Robert Gordon University

Bachelor of Science in Computer Graphics and Animation

Honours Project Report

A Mobile Application that applies a Self-Management Approach to Reduce Sedentary Behaviour

Author:
Georgi Koemdzhiev

Supervisor:
Dr. Stewart Massie

2017

Declaration

I confirm that the work contained in this Honours project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Georgi Koemdzhiev
25th May 2017

Abstract

Sedentary behaviour is a leading cause of numerous chronic deceases, but currently, there is a lack of mobile applications that try to help tackle this problem. The purpose of this paper is to show how behaviour change techniques can be applied in a mobile application to help people change their sedentary habits. This project proposes a smartphone application that uses daily activity guidelines to increase physical activity. This software will be developed based on findings from background research. It will incorporate behaviour change techniques that will allow maintaining a good goal-performance relationship. Also, self-management logic will be present to facilitate self-care features such as setting a psychical activity goal.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor Dr Stewart Massie for his continuous guidance during the research and development of the project.

Also, I would like to thank the designers and developers at xDesign for sharing their knowledge and experience with me which helped the project implementation tremendously. Furthermore, I would like to thank my parents Vangel Koemdzhiev and Violeta Koemdzhieva for their encouragement throughout the writing of this thesis and my partner Mariya Menova for her endless support and understanding.

Contents

Declaration **i**

Abstract **ii**

Acknowledgements **iii**

Contents **iv**

List of Figures **vi**

List of Tables **vii**

Glossary **viii**

1 Introduction **1**
1.1 Aim and Objectives 1
1.2 Document structure 2

2 Literature Review **3**
2.1 Self-Management in the Health Sector 3
2.2 Sedentary Behaviour 4
2.3 Behaviour change 5
2.4 HAR enabled mobile applications 6
2.5 Proposed mobile application 9

3 Specification **10**
3.1 Functional Requirements 10
3.2 Non-functional Requirements 12

4 Design **14**
4.1 Mobile Platform and IDE 14
4.2 HAR 14
4.3 Architectural design 17
4.4 Database Design 20
4.5 User Interface Design 21

5 Development Methodologies **24**
5.1 Development Model 24
5.2 Version Control System 24
5.3 Project Management 25
5.4 MVP and Dependency Injection 25

6 Implementation **27**
6.1 HAR System 27
6.2 Self-Management System 30
6.3 "Active Minutes" application 31

7 Testing and Evaluation **36**
7.1 Acceptance Testing . . . 36
7.2 Functional Testing . . . 38
7.3 Usability Testing . . . 39
7.4 Comparing the system with commercial solutions . . . 41
7.5 Survey results . . . 41

8 Project Considerations **44**
8.1 Professional . . . 44
8.2 Social . . . 44
8.3 Ethical . . . 44
8.4 Security . . . 44
8.5 Legal . . . 45

9 Concluding Remarks **46**
9.1 Objectives achieved . . . 46
9.2 Future versions . . . 47
9.3 Reflection . . . 47
9.4 Conclusion . . . 48

Bibliography **49**

A Use cases **54**

B Screen Wireframes **56**

C Screen Designs **60**

D Dagger Component and Module **64**

E ARFF dataset file sample **66**

F Project Log **67**

G Functional Testing **70**
G.1 Navigational . . . 70
G.2 Goal-setting (Initial Setup) . . . 71
G.3 Goal-setting (Settings screen) . . . 71
G.4 Monitoring Activity . . . 72
G.5 Monitoring Inactivity . . . 73
G.6 Notifications . . . 73
G.7 Other . . . 73

H Usability Testing **74**

I Survey results **76**

J Third-party software used in this work **78**

K ActivityMonitor class **81**

L ActiveMinutesService **82**

M Ethics Form **83**

N Consent Forms **87**

O Presentation Slides **91**

P Project Proposal **98**

List of Figures

2.1 Current technologies that self-monitor and provide feedback on SB (Sanders et al., 2016) . . . 4
2.2 General design of HAR systems . . . 7
2.3 General HAR system data flow . . . 7

3.1 Use case diagram of *"Active Minutes"* . . . 11

4.1 Human Activity Recognition process . . . 15
4.2 *"Active Minutes"* Architectural model (simplified) . . . 17
4.3 Model-View-Presenter design pattern (Syromiatnikov and Weyns, 2014: 28) . . . 18
4.4 Database ER Diagram . . . 20
4.5 Iterative process of UI prototyping . . . 22
4.6 Flinto prototype creation process . . . 23

5.1 Incremental software development model life cycle . . . 24
5.2 Taiga Kanban boards . . . 25
5.3 Data collection screen directory structure . . . 26

6.1 HAR system class diagram (simplified) . . . 29
6.2 Self-Management System class diagram (simplified) . . . 30
6.3 Realm database class diagram (simplified) . . . 32
6.4 ActiveMinutesService class diagram . . . 33
6.5 Today screen information . . . 33
6.6 History screen list item snippet . . . 34
6.7 Settings screen snippet . . . 34
6.8 SET PA GOAL screen snippet . . . 34
6.9 Performance Meeting screen snippet . . . 35

7.1 Comparing the system with other solutions . . . 42

B.1 Main Screen. Navigation drawer opened. . . . 56
B.2 History Screens . . . 57
B.3 Authentication Screens . . . 57
B.4 Utility Screens . . . 58
B.5 Performance Meeting Screen . . . 58
B.6 Initial setup screens . . . 59

C.1 Today Screen. Navigation drawer opened . . . 60
C.2 History Screens . . . 61
C.3 Authentication Screens . . . 61
C.4 Utility Screens . . . 62
C.5 Performance Meeting Screen . . . 62
C.6 Initial setup screens . . . 63

List of Tables

2.1 Examples of light, moderate, and vigorous activities (Harvard TH Chan School of Public Health, 2012) . . . 5

6.1 4 classes Confusion Matrix . . . 28
6.2 2 classes Confusion Matrix . . . 28

7.2 GSC acceptance tests . . . 37
7.4 MC acceptance tests . . . 37
7.6 FC acceptance tests . . . 38
7.8 Monitoring activity functional test snippet . . . 39
7.10 Usability test snippet . . . 39
7.12 User survey extract . . . 42

F.1 Project Log . . . 69

Glossary

CFS

Correlation based Feature Selection

DI

Dependency Injection

FC

Feedback Component

FFT

Fast Fourier Transform is an algorithm that converts a signal from its original domain (often time) to a representation in the frequency domain

GSC

Goal-Setting Component

HAR

Human Activity Recognition

IDE

Integrated Development Environment

kNN

k-Nearest Neighbors

MC

Monitoring Component

MCI

Maximum Continuous Inactivity

MVC

Model View Controller is a design pattern that separates the software logic into tree roles: model, view, or controller

MVP

Model-View-Presenter is software architectural pattern that is a pattern which provides a cleaner separation between the View, the Model and the Presenter

OMS

Object-Oriented Data Management System

PA

Physical Activity

SA

System Architecture

SB

Sedentary Behaviour

SF

Sampling Frequency

SHTI

Sleeping Hours Time Interval. The start and end time when a person normally goes to sleep and wakes up, accordingly

ST

Sedentary Time

UI

User Interface

VCS

Version Control System

Chapter 1

Introduction

Nowadays, with the rapid advance of technology people spend a lot of their time in a static position. According to Wilmot et al. (2012) that increases the chances of all-cause mortality by 49%. Also, people who lead non-active lifestyle are 112% more likely to get diabetes, 147% more likely to experience cardiovascular events, and 90% more likely to die due to cardiovascular events. According to CERB (2015), sedentary behaviour costs the EU economy 80 billion euros annually and is found to be the leading cause of absence from work (Wynne-Jones et al., 2013).

Using self-management systems such as *Selfback* (Sani et al., 2016) to counter sedentary behaviour is becoming a well-accepted solution due to its many advantages over the traditional paper-based methods. This paper proposes a mobile application that aims to help inactive people reduce sedentary time by implementing self-management logic. It allows the user to set physical activity and sedentary time goals and provides them with goal-performance feedback in the form of notifications i.e. when a goal is achieved and when extended inactivity is detected. The system monitors physical activity and sedentary time in real-time by incorporating a Human Activity Recognition (HAR) system capable of recognising four activities i.e. walking, running, cycling and static.

1.1 Aim and Objectives

Aim

The aim of this project is to develop a mobile application to support people with inactive lifestyles as they attempt to change their behaviour and become more active. The behaviour change will be achieved by building a self-management system that allows the user to set their goals. The built-in sensors of the mobile device will then be used to monitor the user's progress towards the set goals with achievement-aware notifications being pushed to the user to provide motivation.

Objectives

The following list of objectives helped to accomplish the above goals:

1. Research the effect of Sedentary Behaviour (SB) as well as investigate behaviour change approaches
2. Investigate and gather information about Human Activity Recognition (HAR) on wearable devices
3. Develop a fully working HAR system
    - Gather training data
    - Use the above to train a classifier (needed to recognise user activities such as walking)
4. Incorporate the above within a SB self-management system
    - Allows the user to set goals

- Perform user activity monitoring
- Provides feedback to the user

5. Design and Implement a mobile application based on the project research findings
   - Produce a system architectural and UI design
   - Incorporate the research findings in the system as business logic
6. Evaluate the system performance using user feedback and analysing the gathered data
   - Test the system using varies techniques such as Acceptance, Functional and Usability tests
   - Gather user feedback regarding the overall functionality of the system via survey
7. Release app to application to the Application Store
   - Publish the mobile application to the general public

1.2 Document structure

This section of the document addresses how this report is organised.

- **2. Literature Review** Gives a wide background information regarding using self-management technologies in the health sector, Behaviour changing techniques, consequences and current situation of of Sedentary Behaviour as well as researching commercial and non-commercial HAR mobile applications.
- **3. Specification** Lists the proposed mobile application Functional and Non-functional requirements as well as discussing the system quality attributes. The information in this chapter is heavily influenced by the *Literature Review* chapter.
- **4. Design** Addresses the system architectural and UI design decisions.
- **6. Implementation** Details about how the system design have been implemented for each of the main system components.
- **7. Testing and Evaluation** Analysing the system in terms of Acceptance, Usability and Functional testing. Also, evaluating the effectiveness of the system on users.
- **8. Project Considerations** A discussion on the professional, social, ethical, security and legal issues relevant to the project.
- **9. Conclusion** Summary of the potential benefits of Sedentary Behaviour self-manageing devices in the health sector, future work and project reflection.

Chapter 2

Literature Review

As it was mentioned in Chapter 1 this document proposes a system implemented on a mobile device that encourages the reduction of sedentary time via self-management techniques. In order to gain knowledge on how to devise such as system, literature review needs to be carried away on several topics. First of all, the current shift towards self-management in the health sector will be discussed. The next section will focus on evaluating how much time people spend in Sedentary Behaviour (SB) and what are the consequences. Section 3 looks at digital behaviour change techniques. Commercial and non-commercial HAR mobile applications and HAR itself are discussed in Section 4. Section 5 summarises the system proposed.

2.1 Self-Management in the Health Sector

Mobile devices with embedded self-management logic are currently being utilised and becoming a well-accepted solution for millions of people in the health sector. For example, the NHS's England Executive – Simon Stevens, launched a programme "Test Beds" (NHS England, 2017a,b) which is a set of collaborative projects between NHS and some technology companies such as Verily, IBM and Philips. The idea behind the project is to test the effectiveness of different technological innovations, including wearable devices and mobile applications. These technologies will enable patients to self-manage illnesses such as diabetes, heart diseases and dementia.

2.1.1 Advantages of self-management

Shifting towards self-care via mobile devices has many advantages over the traditional methods (e.g., over the phone or face-to-face). For example, self-care devices enable patients to monitor their illnesses at home. Roberts (2006: 10) finds that patients using wearables improve their condition and shorten their recovery time if they can be in a familiar environment. Real-time feedback is another advantage of using self-management devices. For example, AliveCor (2017), can prevent critical and even fatal incidents by identifying illness clues ahead of time. Thus, the number of patients becoming seriously ill or needing hospital treatment will potentially be reduced. Consequently, that could save money on hospital treatments (Denis Campbell, 2016). In addition, Shuger et al. (2011: 6) found that real-time feedback on SB could be beneficial for weight loss. Whitehead and Seaton (2016: 97) concludes that self-management approaches delivered via mobile applications have the potential to improve outcomes of many chronic diseases.

2.1.2 Need for self-management devices

It has been found that there is a gap in the current market regarding self-care devices that focus on monitoring SB and not only PA. For example, Sanders et al. (2016) found out that there is a lack of self-management mobile applications or devices on the market that can monitor PA. After research, he was able to find 73 devices that provide self-monitoring of PA, and only 9 can monitor sedentary time (see Figure 2.1). That means that there is a lack of products in the market that offer self-management of SB in addition to PA. Sanders also concluded that the current devices capable of measuring sedentary time and providing feedback have not been used in

behaviour change interventions. In conclusion, there is a need for further research and development of self-care enabled wearables to help people engage in behaviour change interventions and prevent possible chronic diseases.

