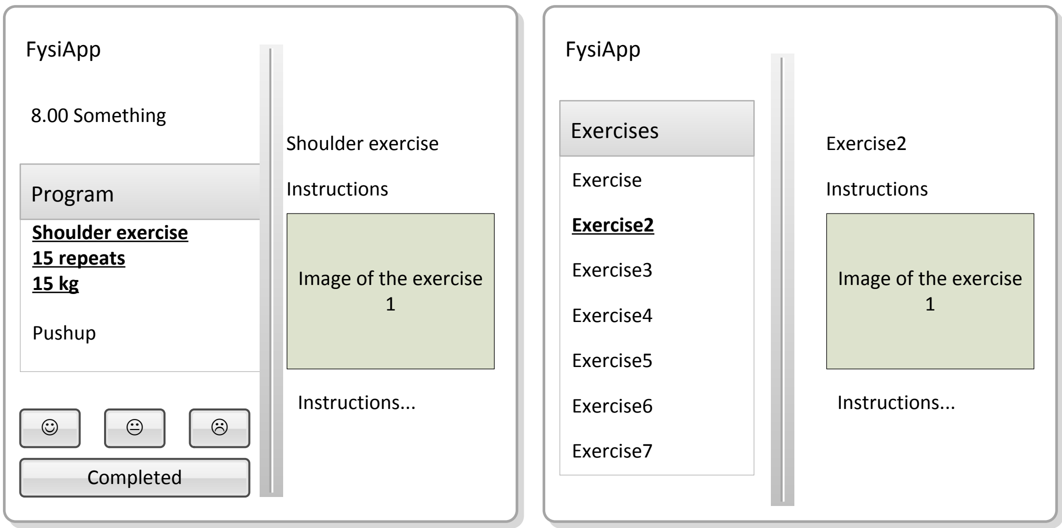

Figure F. 2: Selected exercise info (Left) and Instructions (Right)

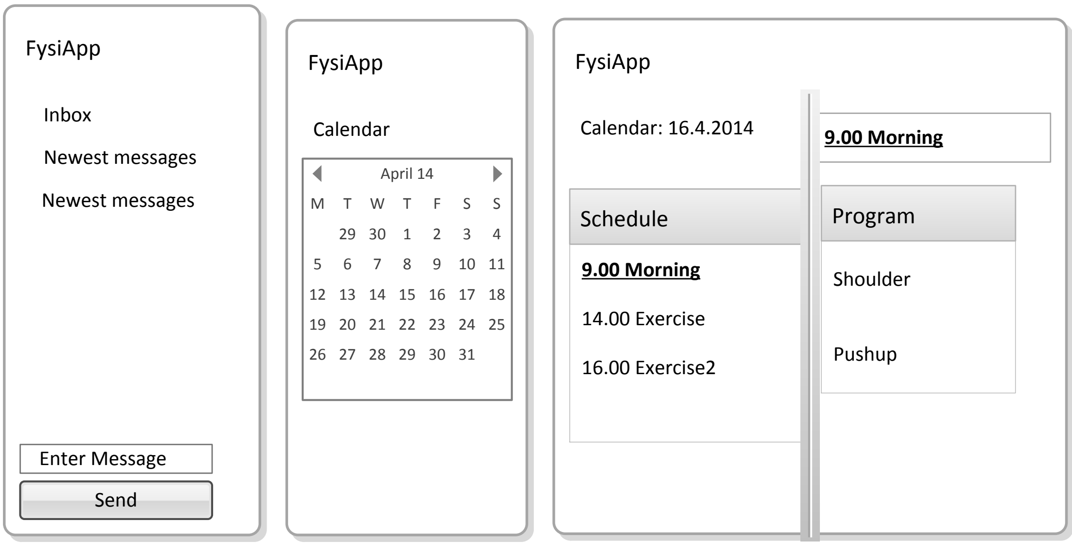

Figure F. 3: Inbox (Left) and Calendar (In the middle and on the right)

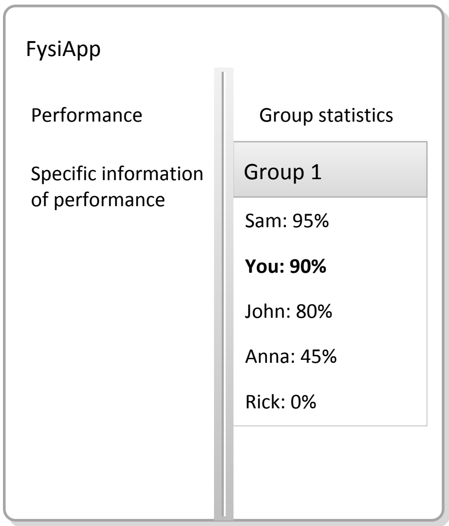

Figure F. 4: Performance

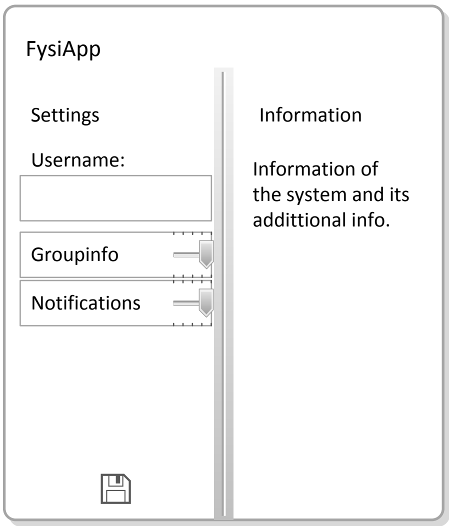

Figure F. 5: Settings (Left) and Instructions

APPENDIX G: DETAILED USER INTERFACE PICTURES (PHYSIOTERAPIST)

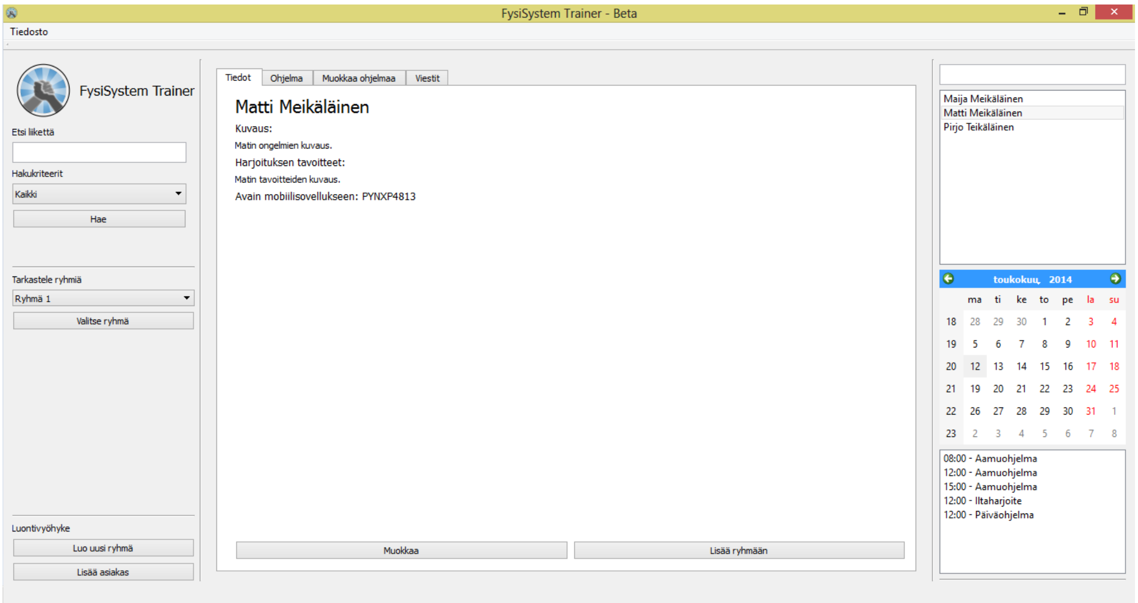

Figure G. 1: Patient info

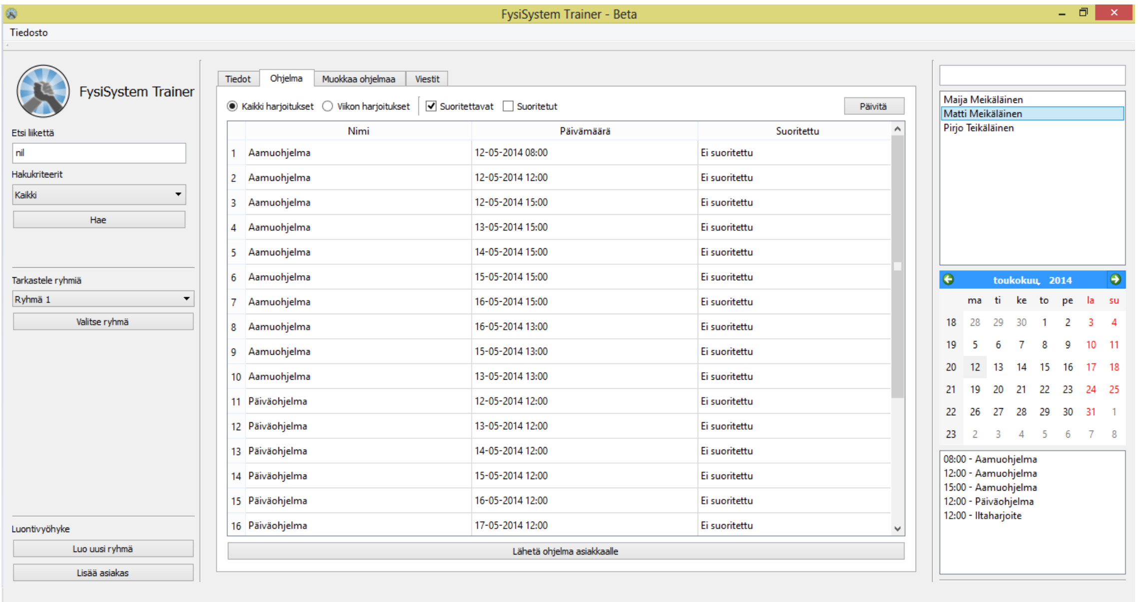

Figure G. 2: Patient's training program

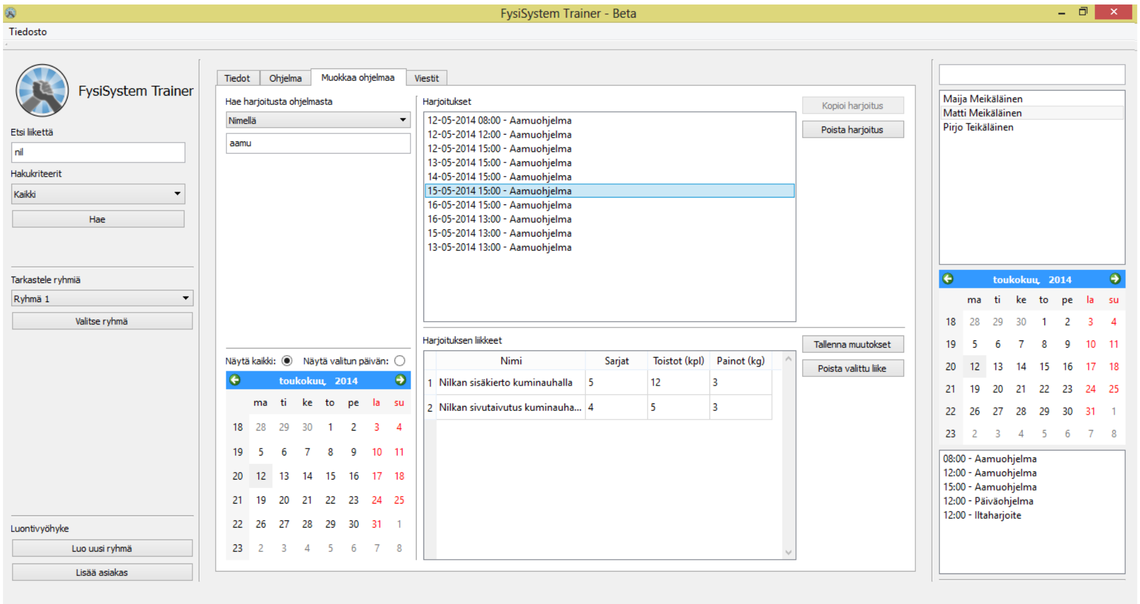

Figure G. 3: Modify patient's training program

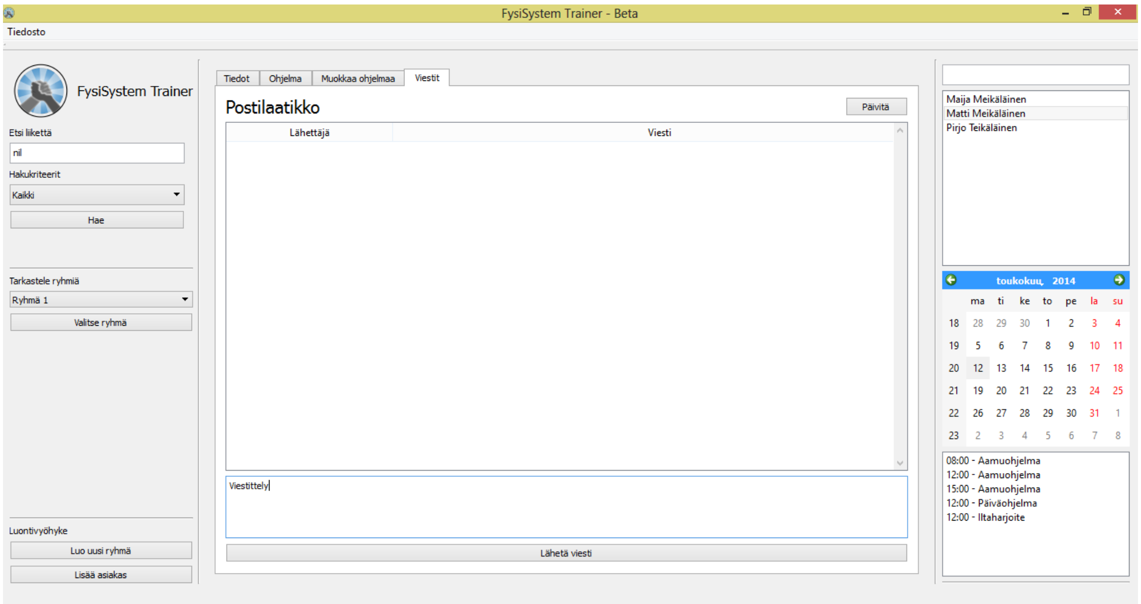

Figure G. 4:Interaction with a patient