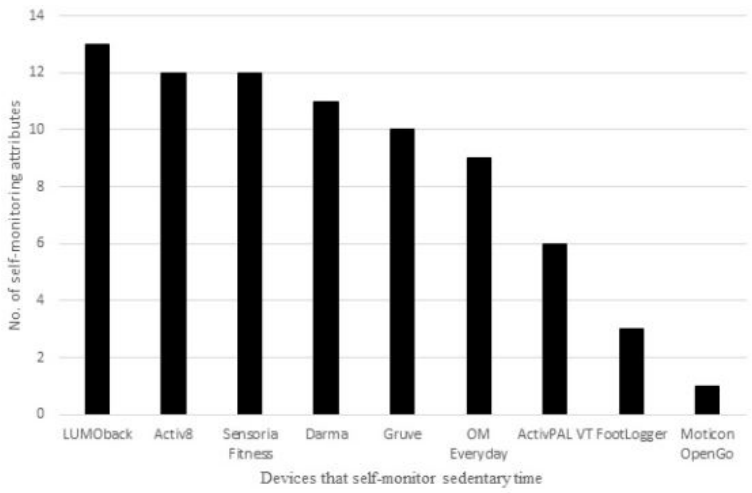


Figure 2.1: Current technologies that self-monitor and provide feedback on SB (Sanders et al., 2016)

2.2 Sedentary Behaviour

Understanding the consequences of SB as well as recommended daily energy expenditure levels is crucial for developing a mobile application that aims to implement behaviour changing theory. It is said that there is a difference between SB and PA and these terms should be treated differently (Networ, 2012: 540). For example, a person can do both be sedentary for prolonged periods of time and spend large amounts of vigorous physical activity in one day. Networ define SB as "energy expenditure less or equal to 1.5 METs while in a sitting or reclining posture" (see Table 1 for comparison of different activities). As it can be seen from Table 2.1, the energy spent in static activities does not result in a lot of energy expenditure and that potentially could lead to the occurrence of chronic diseases. Networ (2012: 540) also defines a person who is "inactive" as "not meeting specified physical activity guidelines". That means that a person has to meet the PA recommendation levels in order to be considered as active. The next section will discuss the consequences of SB and the suggested amount of physical activity per day.

2.2.1 SB consequences and PA recommendations

Sedentary Behaviour has been positively associated with diseases such as back pain, diabetes, cardiovascular disease, cancer and all-cause mortality (Wilmot et al., 2012: 2895–2905; Biswas et al., 2015: 123; Department of Health, 2010: 18). In addition, Biswas et al. (2015: 127) even concluded that SB causes adverse outcomes regardless of the physical activity, although the chances of developing a SB associated chronic diseases are low when involved in higher levels of physical activity.

As far as the PA recommendations are concerned, the Department of Health (2011) and Townsend et al. (2015) advocate for 30 minutes of PA for every day of the week. What is more, Parkinson (2016) and Siddique (2016) state that an ideal PA per day would be an hour. Having said that, simply doing 30 minutes or even an hour of activity is not enough. For example, a person may do 30 minutes of activity a day and spend the rest of the time laying or sleeping. To counter that, Swartz et al. (2011) suggest that every static hour should be interrupted by five-minute walk.

2.2.2 Current statistics of sedentary behaviour

Townsend et al. (2015: 19) found in 2012 that 34% of adult man and 46% of adult women did

Light <3.0 METs*	Moderate 3.0–6.0 METs*	Vigorous >6.0 METs*
Walking—slowly = 2.0	Walking—very brisk (4mph) = 5.0;	Walking/Hiking (4.5mph)= 7.0 Jogging at 6 mph = 10.0
Sitting—using computer = 1.5	Cleaning—heavy = 3.0–3.5 (washing windows, vacuuming, mopping)	Shoveling = 7.0–8.5
Standing—light work = 2.0-2.5 (cooking, washing dishes)	Mowing lawn = 5.5 (walk power mower)	Carrying heavy loads = 7.5
Fishing—sitting = 2.0 Playing most instruments = 2.0–2.5	Bicycling—light effort (10–12 mph) = 6.0 Badminton—recreational = 4.5 Tennis—doubles = 5.0	Bicycling fast (14–16 mph) = 10.0 Basketball game = 8.0 Soccer casual = 7.0 Tennis—singles = 8.0

Table 2.1: Examples of light, moderate, and vigorous activities (Harvard TH Chan School of Public Health, 2012)

not meet the recommended levels of physical activity. Also, according to Townsend et al. (2015: 81), 45% of the adults in the UK spend up to 5h and 30 minutes in SB such as watching TV, using a computer or playing video games. What is more, Swinford (2014) found in a survey that there has been an increase in the time people spend in sedentary activities compared to 1995. For example, he found that people nowadays take 18 per cent fewer journeys, 24 per cent fewer go to do shopping and 28 per cent visit friends compared to 1995. In conclusion, to improve the state of the people who are spending too much time in sedentary behaviour, there is a need for further research on the effectiveness of behaviour change interventions only focusing on sedentary time and physical activity alone (Biswas et al., 2015: 130).

2.3 Behaviour change

Behaviour change is a critical aspect of self-management. It could encourage and motivate people who spend too much time being sedentary to improve their lifestyle. Behaviour change includes the following main components: goal-setting, monitoring and feedback. Strecher et al. (1995: 191) find that setting a specific goal, combined with performance feedback, increases task performance (how well a task is achieved) when compared to having no goal or non-quantitative goal such as "do you best".

Providing feedback is also important in order to successfully implement behaviour change. Locke and Latham (2002: 708) gives the following example "If the goal is to cut down 30 trees in a day, people have no way to tell if they are on target unless they know how many trees have been cut". A person needs to know how much effort they need to put in order to attain a previously set goal. This information is provided by the monitoring component of the behaviour changing circle, which know what PA and SB amounts the user has accumulated at a given time.

2.3.1 Goal-setting reliability requirements

Strecher et al. (1995: 191) state that goal-setting does not evoke motivation automatically. For instance, a person has to be interested in achieving the goal in the first place or, otherwise, the effect of setting a goal will be small or the whole process may turn out to be counterproductive. In addition, they claim that it is possible for a person to fail to attain a goal even though they are interested in achieving it. For example, the goal of becoming more active and less sedentary could conflict with the goal of quitting smoking. However, if there is no significant conflict, goal setting is a viable method for achieving one's motivation towards behaviour change. What is more, Locke and Latham (2002: 714) rounds out this idea by stating that the goal-setting method is very

reliable and it could fail only if some of the task-performance mediators are not executed properly.

2.3.2 Goal-performance moderators

Three moderators[1] have been found to be involved in the goal-performance relationship – **commitment** (directly affected by the importance of the goal and the self-efficacy[2]), **feedback**, and **difficulty** (Locke and Latham, 2002: 707). Those moderators are widely used today in the modern health tracking wearables to increase the level of engagement and motivation.

The goal **commitment** moderator is affected by both the importance of the task and the self-efficacy of the user. It has been found that making a goal public increases the **importance** of a task since 'it makes one's actions a matter of integrity in one's own eyes and in those of others' (Locke and Latham, 2002: 707). For example, Endomondo (2017) – a sport tracking mobile application allows public sharing of workouts data, tagging friends and an option to challenge or get challenged by friends. Also, Locke and Latham (2002: 708) state that commitment is also affected by the **self-efficacy**. For example, finding "models" that a person can relate to by providing the users with the ability to see how they compare to users with similar performance (Endomondo).

What is more, Locke and Latham (2002: 708) found that providing feedback on the goal attainment progress affects the goal-performance relationship. Without knowing the current progress towards the goal, it is difficult for the user to adjust the effort. For example, mobile applications such as Endomondo and Runkeeper (FitnessKeeper, 2017) provide **feedback** on the goal progress to keep the user informed about their goal-performance.

Last but not least, goal **difficulty** also affects the goal-performance relationship. Locke and Latham (2002: 709) concluded that people who can set their goals usually set higher goals and that is associated with better performance than pre-assigned goals (Mitchell, and Dossett, 1978 cited in Locke and Latham, 2002: 709). For example, today's mobile fitness applications allow the user to set goals (FitnessKeeper, 2017; Endomondo, 2017) so how challenging the goal is depends on the user.

2.4 HAR enabled mobile applications

HAR is a crucial technology due to its many applications in self-management. It is involved in the monitoring stage, after setting a goal, and performs the recognition of activities such as walking using sensor data. This section will discuss the general approach to developing a HAR enabled system that detects different types of ambulation activities (e.g., walking and running) as well as explore commercial and non-commercial HAR enabled mobile applications (and devices) in the market. The non-commercial HAR solutions will be discussed regarding *sensor type and location*; *feature extraction and learning algorithm; accuracy and energy consumption.* Next, the commercial products will be discussed with regard to what attributes are gathered.

2.4.1 Design of a HAR system

Depending on the learning method, HAR systems can be categorised into supervised or semi-supervised (see figure 2.2). Supervised HAR systems require labelled data set to train a classifier. Supervised HAR systems can also be classified as, according to the response rate, online and offline (Labrador and Yejas, 2013: 29). The former provides real-time feedback to the user. The latter HAR systems are indented for applications that do not require immediate feedback due to high computations or the need of remote application server. HAR systems usually require two initial stages – training and testing (Labrador and Yejas, 2013: 4). During the training state, time series dataset is provided (see figure 2.3). Time series data [3] commonly is divided into time windows (e.g. 3 seconds) that contains the raw measurements of sensors. Future extraction algorithms are then applied to each time window in order to extract valuable information from the data

[1] A moderator is variable that affects the relationship of two other variables (Butler, 2017). For example, in the case of behaviour change a moderator would affect the goal the user set (i.e. become less sedentary) and task performance (how well the person achieves this).

[2] *Self-efficacy can be described as "the belief in one's capabilities to organize and execute the courses of action required to produce given attainments*" (Williams and French, 2011: 308) (quotes Bandura A, 1997)

[3] Torres-Reyna (2012) define time series as data collected for a specific time for one or more variables

(i.e. features). Later, different learning algorithms are used to generate a *classifier* based on the extracted features. The classifier can then classify "unseen" or unlabelled data into activities such as walking or running.

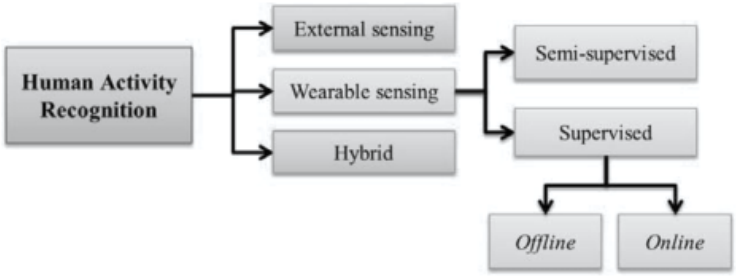


Figure 2.2: General design of HAR systems

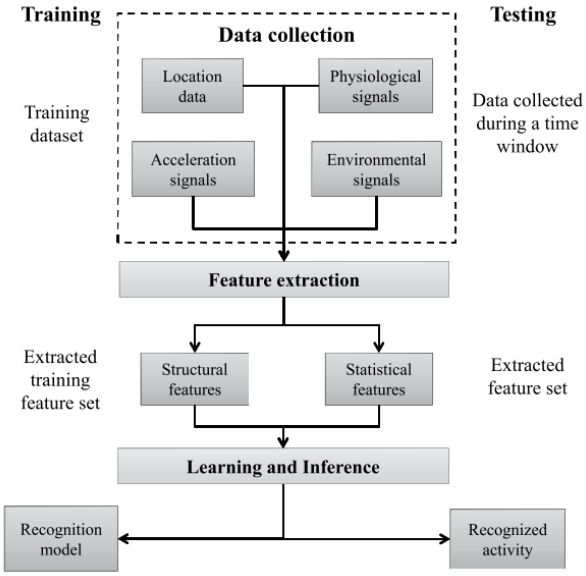


Figure 2.3: General HAR system data flow

2.4.2 Non-Commercial HAR systems

The development of HAR systems date back to the 90s, but there is still need for improvement regarding battery consumption, obtrusiveness, and accuracy Labrador and Yejas (2013: 29). Good example of HAR systems are *Vigilante* (Lara and Labrador, 2012: 38-39), *ActiServ* (Berchtold et al., 2010) and COSAR (Riboni and Bettini, 2010: 271-289).

Sensor type and location

Current and past HAR systems have been successfully implemented using one or many sensors. For example, based on system called *Centinela* (Lara et al., 2012: 38), *Vigilante* uses a sensor strap attached on the chest of the user that measures 3D acceleration and six vital signs such as heart rate, respiration rate and others. Also, there are systems that utilise just one sensor (Paul and George, 2015: 1-3). In addition, Torres-Huitzil and Nuno-Maganda (2015: 1-4) proposed a system that uses only the embedded accelerometer sensor on a smartphone and did not restrict the user of keeping the device in a specific location.

The placement of the sensors is critical and affects the accuracy grammatically. He and Jin (2008: 2245–2250) concluded that the optimal position of the sensor is inside the trousers pocket

since it is convenient. However, Lara and Labrador (2013: 1194) found that the position of the sensors depends on the application of the HAR system and the activities to be recognised. For example, placing an accelerometer on the wrist may introduce errors in the data due to incidental arm movements.

Feature selection and feature extraction

As it can be seen in Figure 2.2 after the raw data is collected from the sensors, it goes into the next stage - feature selection and feature extraction. **Feature selection** algorithms are applied to find what are the best features that can describe the given time window. According to Labrador and Yejas (2013: 21-22), using feature selection algorithms reduces the computations and thus simplifies the learning model (also called classifier).

Next, the raw data is segmented into time windows (Capela et al., 2016: 2) and **feature extraction** algorithms are performed to extract important data characteristics such as mean, standard deviation (i.e. Statistical or Time-domain features); coefficients of the polynomials (i.e. Structural or Frequency-domain features) (Lara et al., 2012: 722). Extracting both Time and Frequency domain features from the raw data is important for the overall accuracy of the system. Wang and Zhang (2015: 269) have stated that using only time-domain features is not enough to cover all of the accelerometer data. For example, Tapia et al. (2007) used both statistical and structural features to classify a total of 30 activities such as walking, cycling and lifting weights.