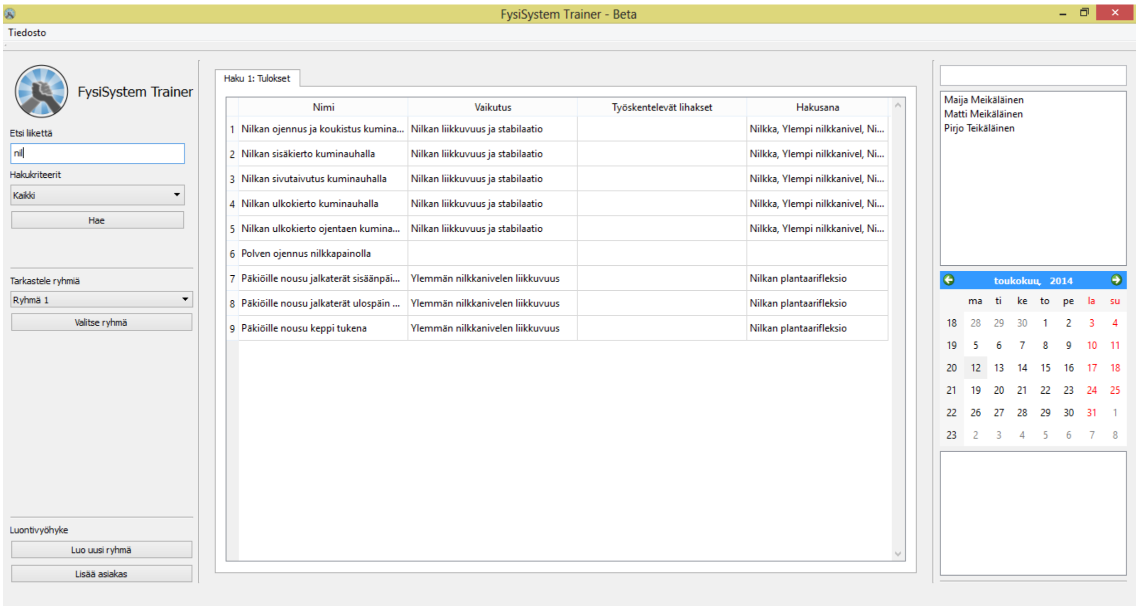

Figure G. 5: Search exercise

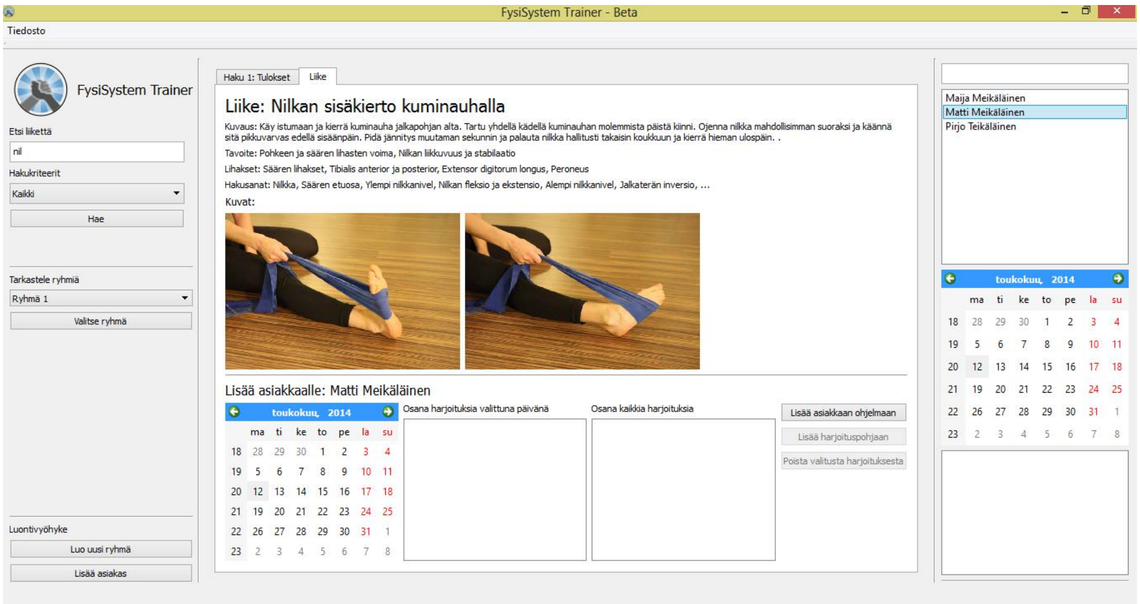

Figure G. 6: Exercise information

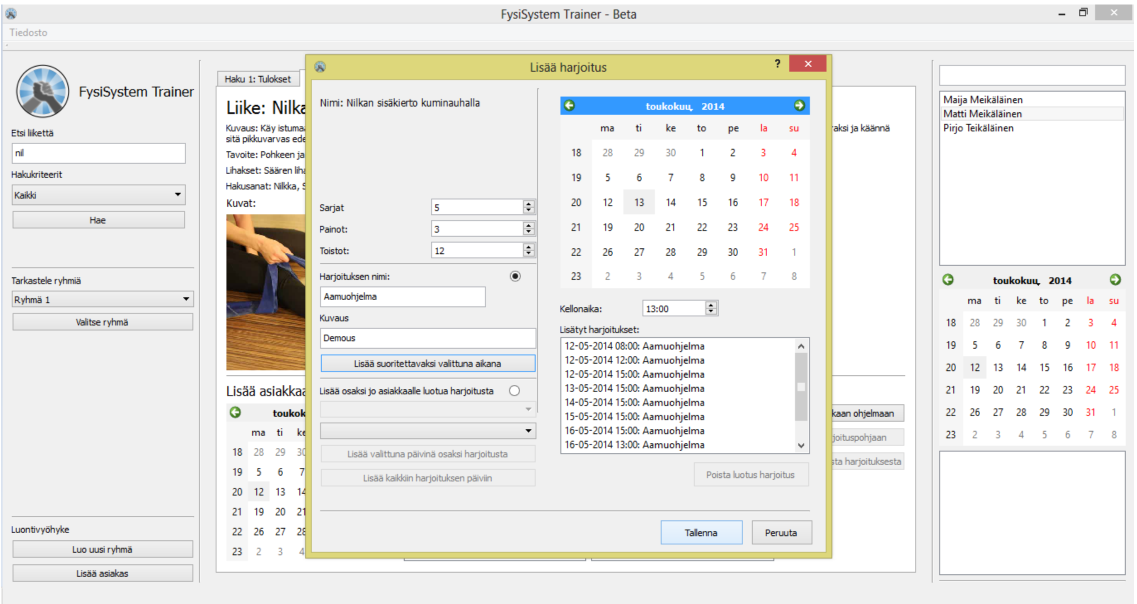

Figure G. 7: Add exercise to patient's training program

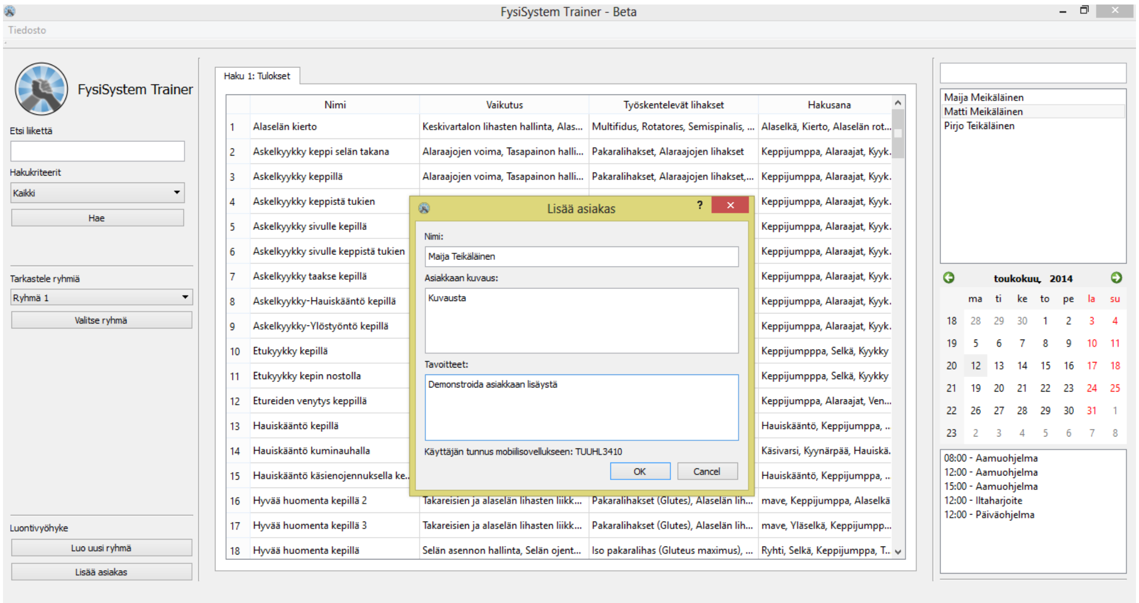