Having said that, a new research proposes a novel method for real-time HAR using only frequency domain features. According to Lee et al. (2016: 3), extracting the first 5 Fast Fourier Transform (FFT) coefficients for the 3 dimensions of the accelerometer, namely x,y and z (as well as the magnitude of the axis - m) - produces a higher accuracy compared to the rest of the features.

Learning algorithm, accuracy and power consumption

After the feature extraction process, different learning algorithms are applied to generate a model (e.g., classifier). For example, a system proposed by Kao et al. (2009: 116) uses fuzzy-basis-function-based classifier to classify seven activities with accuracy up to 94.71%. Also, Anjum and Ilyas (2013: 918) proposed a system on the Android platform that utilised a Decision Tree classifier. The system reached overall accuracy of 94.4%. When choosing a classifier for online activity recognition the device resources should be taken into account. As every portable device, any smartphone is constrained in terms of amount of CPU calculations it can handle and battery capacity so choosing a lightweight classifier to lower the resource consumption is crucial. In a survey Shoaib et al. (2015: 2068) and Ayu et al. (2012: 226) found out that k-Nearest-Neighbour algorithm performs best on a mobile device regarding battery consumption, CPU and memory usage. Having that said, Lara and Labrador (2013: 1193) suggest that the design of a HAR system (e.g., type of a learning algorithm, window size, etc.) depends on the training data set. That means that an examination of different classifiers is needed to find the best-performing classifier for a given data set.

Determining the *window length* is a crucial point to consider when implementing a HAR system. According to Torres-Huitzil and Nuno-Maganda (2015: 2), it affects dramatically the system's power consumption and the time taken to extract data features. Normally the window length is short such as 3 or 5 seconds. For example, Torres-Huitzil and Nuno-Maganda (2015: 3) use 5-second window length (consisting of about 200 sensor readings). Their proposed mobile system uses computations that are light on the battery and can run up to 16 hours. However, a recent journal suggests that 3-second window length (or about 128 sensor readings) produces best results when combined with kNN training algorithm on a mobile device (Bashir et al., 2016: 126-136).

2.4.3 Commercial HAR systems in the market

People, nowadays, are showing more interest in mobile fitness applications compared to 2011 and this is likely to increase even more (Google Trends, 2017). Mobile personal trainers such as Endomondo (2017), FitnessKeeper (2017) and Human (2017) help people track their activities as well as stay active.

Overall application design

Fitness mobile applications can keep a person motivated by providing goal-setting features. For example, Google Fit (Google Inc., 2017) allows the user to set goals based on time, distance or calories burned. In consequence, the task-performance gets increased (see section 2.3). Also, Human (2017) counts on the public aspect of the application to motivate users by allowing them to see how they compare to others (increase commitment, see section 2.3.2).

Measured attributes

Fitness applications rely on the embedded smartphone sensors to measure different attributes. For example, Endomondo (2017) can measure heart rate (if heart monitor is connected to the application) as well as over 40 sports/activities such as running, cycling and biking. It can monitor three times of goals, namely, distance time or calorie. In comparison, Human (2017) only focuses on monitoring minutes being active.

2.5 Proposed mobile application

Based on the above findings, a mobile application is proposed. The software application implements **behaviour change logic** to help people who spend too much time in sedentary activities change their lifestyle. For example, the application measures both – sedentary and active time and notifies the user when prolonged periods of sedentary time is detected.

Also, the application allows the user to **set personal goals** to increase goal-performance (see section 2.3). For example, the user can set a PA goal as well as maximum ST. In addition, the application allows the user to share his or her progress with other people to increase goal-commitment (see section 2.3.2).

In terms of **feature extraction** techniques, the mobile application utilises the novel approach discussed in section 2.4.2. Only 20 data features are extracted from the raw accelerometer data, namely the first 5 FFT coefficients for the three accelerometer axis (x,y and z) and their magnitude (m).

The application detects a total of four activities, namely *static*, *walking*, *running* and *cycling*. HAR is achieved by utilising a supervised machine learning approach. A **classifier** is created from labelled dataset gathered by project participants. Also, the classifier is personalised using user data to bring the recognition accuracy to an individual level for best results. Further details about the functionality of the application will be discussed in the next chapters.

Chapter 3

Specification

As with every software project, a list of requirements has to be produced in order to define the scope of the software (i.e. what it does) before proceeding to its implementation. That normally involves communication with a **client** (Bell, 2005: 23). Due to the lack of a client in this project, my supervisor agreed on playing this role. In a real life scenario this would be a rather group of people. For example, volunteers from the local health centre who suffer from SB related diseases (see section 2.2.1) could be involved in during the development of the application. The software specifications discussed in this chapter are a product of an iterative process involving workshops with the client and a continuous feedback-implementation loop.

The specification of the mobile application can be split into three components: Goal-setting, Monitoring and Feedback. The Goal-Setting Component (GSC), as the name implies, handles setting goals as well as providing additional information such as recommended amounts of physical activity per day. The Monitoring Component (MC) is the most complex part of the system. It is essentially a HAR system with logic for recognising the current activity of the user (e.g. walking or static) and logging the data into a database. In addition, this component is responsible for detecting sedentary behaviour as well as aggregating activity over different time periods. As far as the Feedback Component (FC) is concerned, it provides feedback to the user in the form of notifications.

To help with the naming conventions through the reminder of this paper, a product name has been given to the proposed software product. ***"ActiveMinutes"*** has been chosen to be the name of the application since one of the main features of the software is to encourage sedentary people to be active and potentially achieve at least 30 minutes of PA as suggested by Department of Health (2011: 7). The mobile application's specifications can be defined in terms of functional and non-functional requirements.

3.1 Functional Requirements

Functional are the requirements that concern the system directly. They state what the software product should do or how it should behave when certain conditions are met (Sommerville, 2010: 84). A common approach to identifying functional requirements is the use of use case diagrams since they show the main actors in the system and the actions associated with them. Also, use cases are used to describe the required system functionality (Bell, 2005: 45-47). This approach has been adopted in this work. The developed use case diagram can be seen in figure 3.1 and the important use cases can be seen bellow (all use cases are presented in Appendix A).

- **Set PA goal**
  The user opens the application. The system takes the user to the main screen. The user opens the navigation drawer and selects the *Settings* menu item. The system opens the Settings screen. The user selects the "Physical Activity goal" settings item. The system shows a dialog window with list of options. The user selects the desired option. The system accepts the user input and updates the system accordingly.

- **Change SB remind interval**
  The user opens the application. The system takes the user to the main screen. The user

opens the navigation drawer and selects the *Settings* menu item. The system opens the Settings screen. The user selects the "*Sedentary goal*" settings item. The system shows a dialog window providing list of options. The user selects the remind interval and confirms their choice by pressing the "OK" button. The system accepts the user input and updates the system accordingly.

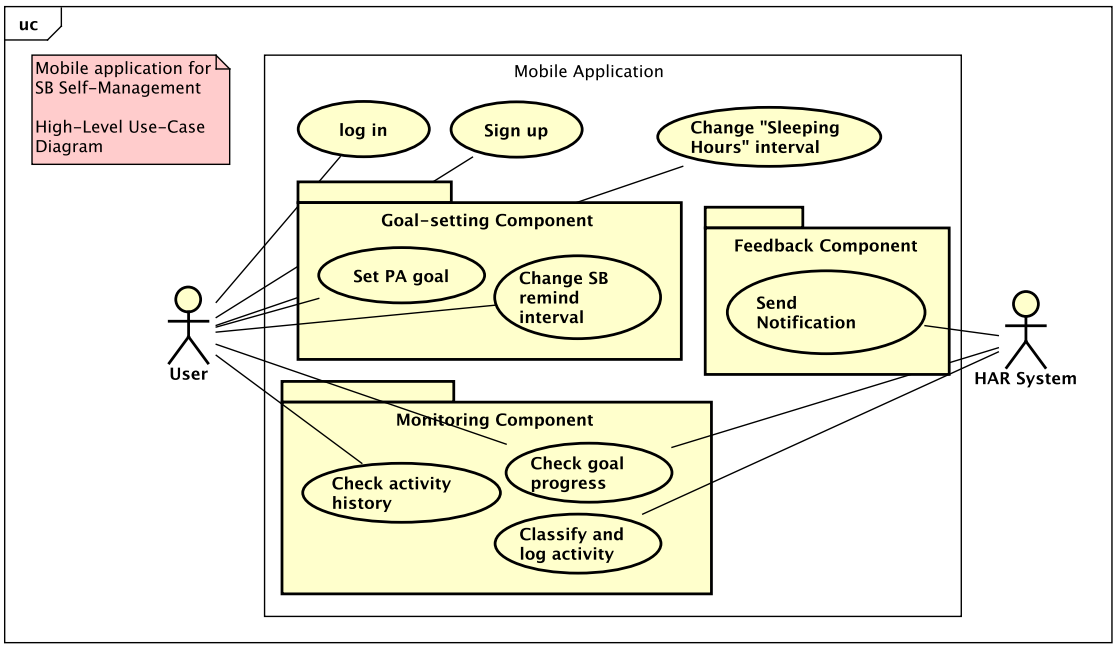


Figure 3.1: Use case diagram of *"Active Minutes"*

3.1.1 Goal-Setting Component

Goal-setting is an important aspect of the self-management and behaviour change process. This component's requirements have been informed by the British Heart Foundation (Townsend et al., 2015: 14), e.g. a minimum of 30 minutes of PA is used as a guideline. The software implementation allows the user to set goals which become a target in the first step towards behaviour change and better lifestyle.

1. GSC shall allow the user to set a PA goal
   (a) GSC shall provide list of recommended PA goals
   (b) GSC shall automatically set the recommended PA goal if the user does not explicitly do so
2. GSC shall allow the user to set a Sedentary Time (ST) goal
   (a) GSC shall allow the user to set a Maximum Continuous Inactivity (MCI) before a reminding notification is sent

3.1.2 Monitoring component

The main purpose of the MC is to continuously recognise activities and analyse the levels of PA and ST. The following requirements have been derived from analysing past and current mobile devices as well as from discussions with the client.

1. The MC component shall classify physical activities
   (a) HAR component shall recognise the following activities:

i. walking;
ii. running;
iii. cycling;
iv. static

2. The MC component shall analyse and aggregate user's PA and ST amounts in real time
   (a) MC shall increment user's PA or ST value whenever physical activity or sedentary behaviour is recognised
3. The MC shall personalise HAR's classifier
   (a) When enough user data is collected, the system shall retrain the classifier combining the general data collected from the project participants with user's own data. According to Arapakis et al. (2010: 376), that is expected to improve the classifier's classification accuracy.
4. MC shall keep history of previously accumulated PA and ST amounts
   (a) The system shall allow the user to group the collected data in "Daily" and "Weekly" representation
5. The user shall be able to set the *Sleeping Hours Time Interval* (SHTI) during which no measurement of PA or ST should occur

3.1.3 Feedback Component

The FC is responsible for sending notifications to the user. It can provide important information such as when a previously set goal is attained or reminder for the user that they are spending too much static time.

1. FC shall notify the user when they are being sedentary for 1 hour (or the value set by the user) as recommended by Swartz et al. (2011)
2. FC shall notify the user when a previously set goal is attained (e.g. being active for 30 minutes)
3. FC shall not send notifications during the user set "Sleeping Hours" time interval (e.g. 20:00 - 8:00).
4. FC shall send a notification to the user every Sunday to show goal-performance feedback for the past week

3.2 Non-functional Requirements

The mobile application proposed in this work has to comply with a number of additional requirements. According to Sommerville (2010: 87), Non-functional are those requirements that do not concern the system directly. They apply to the whole system such as performance and security. Since the GSC, MC and FC are all part of the same software, their non-functional requirements will be discussed all together.

3.2.1 Accuracy

The accuracy of the system is measured as the *Active/Static* error rate rather than the classification accuracy of the classifier on specific activity classes e.g. cycling. The system shall minimise errors where an activity of type *Active* is misclassified as type *Static* and vice versa since that would lead to incorrect PA daily readings and user may be doing less physical activity if they are notified incorrectly for attained goal. The Active/Static error rate shall be less than 10%. This number has been chosen based on two reasons. First, that error rate can only have a small effect on goal achievement, for example, if we consider a user with a goal of 30 minutes PA and a 10% error that gives us 27 minutes of PA (30 - 30*0.1) which is still close to the initial 30 minutes goal. Second, this error rate amount is achievable within the time constraints of the project.

3.2.2 Battery consumption

Since the resources of every portable device (i.e. mobile phone) are constrained, the system shall be designed so that it consumes a reasonable amount of battery power. The device shall be able to function without the need for a charge for at least 12 hours. For example, the user must not need to charge the device in the middle of the day as this would prevent the MC from monitoring activity.

3.2.3 Security

The mobile application shall store user data only on the system in order to minimise the risk of data breach. In addition, the following considerations have been taken into account:

1. The system shall show authentication screen before showing any data such as how active the user has been during the day
2. The system shall encrypt the data stored in a local database

3.2.4 System Quality Attributes

Adaptability

Modern mobile applications should be able to quickly adapt to new requirements as the market changes. The intention is to achieve adaptability by implementing the MVP software development pattern. That allows the system to incorporate new features easily as the system is separated into loosely coupled logical parts.

Scalability

The system should be scalable to accommodate large amounts of data collected by different users. For example, the system should be able to handle the data for at least five users.

Modularity

As living in a fast-changing world, user requirements can change quickly in a short period of time. Thus, the system should be able to change or replace features so it can adapt to change. For example, each mobile screen's logic is encapsulated in a separate directory which can be interchanged easily without affecting the integrity of the system.

Testability

Testing any system is crucial since it allows the discovery of unwanted system behaviours. To assure that the system works as intended, the system shall be tested by both assertions and Quality Assurance tests.

Maintainability

As with every software product, the need for maintains is unavoidable. Thus, to ease this process, the system shall be designed in such a way so that it allows for easy evolution. For example, extending the system by adding new features.

Robustness

Since the system requires a constant monitoring of the user (e.g. collecting contextual data), it is imperative that the system encompasses as many points of failure as possible. Thus the system shall be taken through series of QA tests as well as unit tests to minimise the points of failure.

Chapter 4

Design

This chapter discusses the design considerations during the design stage of the system. This stage of the development is very important since it lays down the foundation for the whole system. Thus a considerable amount of time has been spent to carefully plan how the system should be designed. Bell (2005: 12) states that about 5% of the total time that takes for the development of software should be spent for the design state alone. For comparison, the coding process takes about 7%; Testing takes 8%.

The design process has gone through many iterations to ensure the best possible quality of the product. The research presented in chapter 2 had a major impact on the design. The design can be split into four main sections, namely Mobile Platform and IDE; Database Design; Architectural design; and User Interface (UI).

4.1 Mobile Platform and IDE

The first point that was considered in the design process was the **mobile platform** on which the proposed application would be implemented and distributed. After research on the current market, Google's *Android* was found to dominate in Europe (Williams, 2016). Android-based smartphones hold 75.6% of market share dominating other mobile platforms such as Apple's *iOS* (18.9%) and Microsoft's *Windows Phone* (4.9%). Thus, Android OS was chosen to form the basis of the proposed mobile application since it allows the application to be downloaded and used by as many people as possible. In addition, the platform provides an excellent opportunity to reinforce the target market as sedentary people rather than fitness focused users who may favour Apple.

After choosing the mobile platform, the development environment or the **Integrated Development Environment** (IDE) was picked. That was an easy decision since *Android Studio* is known to be the official IDE for Android development (Android Studio, 2017). It provides all the necessary developer tools such as *Android Emulator* which is a virtual phone that allows a developer to test an application on different versions of the platform. In addition, it supports direct integration with *Git* which is the version control system used in this project.

4.2 HAR

The Human Activity Recognition process involves standard stages such as *Data Preprocessing*, *Feature Extraction* and *Feature Selection* (see section 2.4.2). The block diagram of the HAR process can be seen in figure 4.1. It shows the main stages of the HAR process implemented in this project. The raw accelerometer data is segmented into *Time windows*. After that it undergoes the *Data Processing* stage where the data is pre-processed, e.g., removing the earth's gravitational force from the data. Next, *Feature Extraction* process abstracts the data into 20 features. The *Feature Selection* process is responsible for reducing the number of the features without compromising the data quality. Finally, the *Classifier* determines what class the recognised activity belongs to (e.g. walking).

Weka[1] Data Mining Software has been chosen for the implementation of the HAR component of

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

the proposed system. Since there is no official support for the Weka API on the Android platform, a stripped-down version[2] of the original Weka Java API has been chosen. This library was tested prior to the design stage to ensure it functions well on an Android device.

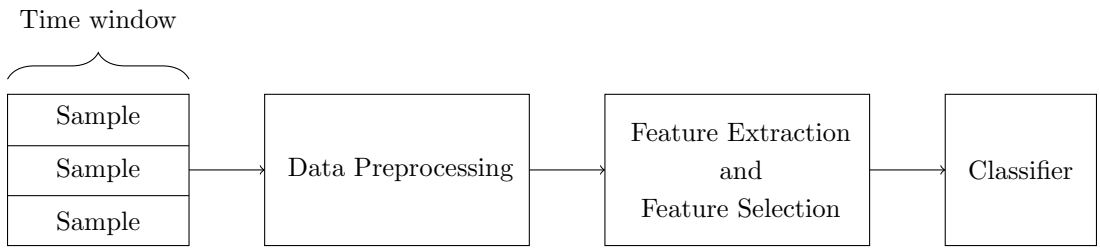


Figure 4.1: Human Activity Recognition process

4.2.1 Dataset

"Active Minutes" is using a *Supervised* learning method. This learning method require labeled *"training data"* or *"data set"* in order to build a classifier (or model). After a classifier is build it can be used to classify *"unseen"* or *"unlabelled"* data acquired by the build-in accelerometer sensor in the device.

4.2.2 Data Processing

The data processing stage of HAR normally involves segmenting the data into time windows. Considering the fact that the system will be running on a mobile device with limited resources, the window length has to be picked carefully. For example, choosing a 20-second window length (see figure 4.1) will cause the CPU to do more work due to the number of *samples* pulled from the accelerometer sensor in the time window. Consequently, that would lead to a higher battery consumption (Torres-Huitzil and Nuno-Maganda, 2015: 2). As it was found in the literature review (see section 2.4.2) 3-second time window produces good results regarding power consumption. Based on those findings, the proposed application will adopt this window length as well.

During this stage, different filters could be applied to the raw data to remove unwanted information. For example, according to Google (no date), the raw data from an accelerometer sensor contains the earth's gravity force as well. This will cause the device to read magnitude of g = 9.81 m/s2 when it is not accelerating (i.e. laying down on a table). In order to obtain just the linear acceleration without natural forces (i.e. earth's gravitation), a low-pass filter is applied.

In order to reduce the effect of device location (e.g. where the device is) the magnitude of the accelerometer's x,y and z-axis are also calculated during this stage and used as a 4th axis *"m"* (see equation 4.1). This technique has been used in previous works as well (Torres-Huitzil and Alvarez-Landero, 2015: 153).

$$m = \sqrt{x^2 + y^2 + z^2} \tag{4.1}$$

4.2.3 Feature Extraction and Feature Selection

Feature extraction is an important process in a HAR system. It allows extracting key characteristics from raw data which can be used to represent important patterns with reduced dimensionality (Torres-Huitzil and Alvarez-Landero, 2015: 154). During the initial research (see section 2.4.2) it was found that extracting the first FFT algorithm coefficients produce higher accuracy compared to the other features (e.g. Time domain features such as Mean or Standard Deviation). To ease the development process, a third-party library called *JTransforms* [3] has been utilised to convert the Time-Domain raw accelerometer data (data collected over a time period) into a Frequency-Domain data by applying the above-mentioned algorithm. The total number of features that are extracted

[2] `https://github.com/rjmarsan/Weka-for-Android`
[3] `https://github.com/wendykierp/JTransforms`

from the raw accelerometer sensor is 20. That includes the first five FFT coefficients of the x,y and z-axis as well as their magnitude.

Feature selection algorithms can be applied after the Feature Extraction process to reduce the number of features, and thus simplifying the classifier. The main goal of the feature selection algorithm is to select a subset from the original data features that can accurately carry the original meaning of the data (Wu et al., 2008: 22). For example, in this work after applying the feature selection algorithm, the original number of features is reduced from 20 to 16 which should lead to a faster CPU processing. A Correlation-based Feature Selection (CFS) algorithm has been chosen for the implementation of the feature selection process since it showed good results in previous works (Le and Nguyen, 2015: 220-224). The CFS algorithm selects a subset of features based on the correlation between the feature-class and feature-feature value. The former shows how much a feature is related to a certain class (i.e. such as *"walking"* or *"static"*) (Michalak and Kwasnicka, 2006) whereas the latter shows how individual features relate to each other.

After applying the above-mentioned algorithm over the 20 extracted features, it reduced the initial number to 16, removing four unnecessary features from the dataset. The selected features (after applying the algorithm) can be seen in listing 4.1. The classification accuracy was not changed after this stage. All of these observations were carried out offline using the PC version of WEKA software. The appropriate changes were done on the device accordingly (e.g. applying the CFS algorithm pragmatically on the device).

```
Selected attributes: 1,3,4,5,6,8,9,10,11,13,14,16,17,18,19,20 : 16
                     accX__fft1
                     accX__fft3
                     accX__fft4
                     accX__fft5
                     accY__fft1
                     accY__fft3
                     accY__fft4
                     accY__fft5
                     accZ__fft1
                     accZ__fft3
                     accZ__fft4
                     accM__fft1
                     accM__fft2
                     accM__fft3
                     accM__fft4
                     accM__fft5
```

Listing 4.1: CFS algorithm result over the initial feature set of size 20

4.2.4 Classifier

After the Data Processing and Feature Extraction and Feature Selection stages a classifier is built. The initial research suggested the use of the k-Nearest Neighbors (kNN) algorithm to build the classifier due to the fact that it does not require much device resources such as CPU calculations (see 2.4.2). Based on those findings, the proposed mobile system employs the same learning algorithm.

There are several advantages of training a classifier for the project and not using solutions such as *Google Fit*[4]. The mobile application would not require a Wifi connection to monitor the user. Commercial applications such as *Human* (Human, 2017) use the *Google Fit* platform to monitor the user which renders them useless if no Internet connection is available. The reason for this is that applications using Google Fit services need to process the collected information offline. Another advantage of building a classifier for the project is that it can be personalised in the future. When enough data is gathered from the user, it can be used to retrain the model whereas that would not be possible if using the *Google Fit* recording API since the application would be using external general classifier offline.

[4] `https://developers.google.com/fit/`

4.3 Architectural design

This section discusses the software software architecture of the mobile application. The architectural design is the first stage in the design process of a system. It creates a critical link between the requirements discussed in Chapter 3 and the research knowledge gained in Chapter 2. The end-goal of the architectural design is to discover the major structural components of the system and how they communicate to each other (Sommerville, 2010: 148). The architectural model of the system can be seen in figure 4.2.

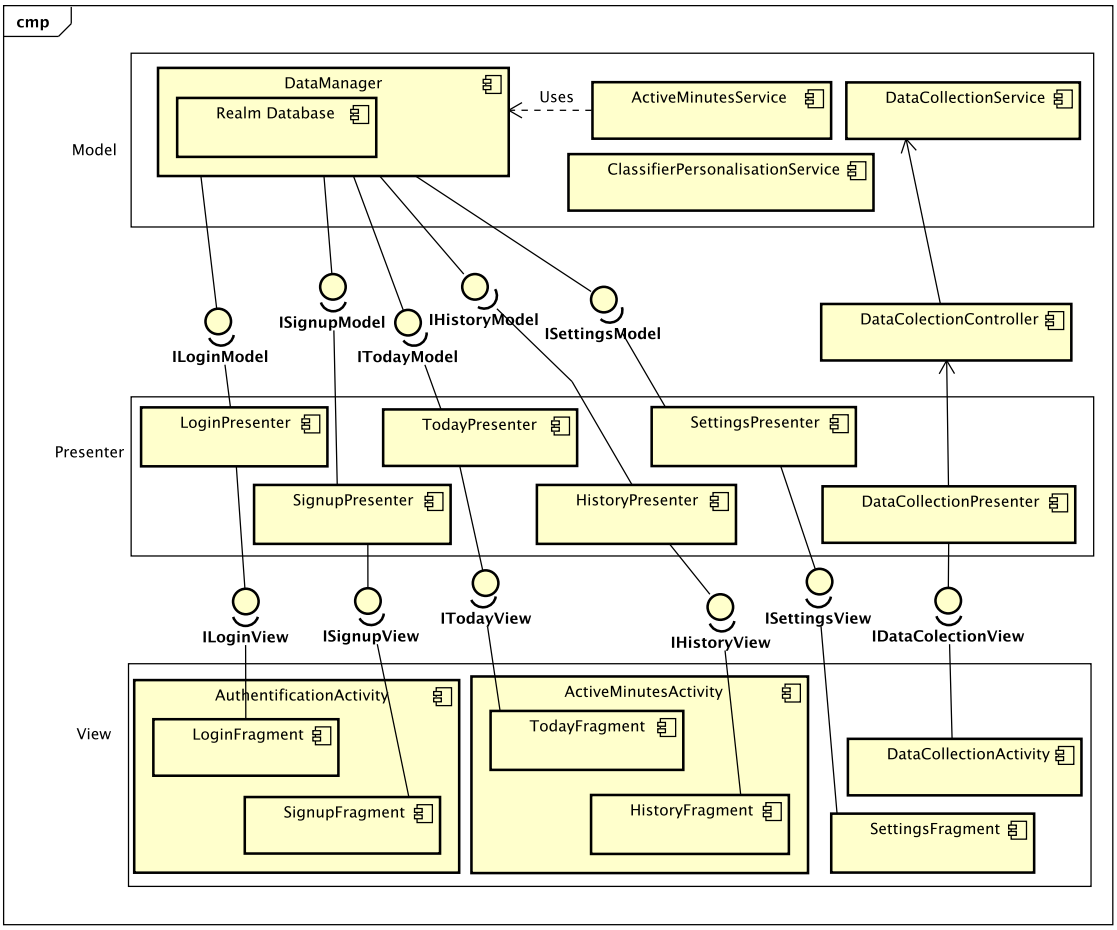


Figure 4.2: *"Active Minutes"* Architectural model (simplified)

Model-View-Presenter or **MVP** software architectural pattern was chosen to form the base of the proposed mobile system since it provides clean separation of concepts (Zhang and Luo, 2010: 532). For example the *View* represents the UI of the application. It contains logic for receiving user interaction and notifying the *Presenter*. The *Model* is the data layer. It is responsible for providing and formatting (i.e. applying the business logic to) the data and returning it to the Presenter. The *Presenter* acts as a mediator, it requests data from the *Model* and returns it to update the *View*. Figure 4.3 shows the interaction flow of the pattern. When the user interacts with the UI of mobile application, the View receives the interaction (i.e. button click) and passes that information to the Presenter via Interface (*IView*). The Presenter then invokes the appropriate method of the Model. Next, the Model returns the requested data to the View (via the Presenter).