Figure G. 8: Add new patient

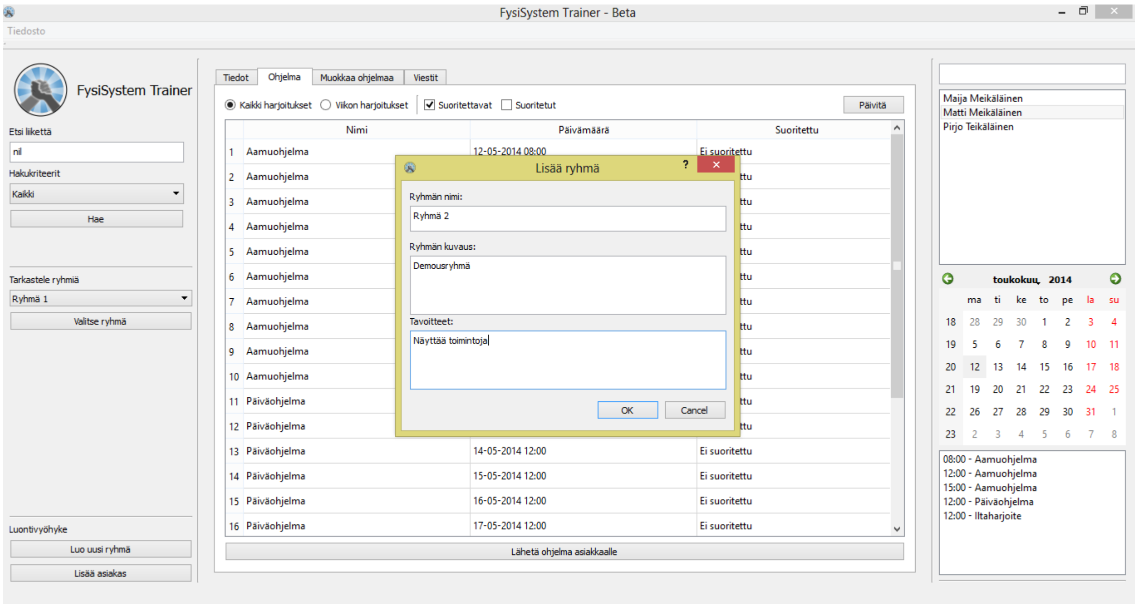

Figure G. 9: Add new group

Detailed images of the Windows Phone application are presented below. They are either panorama, pivot or single page views. These images were taken from screen captures of the implemented prototype with Nokia Lumia 720. The physiotherapist's mobile applications functionality was limited and all of the functionalities were not implemented.

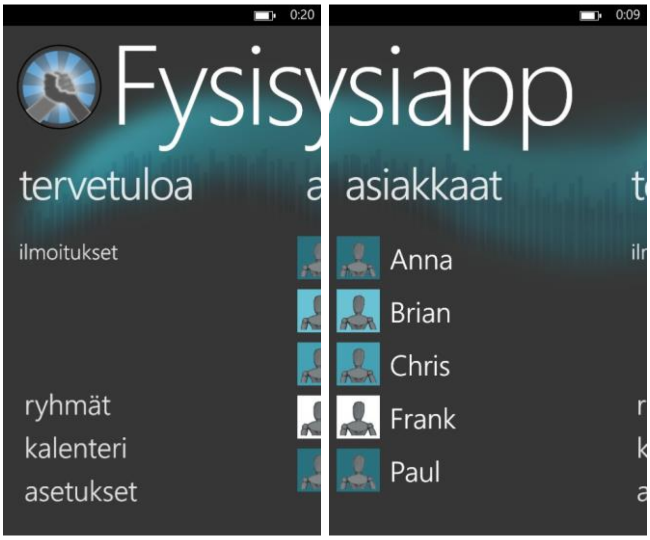

Figure G. 10: Main panorama

Figure G. 11: Group information, panorama

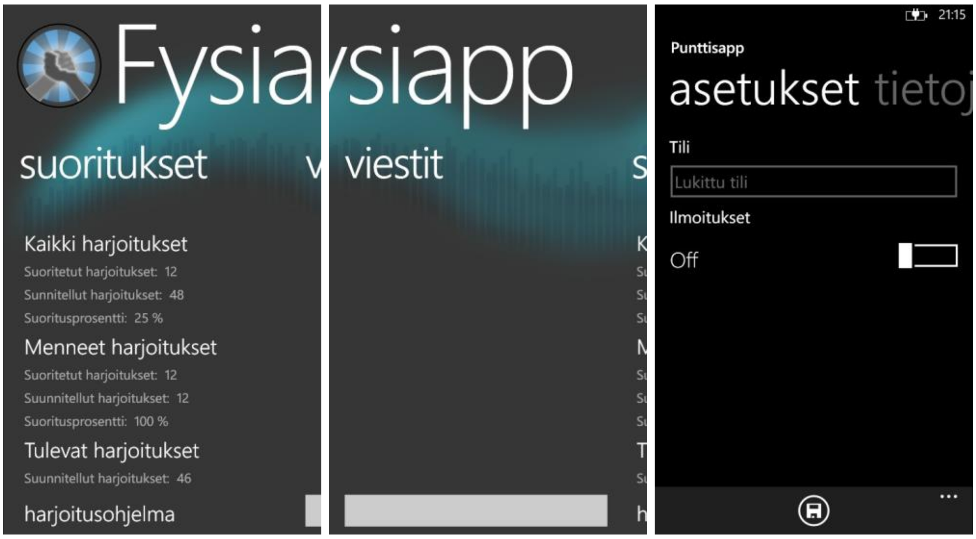

Figure G. 12: Individual patient panorama (Left) and Settings (Right)

APPENDIX H: DETAILED USER INTERFACE PICTURES (PATIENT)

Detailed images of the Windows Phone application are presented below. They are either panorama, pivot or single page views. These images were taken from screen captures of the implemented prototype with Nokia Lumia 720.