The main **advantages** of using the MVP pattern is that it increases the **testability** of the software via the concept of logic separation. For example, the View is only allowed to have framework specific code (i.e. Android UI components such as Button, ListView); Presenter and Model are comprised of Plain Old Java Objects or **POJO**. These POJO classes can be thoroughly tested outside the mobile application. In addition, the MVP pattern introduces a certain level of abstraction in the software architecture via the use of Java Interfaces such as *ILoginView* (see figure

4.2). That allows different parts of the software to be developed separately and then integrated into the application. The components of the system will be discussed in the following sections.

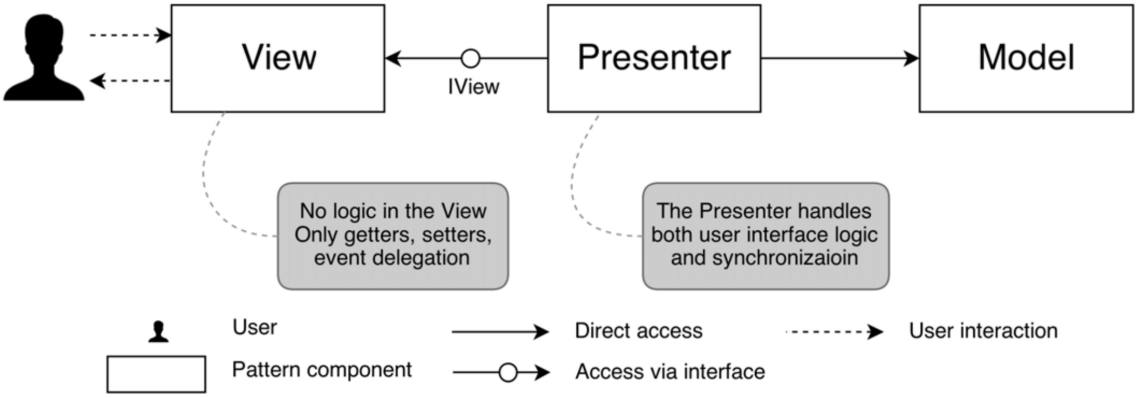


Figure 4.3: Model-View-Presenter design pattern (Syromiatnikov and Weyns, 2014: 28)

4.3.1 Models

As can be seen from figure 4.2 the *Model* layer of the system architecture is comprised of *DataManager*, *ActiveMinutesService*, *ClassifierPersonalisationService* and *DataCollectionService*. These components will be discussed next.

DataManager

The ***DataManager***, as the name implies, is responsible for providing means of accessing the underlying database object. For example, retrieving a list of activities for the logged-in user. Almost all of the diagram components have access to DataManager. Thus they can perform operations such as saving a new user or loading user's physical activity goal. This class essentially is a wrapper of a **Realm**[5] database object. After initial research, Realm (2014) was chosen to form the database of the mobile application. It is a third-party Android library found to be x100 faster compared to the Android native SQLite Object-Oriented Data Management System (OMS).

ActiveMinutesService

Since *"Active Minutes"* has to collect contextual information throughout the user's day continuously, ***ActiveMinutesService*** is implemented as an Android *Service* class. According to Google (2017g), a Service is an application component that continuously runs in the background even if the user closes or switches to another application. The responsibilities of this service include HAR, storing the recognised activities in the database and performing additional checks such as whether or not to send reminder notification to the user for sitting for too long.

Another important responsibility of the this service is to collect personal data. This data is used later on for classifier personalisation (e.g. adapting the general classifier so it better recognises user activities).

ClassifierPersonalisationService

The sole purpose of this service is to personalise the general classifier that is included in the application initially. The service is started by the above-mentioned *ActiveMinutesService* service when enough labelled data is collected. For example, the service is executed when there are x100 instances of *walking*, *running*, *cycling* and *static* activity training data. Since it will run once in the lifetime of the application on the device, this service is implemented as a special kind of Service called *IntentService*. It allows for executing heavy tasks such as training a classifier in a worker thread so the application's main UI thread is not affected.

[5] `https://realm.io/news/realm-for-android/`

DataCollectionService

In order to train a HAR classifier, a training dataset is required. This service is responsible for the data collection process. That includes reading, prepossessing and windowing the raw accelerometer data as well as abstracting the data via the means of feature extraction (see section 2.4.2). Each feature vector stores 20 values (the first 5 FFT coefficients of every sensor axis, namely x,y and z and their magnitude) as a record in the *Training_data* table (see figure 4.4).

4.3.2 Views

The View layer of the MVP pattern is comprised of Interface classes that are implemented by Android components such as *"Activity"* or *"Fragment"*. This approach provides a layer of abstraction and logic separation which is one of the benefits of using the MVP design pattern. From an Android standpoint, an *Activity* is a class that creates a window (e.g. single screen) and its layout (a specific arrangement of UI components such as *"Buttons"*) can be set via using the method *"setContentView(View)"* (Google, 2017b). According to Google (2017e), *"Fragment"* is a *"sub-activity"* like class with its own layout. They can be swapped with one another easily thus they are highly reusable. Fragments typically require a host Activity in order to be presented to the user. For example, the *"LoginFragment"* and *"SignupFragment"* are hosted in an Activity called *"AuthenticationActivity"*. As it can be seen from figure 4.2 all of the *Views* in the MVP pattern communicate with a dedicated Presenter class in the *Presenter* layer via an interface. The following sections will describe what are the responsibilities of each one of those components.

AuthenticationActivity

As the name of the component suggests, this is an Android *Activity* class that handles the login and sign-up the functionality of the application. This activity is responsible of presenting the two fragments it hosts, namely *"LoginFragment"* and *"SignupFragment"*. The LoginFragment is in charge of checking the login user credentials and eventually transferring the user to the main screen of the mobile application. As for the SignupFragment, it handles the process of user registration.

ActiveMinutesActivity

The main purpose of this activity is to display the contents of the *"TodayFragment"* (e.g. Today screen) and *"HistoryFragment"* (e.g. History screen). The TodayFragment shows current information to the user such as how much time a user has been active today as well as details for the user's inactive time. The **HistoryFragment** is responsible for showing past activity information both daily and weekly. This way the user can refer to a previous day goal-performance.

DataCollectionActivity

This Android *Activity* class is responsible for providing an interface for controlling the *DataCollectionSertice* mentioned in section 4.3.1. Its UI provide buttons for starting and stopping the service as well as exporting the acquired data into a labelled dataset file format - *ARFF*. Later on, this file will be used to build a classifier using the machine learning library WEKA on the device itself.

SettingsFragment

As the name implies, this Fragment class is responsible for providing the user with the ability to change certain parameters of the application. For example, the user can change their PA goal as well as ST goal. In addition, the user can change the *"Sleeping Hours"* time window and disable MCI reminding notifications.

4.3.3 Presenters

The *Presenter* layer of application's architecture is comprised of a set of Presenter classes (see figure 4.2) needed for the different screens of the application. For example, the *LoginFragment* (e.g. screen) has a presenter object *LoginPresenter* which allows the screen to participate in the

MVP pattern e.g. communicate with the appropriate *Model* and *View* objects, namely *ILoginView* and *ILoginModel*.

4.4 Database Design

This section discuses the design of the database system that will be used to store the data needed for the functionality of the mobile application. The design of the database can be seen in figure 4.4. It includes the following entities **User**, **Activity** and **Training_data**. The following sections will explain the responsibilities of each entity (i.e. database tables).

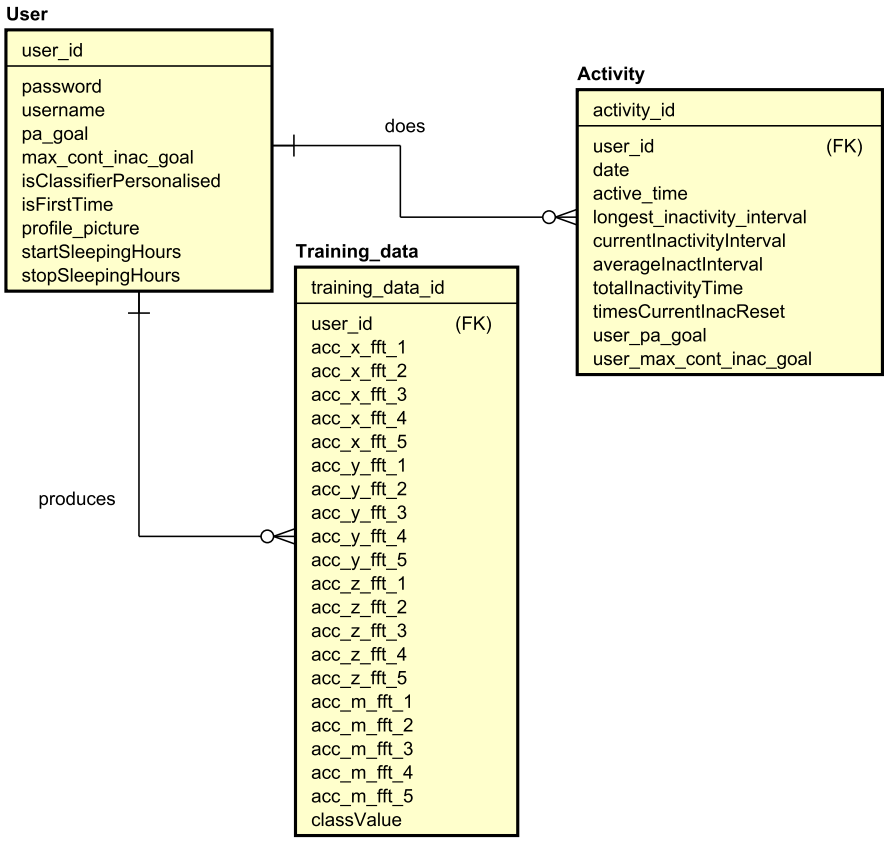


Figure 4.4: Database ER Diagram

Entities

- User(user_id, password, username, pa_goal,max_cont_inac_goal, is_classifier_personalised, profile_picture)
- Training_data(user_id, acc_x_fft_1, acc_x_fft_2, acc_x_fft_3, acc_x_fft_4...)
- Activity(activity_id, user_id, type, date, duration)

Relationships

- Does — User does Activity [1:m][m:o]
- Produces — User produces Training_data [1:m][m:o]

4.4.1 User

The user table is responsible for storing the personal information of every registered user. For example, ***user_id*** allows every user to be uniquely identified in the system. As for the ***username*** and ***password*** fields, they are used to authenticate different users of the application upon login.

The user table stores additional information such as the PA goal (***pa_goal***) and the maximum continuous inactivity goal (***max_cont_inac_goal***). The table also stores a Boolean field called ***isClassifierPersonalised***. This field is set to *False* by default and only changed when the general classifier, bundled with the application, is personalised with user's data. Since every user sets their *Sleeping Hours* by design (i.e. a time window during which the user would not be monitored) fields called *startSleepingHours* and *stopSleepingHours* are needed to store the start and stop sleeping hours set by the user. Last but not least, fields called ***profile_picture*** and ***isFirstTime*** are responsible for storing the user's profile image and if the user uses the application for the first time, respectively.

4.4.2 Activity

As it can be seen from figure 4.4, the Activity table is responsible for storing information for the different activities. Every table record is created for every day of the week. Once created, its fields are amended accordingly throughout the day depending on the PA and ST levels.

Every activity record in the table has a unique identifier ***activity_id*** that uniquely identifies it. Also, the table stores the user's id in a field called ***user_id***. That allows for associating a given activity record with a specific user.

The table is also storing fields that store user's activity levels. For example, ***active_time*** field stores the amount of PA the user is accumulated for a specific day. Other table fields are responsible to storing statistical information regarding sedentary information such as *longest_inactivity_interval*, *currentInactivityInterval*, *averageInactivityInterval* and *totalInactivityTime*. In addition, a field called *timesCurrentInactivityReset* is used to store how many times the current inactivity is being reset (needed to calculate the average inactivity interval). A reset happens when an activity of type *Active* is recognised by the classifier. Consequently, the *currentInactivityInterval* is reset to zero and *timesCurrentInactivityReset* incremented.

Activity table also stores user's PA goal, represented by *user_pa_goal* as well as ST goal represented by *user_max_cont_inac_goal*.

4.4.3 Training_data

The last database table to discus is the Training_data. The fields from the table ***acc_x_fft_1***, ***acc_x_fft_2***,***acc_x_fft_3***... are responsible for storing the extracted features from the raw accelerometer data as discussed in section 2.5. The table field ***user_id*** is used to associate a user with a particular record in the table. *ClassValue* field stores the label of the training data record (i.e. walking or static).

One point to note, this table is used for two purposes at different times of the development of the project. Firstly, it is used during the application development to store all of the project participants training data. This data will be used to build a general classifier. Secondly, this table is used as part of the final solution to store labelled data needed to build a personalised classifier for a particular user.

4.5 User Interface Design

The process of designing the user interface is highly dependent on the system's specifications discussed in chapter 3. For example, the application requires user authentication thus a design for the log-in and sign-up screens have been developed. A UI design process proposed by Bell (2005: 60) has been applied in this work (see figure 4.5). According to Bell, the UI design process is usually informed by certain design principles and guidelines to make sure that the application is user-friendly. Next, prototyping tools are used to create a system prototype which is evaluated by the user for quality. That is followed by integrating the client's feedback into the prototype. This process is repeated until the client is happy with the prototype of the system. The following

sections discuss how the mobile application UI has been developed as well as what design principles and guidelines were considered during this process.