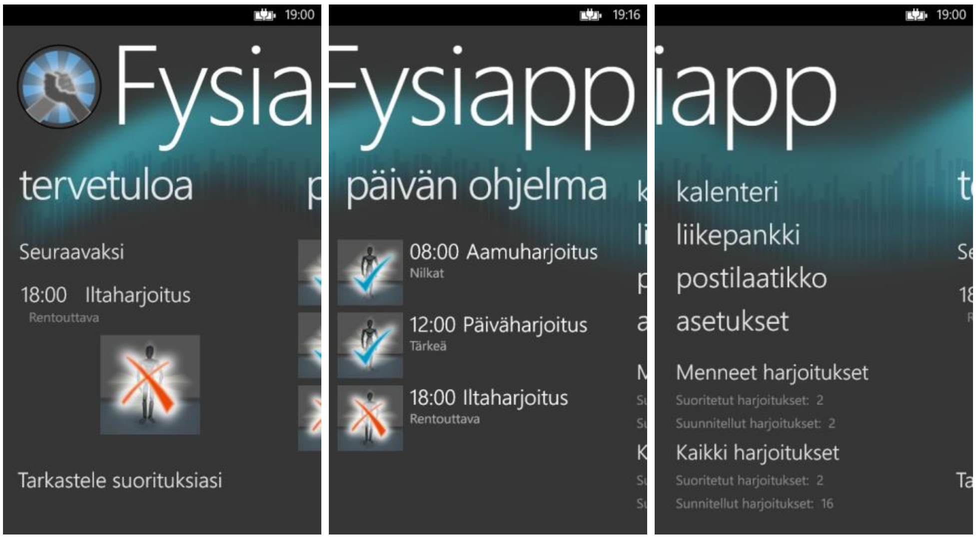

Figure H. 1: Main panorama

Figure H. 2: Completing exercise, panorama

Figure H. 3: Exercise instructions, panorama

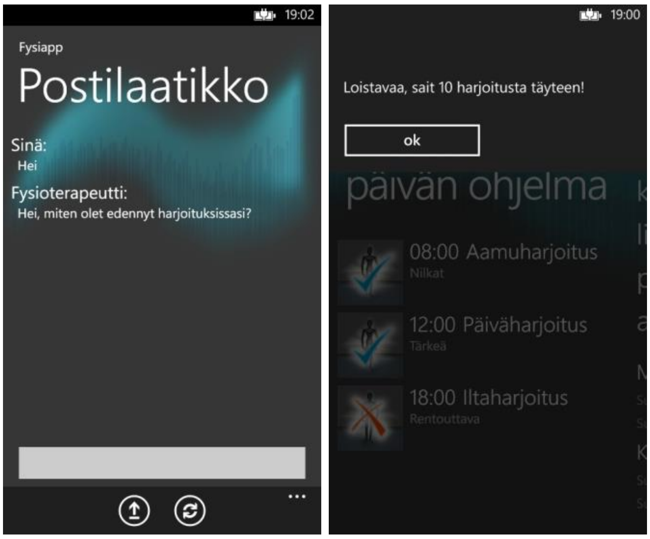

Figure H. 4: Inbox (Left) and Notification (Right), single page view

Figure H. 5: Own performance (Left) and Group performance (In the middle and on the right), panorama

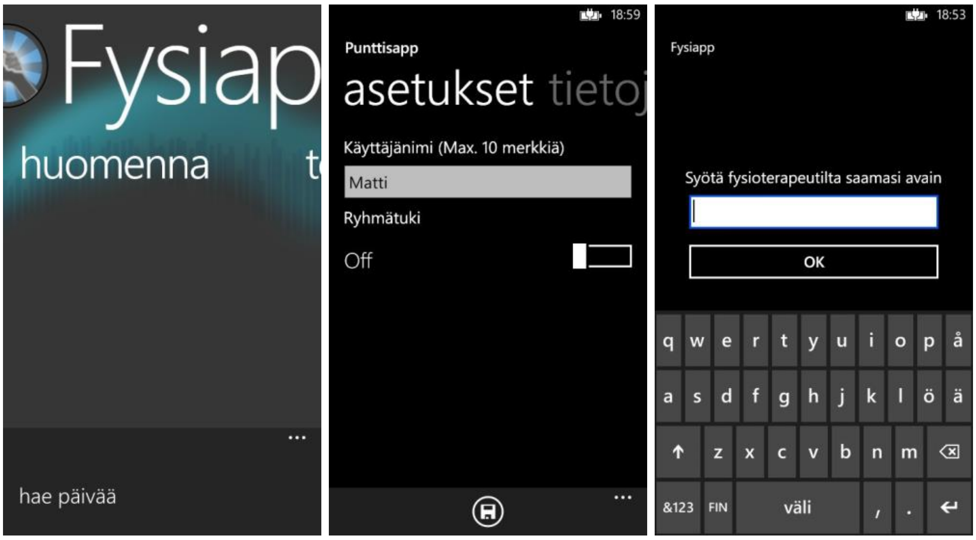

Figure H. 6: Calendar (Left), Settings (Middle) and Login (Right)

Detailed images of the Android application are presented below. These images are from low-fi prototype with exercise information and performance support pages missing. These images were taken from screen captures of Eclipse's emulator for Nexus S with screen resolution of 480 x 800 pixels.