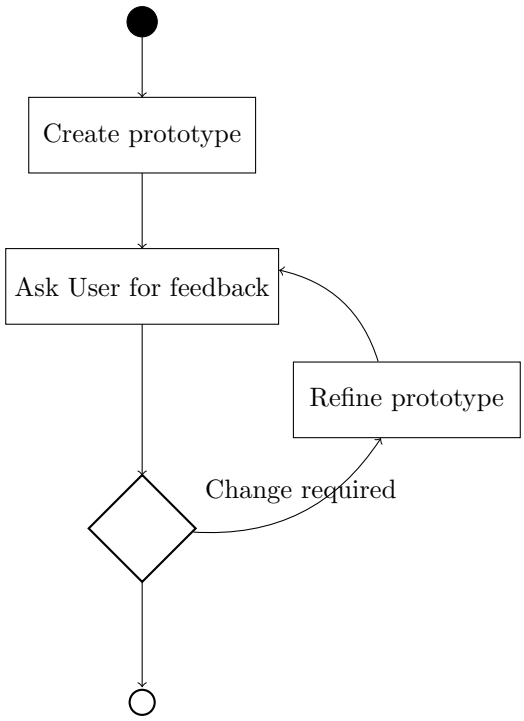


Figure 4.5: Iterative process of UI prototyping

4.5.1 System Prototyping

In order to create the system prototype, a set of wireframes were created (see appendix B). Next, the wireframes were further developed into a UI design. Both the wireframe and user interface designs were created using a professional software design tool *Sketch*[6]. The design of every mobile screen implemented in mobile application the can be seen in appendix C.

When the initial versions of all mobile screen UI design were finished, a software called *Flinto*[7] was used to create a system prototype. One of the reasons of choosing *Sketch* as a design tool is because it allows for easy export of Sketch pages to *Flinto* documents thanks to a dedicated plugin.

In order to create the prototype, special objects called *Links* had to be drawn over the mobile screen designs . Every link needs a target to be specified. For example, in figure 4.6) there is a link object drawn over the *"Weekly"* button and its target is set to be the whole *"History Screen Weekly"* screen. So when the client inspects the design prototype and presses the *Weekly* button they will be taken to the weekly view page. This process of creating links and targets was used for all UI screen designs to create a functioning system prototype that the client can interact with and gain a clearer idea of how the system will be implemented and share valuable feedback.

4.5.2 Design Principles and Guidelines

Design Principles

Design principles are well-accepted principles that can inform how a user interface is being designed. Platform specific principles have informed the system design. For example, the first design principle suggested by Google (2017a) **"Delight me in surprising ways"** have been implemented in the authentication screens as the ability of the user to change screens via a swipe gesture. This action is reflected by the *page indicator* at the bottom of the screens (see figure C.3). According to Google, subtle effects such as these contribute to a feeling of effortless.

[6] `https://www.sketchapp.com/`
[7] `https://www.flinto.com/`

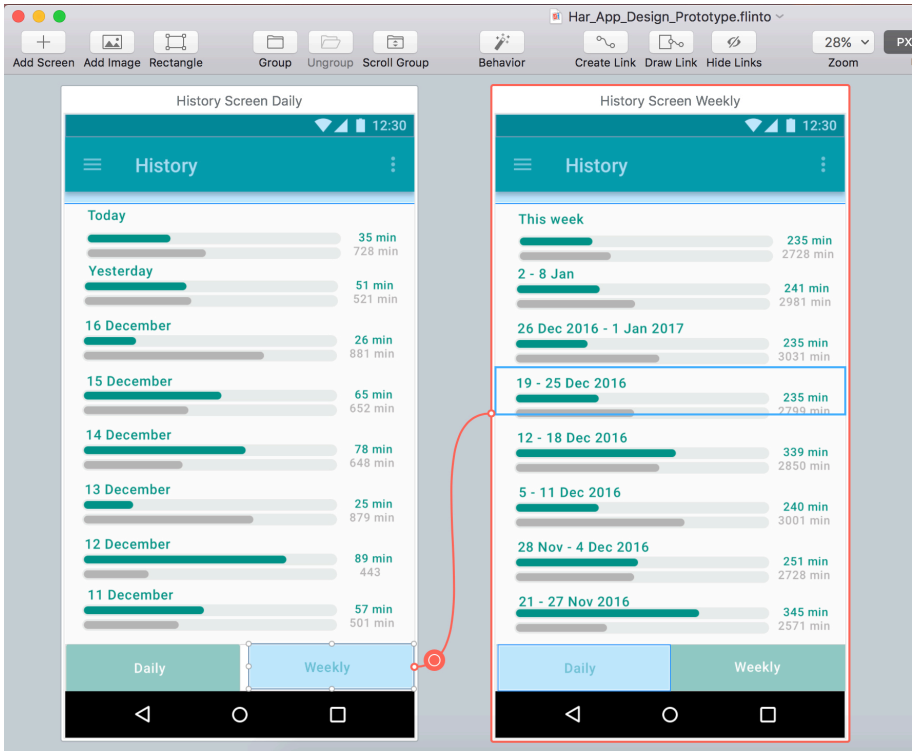


Figure 4.6: Flinto prototype creation process

"Only show what I need when I need it" is another principle suggested by Google (2017a). It focuses on the fact that people get confused when presented with much information at once. *"Active Minutes"* incorporates this principle by providing a *"Navigational Drawer"* that provides additional options for the user such as *"Settings"* and *"Log Out"*. When not needed, the drawer can be closed (see figure C.1a).

"I should always know where I am" is a principle that allows the user to gain confidence that they know how to use the system. For example, when switching between the *Login* and *Signup* screens the text of the button is changed to *"LOG IN"* and *"SIGN UP"* accordingly so the user can infer where they are at the moment (see figure C.3).

As per system specifications, the system will notify the user when a goal is met (see section 3.1.3). The next principle **"Only interrupt me if it is important"** implies that the user should not be interrupted unless it is important. For example, the system sends a notification when a prolonged sedentary time is recognised. In this case, the user must be notified so they can take action.

Design Guidelines

Bell (2005: 57-59) states that Design Guidelines need to be considered in the context of the platform the software is being designed for. Since *"Active Minutes"* is developed as an Android application, it needs to comply with the platform specific design guidelines. That is why the system design has been informed by the official Android design guidelines (Google, 2017c). Google's new design language called *"Material Design"* have been applied to the design of the application. It provides various guidelines such as the ideal layout metrics to be used when designing a *Navigation Drawer*.

Chapter 5

Development Methodologies

This chapter discuses the development practices that are used during the development process of the proposed system.

5.1 Development Model

As with every software product, specific rules are needed to ensure that the software is developed in a manner that satisfies its specifications and achieves the end-goals. In the IT industry that is achieved by following a software development model. It specifies the required stages of the development and the order in which they are carried out. In this project the *Incremental Model* has been chosen to drive the development of the software. The main idea behind this model is developing a system through iterative steps and in small incremental portions (Bhuvaneswari and Prabaharan, 2013: 264). It ensures that each iteration (e.g. software build) is tested and working before using the previous iteration as a base for the next one (see figure 5.1). The main benefits of using this model are:

- A product is produced early in the project's development
- Because of the previous point, unforeseen problems can be detected early
- Client change requests can be implemented between increments

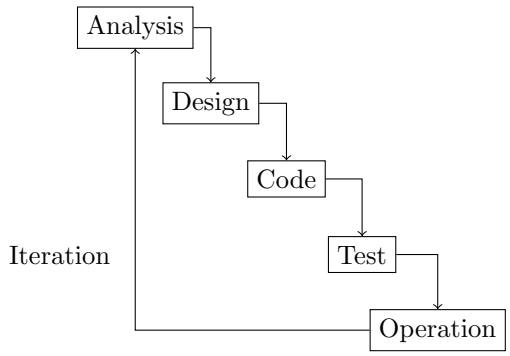


Figure 5.1: Incremental software development model life cycle

5.2 Version Control System

When developing software challenges can occur unexpectedly, and a developer has to do whatever possible to ensure their work is protected. For example, an untraceable bug can be introduced in one of the iterations of the iterative development model (see section 5.1) that could lead to poor system performance. In order to prevent the (or at least dramatically lower)the occurrence these situations, the use of *"Version Control"* is encouraged.

There are many Version Control Systems (VCS) out there such as Mercurial, Fossil and Git. The main purpose of a VCS is to keep track of the changes in a project. If a developer is satisfied with a specific change, they can perform a *commit*. That action will merge the change to the codebase, and the commit command will be added to the changelog (list of commits). If there is a problem that has been caused by a previous *"bad"* commit, the developer can simply revert the whole project to a more stable state by restoring the codebase to a previous known *"good"* commit.

For this project, Git has been chosen along with its web service GitHub[1] to serve as a VCS due to the fact that it is easy to use and it is well-documented.

5.3 Project Management

It is important to stay organised when it comes to working on a project, especially when it involves the usage of limited resources (i.e. Time). Thus, a project management tool called *Taiga.io*[2] was utilised to help keep track of the project's progress via ***Kanban* boards**. With every incremental iteration (see 5.1 above) new tasks are identified, created and added to the Kanban boards.

Using *Kanban* boards introduces several benefits to the development of the project. Firstly, it provides a quick way to organise the project work visually. For example, boards titled *"READY"* and *"IN PROGRESS"* tables contain that tasks that are already implemented and what those that are being implemented, respectively (see figure 5.2). Secondly, it increases the development efficiency since one of the key aspects of the Kanban methodology is to limit the work in progress (WIP) by setting a limit (Atlassian, 2017). For example, if the number of tasks in *IN-PROGRESS* board is 1, that means that no other tasks should be started until the currently undergoing one is finished.

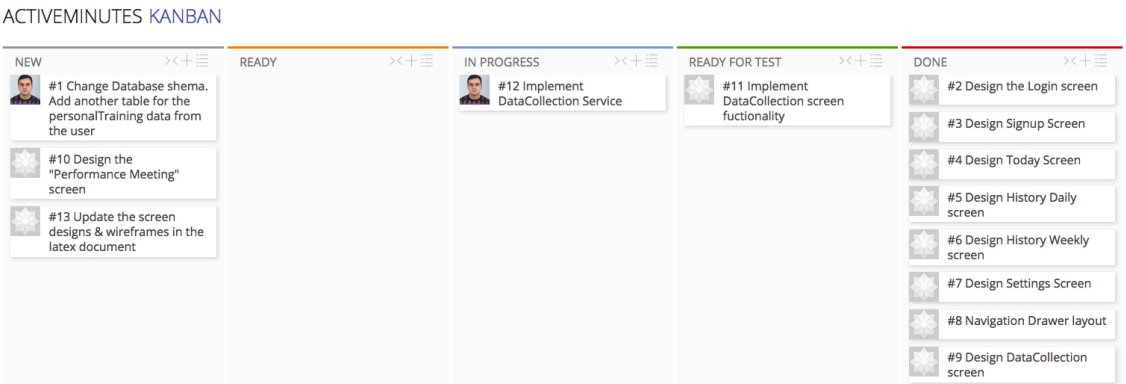


Figure 5.2: Taiga Kanban boards

5.4 MVP and Dependency Injection

As discussed before, the implementation of the mobile application is following the MVP software pattern, which allows for the clear separation of code concepts (see section 4.3). To better satisfy the requirements discussed in section 3.2.4, MVP is combined with a Dependency Injection (DI) principle. What DI do is shifting the responsibility of one class to create its dependencies to another class. For example, class Car requires class Engine as a parameter. Instead of creating a new instance of class Engine when class Car is created it is simply provided as a constructor parameter in class Car's constructor (see listing 5.1).

Dagger[3] is a framework that utilises the DI principle in the form of *Component* and *Module* classes. A module creates the necessary dependencies (e.g. object of class B) whereas the Component class determines where those dependencies will be "injected" (e.g. satisfy dependence's in class

[1] `https://github.com`
[2] `https://taiga.io`
[3] `https://github.com/google/dagger`

A). A concrete example of DI can be seen in Listing D.3. In this example, the **LoginFragment** (e.g. the login screen) requires an object of type ***"ILoginPresenter"*** to function. The method ***"satisfyDependencies()"*** provides the dependencies to the class (e.g. externally). All of the dependencies of the Login screen are provided from a Dagger module named ***"AuthModule"*** which is responsible for providing and creating the concrete implementation of the above interface (e.g. ILoginPresenter). The full code of ***"AuthModule"*** as well as ***"AuthComponent"*** can be seen in Appendix D.

Combining the MVP design pattern with dependency injection framework such as Dagger brings many advantages for the software product. Firstly, the software becomes more loosely coupled as the MVP enforces the use of interfaces and Dagger is responsible for injecting the concrete implementation of those interfaces where required. Secondly, it allows for logically structuring the codebase. For example, a common approach when using both MVP and Dagger is to create a directory for each one of the mobile screens (see figure 5.3) containing the necessary MVP classes such as ***Presenter*** and ***Model*** and the as well as Dagger's ***Module*** and ***Component*** to handle the DI process.

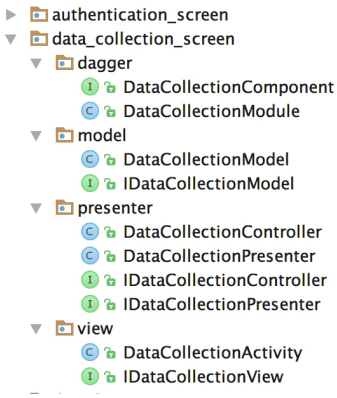


Figure 5.3: Data collection screen directory structure

```
class Car{
    Engine engine;
        Car(Engine engine){
            this.engine = engine;
        }
    }
}
```

Listing 5.1: DI example

Chapter 6

Implementation

This chapter discusses the implementation details of the proposed mobile application. First, the HAR stages will be discussed. That includes the data gathering process from the project participants as well as data prepossessing and training of the kNN classifier. Next, the self-management system will be discussed followed by the specifics of the mobile application itself such as database and overall system design implementation.

6.1 HAR System

In order for the system to take actions such as notifying the user for a prolonged periods of inactivity or sending a notification when a PA goal is achieved, the application needs to recognise human activities (i.e. walking and running). Implementation of the HAR system will be discussed bellow.

6.1.1 Data Collection

One of the key stages in the development process of a typical supervised (online) HAR system is the data collection stage. A labelled dataset is needed in order to train a model (or also called classifier). The produced model can classify an "unseen" (or unlabelled) data to a specific activity (i.e. walking or being static).

Setting and project participants

The data needed to train the model in this work is collected from three project participants in a controlled environment. Each participant signed a consent form before the data collection process to ensure they understand the purpose of the project (the consent forms can be seen in Appendix N). The participants were equipped with a device running partially implemented *"Active Minutes"* application (e.g. only the *"Data Collection screen"* see C.4a). Before performing the data collection process, the participants had to, first, select the activity they will perform from a list of activities and then press the Start button on the application's UI in order to start the data collection process. The location of the device during the recording was chosen to be the front pants pocket. This location has been found to be the optimal position for HAR (see 2.4.2). All of the participants recorded 3 minutes of each one of the following activities: *walking*, *running*, *cycling* and *static*. To avoid any unwanted information during the recording stage, the first and the last 7 seconds of each activity was removed since it contained noise information such as the linear acceleration taking place when the device was put in and pulled out of participants pockets. That was pragmatically done and no direct human intervention was needed to further filter the data.

Recorded data

When all of the required data was collected, it was converted into a WEKA ARFF file format by pressing the "EXPORT" button on the Data Collection screen. The ARFF file was stored on the

device's external memory (e.g. SD card). A sample of the produced ARFF file can be seen in Appendix E. A dataset containing 720 labelled data points (or a total of 36 minutes of data) was produced as a result of the data collection process.

The device used to collect the data was *OnePlus One* equipped with low-power high performance 3-axes accelerometer [1]. As for the sampling frequency (SF) of the sensor, the Android Operating system provides a total of four different sampling frequencies for reading the accelerometer sensor, namely *NORMAL: 5 Hz*, *UI: 16 Hz*, *GAME: 50 Hz*, and *FASTEST*. The latter has been chosen for the SF of the built-in accelerometer as it has produced good results in Lee et al. (2016: 3-5). According to the Google's documentation, *FASTEST* SF depends on the hardware of the device (Google, 2017f). In this case, the SF was **115** Hz.

6.1.2 Training the kNN classifier

The exported ARFF Weka file (see section 6.1.1) is then bundled into the application itself by transferring the collected data straight into the *Assets* directory of the mobile application. That allows for "shipping" the application with the collected training dataset. The kNN classifier is then trained and serialised when the *onCreate* method of the *Application* is called. This method is only called once during the life of the application (Google, 2017d).

Accuracy

The ARFF Weka file (dataset) was tested externally on a PC to find the optimal kNN parameters such as the number of neighbours the classifier should consider before making a class prediction as well as whether to perform a cross-validation of the data. The testing was done using the PC version of WEKA [2] Machine Learning application. After performing thorough testing, the default parameters of the kNN classifier performed the best reaching accuracy of *83.7%* for dataset containing 763 data points. The performance for the trained model can be seen in table 6.1.

a	b	c	d	classified as
140	4	2	33	a = walking
57	112	0	10	b = running
0	1	218	0	c = static
5	1	11	169	d = cycling

Table 6.1: 4 classes Confusion Matrix

According to the confusion matrix in the table above, data points of class "running" has been incorrectly classified 57 times as class "walking" and 10 times as class "cycling". However, for the implementation purposes of the proposed application that performance of the model is still acceptable because even though the classifier has been designed to distinguish between 4 classes, only 2 classes are taken into account, namely "static" and "active" (including "walking", "running" and "cycling") on a programming level. Feature versions of the application may utilise the "active" classes to show a more detailed information to the user such as how much time the user has spent walking or running throughout the day.

To demonstrate the actual accuracy of the model taking account only those 2 classes, the original dataset has been modified so that only 2 classes are present - static and active. The confusion matrix after training the model(again the same kNN classifier with default parameters) can be seen in table 6.2. The actual accuracy of the classifier reaches 98.1% since only 14 data points have been incorrectly classified.

a	b	classified as
531	13	a = active
1	218	b = static

Table 6.2: 2 classes Confusion Matrix

[1] `http://www.st.com/en/mems-and-sensors/lis3dh.html`
[2] `http://www.cs.waikato.ac.nz/ml/weka/`

6.1.3 HAR System Implementation

The basic architecture of the HAR system integrated into the application has been informed by Labrador and Yejas (2013: 149). For example, classes such as *Point*, *TimeSeries*, *TimeWindow*, *FeatureSet* and *FeatureExtractor* form the core of the HAR system. Simplified diagram of the HAR system can be seen in figure 6.1. The HAR system is integrated in the *ActiveMinutesService* mentioned in section 4.3 so it can recognise and provide the recognised activities to *ActivityMonitor* class which will be discussed shortly. A brief discussion on the main responsibilities of the HAR system classes is presented bellow.

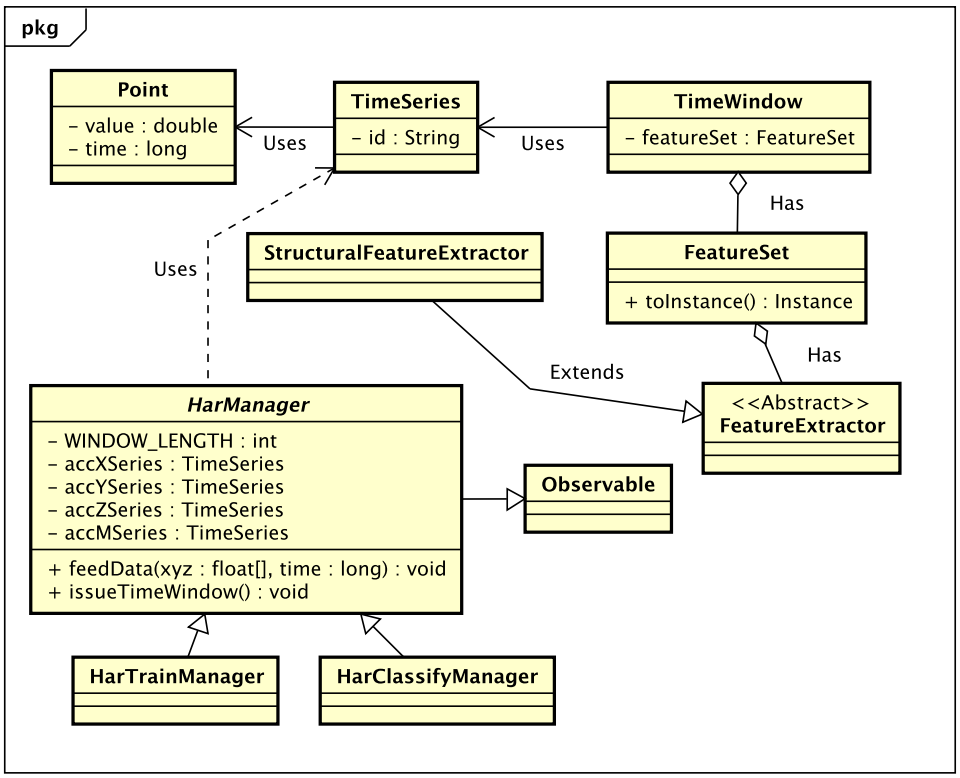


Figure 6.1: HAR system class diagram (simplified)

Class **Point** is responsible for storing the readings of the smartphone accelerometer sensor. Every instance of the class stores the accelerometer readings such as the amount of acceleration in X, Y and Z direction in a class variable *value* as well as a *time* class variable which is used to compare different instances of the class.

TimeSeries class extends *ArrayList* which is one of the widely used data structure in the Java language since it allows easy storing and manipulation of list of items (Oracle, 2017a). TimeSeries class stores a list of *Point* values labeled by an *id* value (see figure 6.1). For example, to store the accelerometer *X* direction acceleration values in a TimeSeries instance, its *id* could store value such as "accX".

TimeWindow class extends Hashtable. That allows for storing values accessed by key (Oracle, 2017b). For example, TimeWindow class provides means of storing *TimeSeries* objects and accessing them via key. The key value used to associate every TimeSeries object is the same as the *id* value of the TimeSeries object itself.

StructuralFeatureExtractor is a class that extends the abstract class FeatureExtractor. The reason behind this abstraction is that in the future more extractors can be added such as *StatisticalFeatureExtractor* which could extract statistical data features from a dataset such as mean or standard deviation. StructuralFeatureExtractor is responsible for extracting the structural data features such as the first five coefficients of the FFT algorithm for all of the accelerometer axis and their magnitude as described in section 2.5.

FeatureSet class is responsible for storing the extracted data features of the *StructuralFea-*

tureExtractor. Similarly to TimeWindow, this class extends Hashtable as well. The key, in this case, is a combination between the id of the TimeSeries instance and the name of the feature such as "_fft". The value that can be accessed by this key is the value of the specific TimeSeries FFT coefficient. For example, if we want to store the FeatureSet (e.g. feature vector) of *TimeSeries* with id = "AccX" (e.g. the values of the accelerometer in X direction for a given time) the key to access the first FFT coefficient of this TimeSeries would like that *"AccX_fft1"*.

HarManager is a class that manages the raw data from the smart phone's accelerometer. The data is delivered in the class via its method *feedData*. This method accepts an array of float values *xyz* which are the readings of the accelerometer at a given time. The method feeds data into the appropriate TimeSeries (i.e. accXSeries). When the window length is reached (e.g. 3 seconds) the method *issueTimeWindow* is called. The purpose of this method is to extract data features from the raw data. The two direct subclasses of HarManager class - *HarTrainManager* and *HarClassifyManager* override the method so that it stores the labelled data points during the data collection process or classifies the unseen (or unlabelled) data points during the classification or monitoring process. For example, the *DataCollectionService* class discussed in section 4.3 uses an instance of the HarTrainManager class to collect data and train kNN classifier. On the other hand, the HarClassifyManager is used in the main service of the application called "ActiveMinutesService" (again discussed in section 4.3 to classify unseen instances periodically).

6.2 Self-Management System

This section discusses the implementation of self-management system. Its responsibilities include monitoring the user's activity levels as well as providing feedback upon reaching a specific threshold. A class diagram of the system can be seen in figure 6.2. The following subsections will discuss the main components of the self-management system.

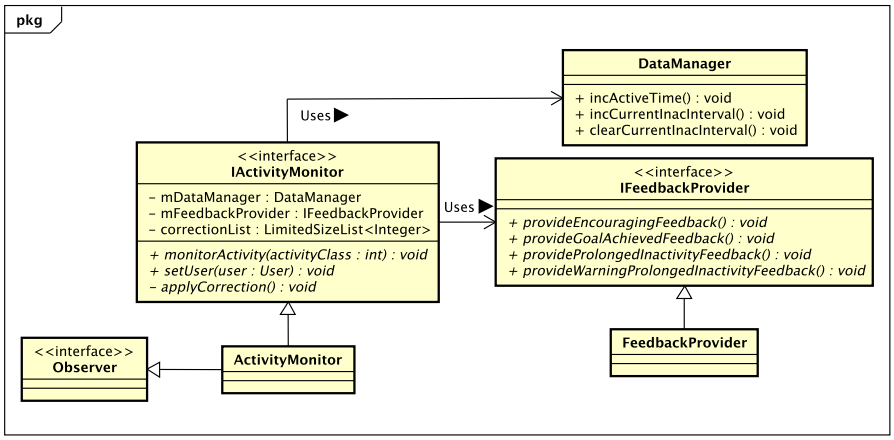


Figure 6.2: Self-Management System class diagram (simplified)

ActivityMonitor

As the name implies, this class monitors user's PA and ST levels. ActivityMonitor accesses the previously set daily physical activity and maximum continuous inactivity goals set by the user. The User object that this class needs is provided by a method called *setUser(User user)*. The details of how the user is set will be discussed in more detail in a subsequent section.

One of the most important methods of the class is called **monitorActivity(activityClass: int)**. This method is called by the **HarManager** mentioned above as soon as an activity type is recognised (every 3 seconds). It passes the recognised activity to the ActivityMonitor via the method mentioned above. ActivityMonitor then is able to react on the activity type. For example, if the activity type is of type *Static*, ActivityMonitor calls the DataManager's *incCurrentInacInterval()* method which increments the current inactivity interval in the database.

Frequent positions changes

One problem that had to be solved during the implementation of this class was dealing with the frequent changes of the device position. For example, if the user pulls out their phone (e.g. to check for missed calls) and put it back in the pocket, these series of events are usually interpreted by the HarManager as activity class of type Active (e.g. running, walking or cycling). Without applying any kind of data filtering, ActivityMonitor will increment the user's active time incorrectly especially if the user is static at the time of the interaction with their phone. Consequently, that will lead to wrong activity data and potentially misleading the user.

After series of experiments it was found that by storing the last 3 recognised activities in a list (i.e. walking, running, walking) and comparing them with the "current" activity, these frequent device changes are dramatically avoided. That is because, in order for the activity to be classified as *Active* all of the activities in the list must be of the same *Active* type (i.e. walking, running or cycling). Pragmatically, this is achieved by a method called *applyCorrection()*. The interested reader can find the implementation of **ActivityMonitor** class in Appendix K.