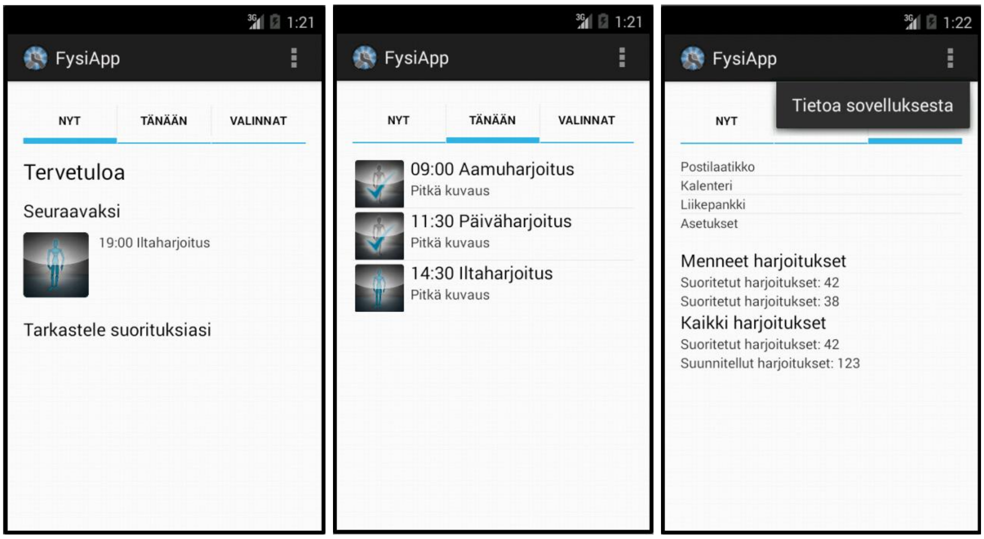

Figure H. 7: Main view

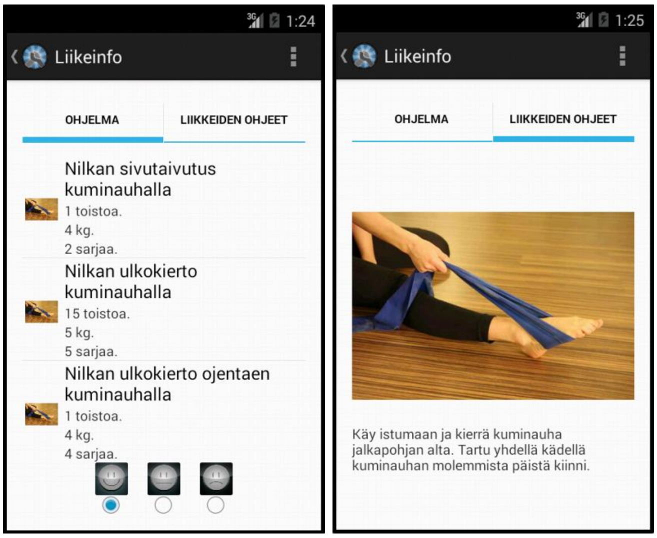

Figure H. 8: Completing exercise

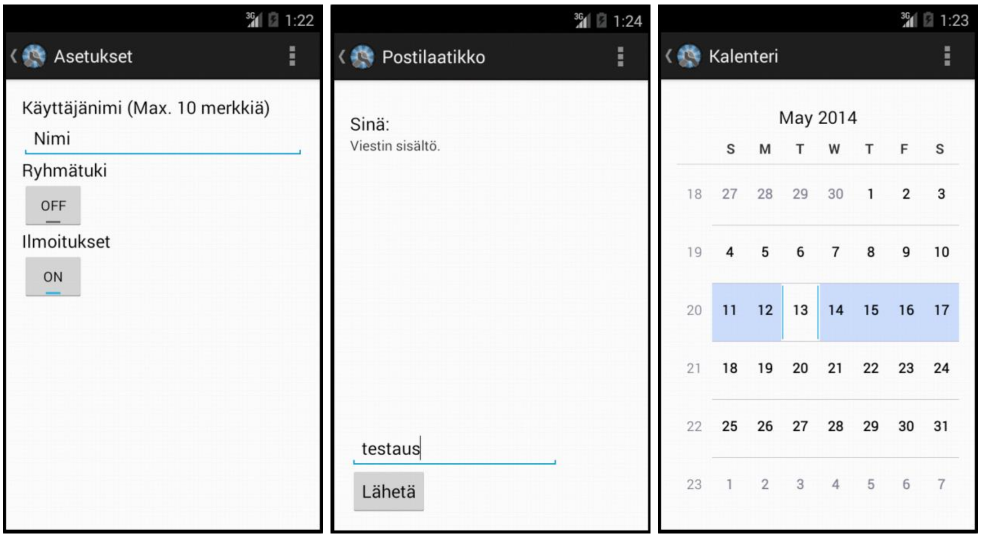

Figure H. 9: Settings (Left), Inbox (Middle) and Calendar (Right)