DataManager

As can be seen in figure 6.2 a reference of this class is used in *ActivityMonitor*. The role of **DataManager** is to provide methods of manipulating the underlining database so that it is relevant to the needs of *ActivityMonitor*. For example, *ActivityMonitor* receives a recofnised activity of type *Active* (i.e. walking), *ActivityMonitor* calls *incActiveTime()* and *clearCurrentInacInterval()* consequently. The former method increments the user's active time and the latter clears the current inactivity interval to 0.

FeedbackProvider

This class is responsible for providing information to the user in the form of Android notifications[3]. For example, when *ActivityMonitor* detects that the user has reached their daily maximum continious interval (set sedentary goal) it would call *provideProlongedInactivityFeedback()* method of the *FeedbackProvider*.

Notifying the user of past events specially associated with user's inactivity is not very helpful from a self-management perspective. Thus, the class has been designed to include two methods that are called at 80% achieved goal progress, namely *provideEncouragingFeedback()* and *provideWarningProlongedInactivityFeedback()*. For example, if the ST goal, i.e., maximum continuous inactivity (MCI) is set to be 60 minutes per day. The user will receive a notification at 48 minutes of continuous inactivity giving the user a warning that they are approaching their MCI.

6.3 "Active Minutes" application

This section will discuss the main and most interesting building blocks of the proposed software. First, the realisation of the database design will be addressed. Next, the *ActiveMinutes Service* functionality will be discussed followed by the Application UI.

6.3.1 Realm Database

As it was mentioned in section 4.3.1, a library called Realm (2017) is utilised to perform all the data management responsibilities. This third-party software was carefully chosen based on several factors. It offers Fast queries and Safe threading (accessing the data from many threads without conflicts). What is more, it saves a lot of time compared to implementing a database from scratch. The following sections will address how the database was implemented using the above-mentioned library. A simplified version of the Realm database can be seen in figure 6.3.

RealmObject

The database design discussed in 4.4 is implemented using Java classes. The **Realm** database library allows to represent a typical database table as a Java class with the difference that it

[3] `https://developer.android.com/reference/android/app/Notification.html`

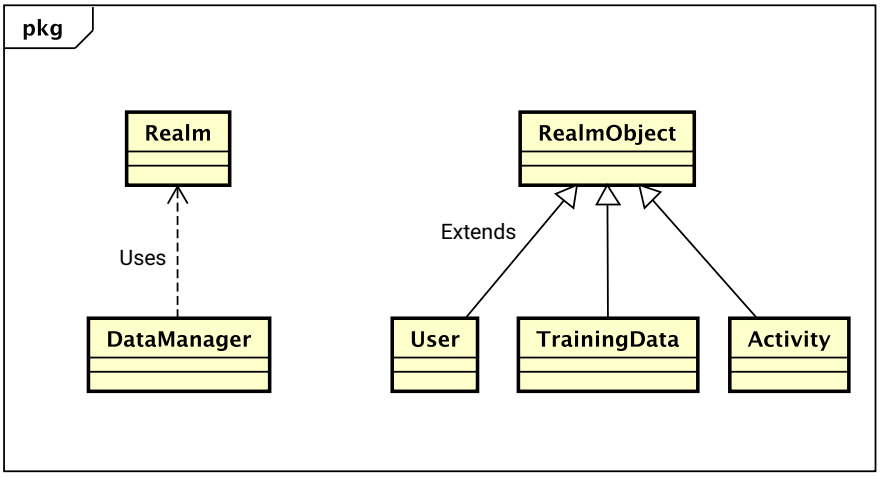


Figure 6.3: Realm database class diagram (simplified)

must extends *RealmObject*. That allows the class to be automatically "converted" into a database table. Also, it becomes available for access via a *Realm* object anywhere in the system. For example, in listing 6.1 the method *getTrainingData()* queries the *TrainingData* table for records matching a specific *user_id* and returns the result as a WEKA *Instances* object (i.e., dataset). The *TrainingData* is accessed in the *where* clause.

```
public Instances getTrainingData(int userId) {
    RealmResults<TrainingData> realmResults = realm
    .where(TrainingData.class)
    .equalTo(USER_ID, userId)
    .findAll();

    return getWekaInstanceFrom(realmResults);
}
```

Listing 6.1: Accessing database tables

DataManager

This class encapsulates a *Realm* object and provides methods methods to manipulate the realm database. The *DataManager* class is split into 3 main data managers in the system to help maintain the code, *AuthDataManager*, *ActivityDataManager* and *TrainingDataManager*. **AuthDataManager** is responsible for the authentication as well as the registration of users in the database. For example, the Login and Sign up screens mentioned in section 4.3 use an instance of *AuthDataManager* in order to *login* or *register* users. **ActivityDataManager**, on the other hand, is a class that manages *Activity* records data (see section 4.4 for details about Activity database table). A reference of this class is used in *Today* screen as well as in *History* screen. For example, in order to retrieve the the list of activity records needed to populate the *Daily* view list in the *History* screen, a *getAllActivitiesSortedByDate()* method of *ActivityDataManager* is called. Last but not least, **TrainingDataManager**. This data manager is involved in the data collection process as the name implies. It provides methods for storage, deletion and retrieval of the collected data during the data collection process. For example, in order to save the collected training data *saveTrainingData()* method of the class is called.

6.3.2 ActiveMinutes Service

This service is the most important class in the mobile application. It is responsible for the 2 out of 3 key behaviour change components, namely Monitoring and Feedback (refer to section 2.3). This

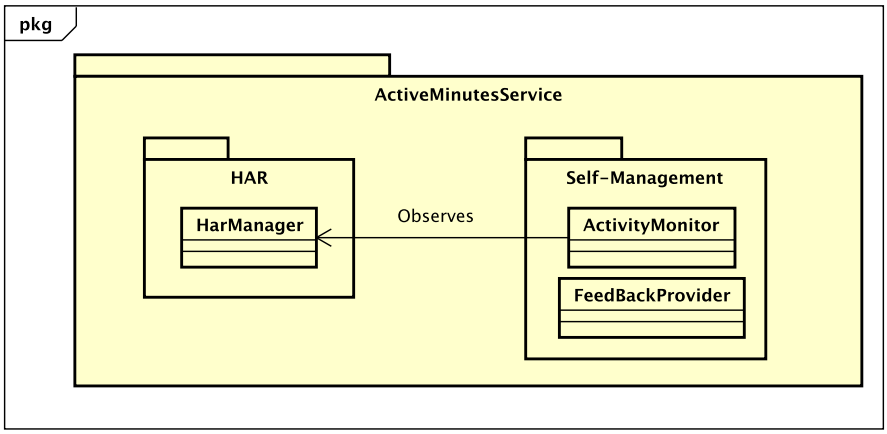


Figure 6.4: ActiveMinutesService class diagram

class accomplishes that by utilising the two systems discussed earlier in this chapter - *HAR* and *Self-Management*. A logical view of how this is achieved can be seen in figure 6.4.

Implementing *HarManager* as **Observable** and *ActivityMonitor* as **Observer** (refer back to figure 6.1 and 6.2) allows to connect the two systems by the *Observer* pattern so they can work together and exchange information in the ActiveMinutes service. For example, when the service is started (e.g., when a user logs in) it calls a method called *onCreate()* which consequently adds *ActivityMonitor* to the list of observers of *HarManager*. That allows *ActivityMonitor* to "listen" for updates from *HarManager*. Every time *HarManager* recognises an activity it notifies *ActivityMonitor* passing over the recognised activity. ActivityMonitor is then able to react on the recognised activity as discussed in section 6.2. The interested reader can refer to Appendix L for a simplified version of *ActiveMinutesService* class.

6.3.3 Application UI

The application UI allows the user to access the collected data from the *ActiveMinutesService*. Its main purpose is to visualise the data so the user can have a better understanding of their performance over time. For example, the user can see a history of the past days and check which days they have achieved their set goals. In addition, the application UI provides the user to set PA and ST goals. In other words, the application UI, more specifically the *Settings* screen, plays the important role of the Goal-Setting component (see section 2.3). This section discusses will detail the main functions of the mobile application screens. First, the *Today* screen will be addressed, followed by the *History* and *Settings* Screens. All of the mobile application screen designs can be found in Appendix C.

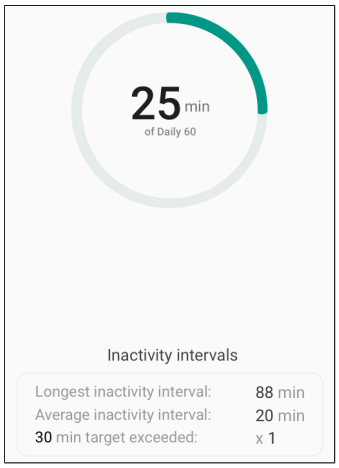


Figure 6.5: Today screen information

Today Screen

This screen is the main screen of the mobile application. It provides information such as the today's activity physical activity and sedentary time amounts. For example, *Today* screen provides shows the user their set PA goal and their progress (see figure 6.5).

Since the application focuses on sedentary time and not only on active time, it provides information such as *Inactivity intervals* as well. For example, information such as *Longest inactivity interval*, *Average inactivity interval* and how many times their target is exceeded provide the user with a clear view of how they are performing in terms of sedentary behaviour. As with PA, the *Inactivity intervals* displays the user ST goal as well *30* min.

History Screen

Allowing the user to access goal performance in the past, plays an important part in the self-management process. For example, if the user has not achieved most of the 60-minute PA goals for the previous week, they may decide to correct the goal to 30 minutes of daily PA activity instead. The History screen allows the user to clearly see what goals they are achieving.

As shown in the History screen snippet in figure 6.6, the user can easily see if they achieved a PA or ST (i.e. MCI) goal by looking at the colour of the specific list item for a given day. Every list item is composed of a *title* to show the date; two progress bars to show the progress of the PA and ST (i.e. MCI) goals and a pair of "progress/goal" text labels that show the progress of goals in numerically.

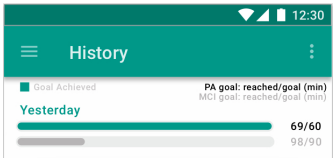


Figure 6.6: History screen list item snippet

The **legend** at the top of the screen helps the user make a connection between the progress bar colour and the "progress/goal" pair. For example, in the single list item shown in the figure at the right-hand side we can see that the PA goal is achieved since the current progress of the goal "69" is greater than the goal "60" minutes. Thus the colour of this list item's PA progress bar is set to "GREEN" (i.e., "Goal Achieved"). On the other hand, the progress bar's colour showing the progress/goal pair for the MCI goal is set to "GREY" since the ST goal has not been achieved (i.e. the progress "98" is greater than the user set MCI "90").

Settings Screen

The settings screen enables the user to change certain settings as well as plays the role of the Goal-Setting component discussed in section 3.1.1. Small snippet of the UI design of this screen can be seen in figure 6.7

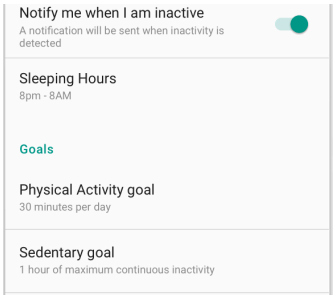


Figure 6.7: Settings screen snippet

As mentioned in the specifications chapter (see section 3.1.2), the user is able to set a time window where no monitoring occurs - *Sleeping Hours*. This interval normally starts at *8pm* till *8am*.

This screen offers the user the option to switch off ST notifications when they do not want to receive any (e.g. during a business meeting) by tapping on the switch UI component (see figure 6.7). In addition, the *Settings* screen offers the user to change their PA and ST goals. The default values for the former and latter are 30 and 60 minutes respectively. The complete UI design of this screen can be found in Appendix C.

Initial Setup Screens

When a new user registers with the application, an initial-setup screens are shown. This set of screens allow the user to set their *SleepingHours* time interval; *PA* and *MCI* goals upon the first launch of the application.

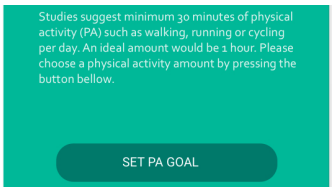


Figure 6.8: SET PA GOAL screen snippet

In addition, the screens provide an important information to the user informed by the the project research such as the recommended PA and ST guidelines per day. For example, the *SET PA* screen shown in figure 6.8 tells the user that the desired amount of PA activity is one hour. Consequently, the user can make a better decision regarding setting their daily PA goal. As part of the *initial-setup*, the application shows a disclaimer screen warning the user that they should not be using the application if they suffer from a disease that could be worsened, e.g. asthma. All of the screens can be seen in Appendix C.

Performance Meeting screen

To provide a better feedback, e.g. not just sending notifications when prolonged inactivity is detected, the system offers an additional feature. At the end of every week (e.g. Sunday) the user would receive a notification. When tapped, the *Performance Meeting* screen will be shown. It provides goal-performance information regarding user's *Activity* and *Inactivity*. For example, the screen shows how many times the user has achieved a PA goal and how many times the user exceeded their MCI target. Consequently, the user is able to see how well they perform and thus make an informed decision when changing their PA and ST goals. A snippet of the UI design of this screen can be seen in figure 6.9, the complete design can be seen in Appendix C.

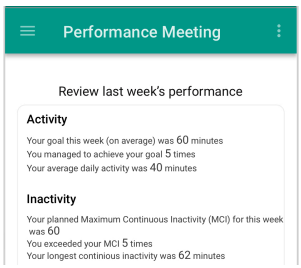


Figure 6.9: Performance Meeting screen snippet

Chapter 7

Testing and Evaluation

The purpose of this chapter is to evaluate how effective the developed mobile application is in respect to applying the behaviour change theory discussed in Chapter 2. To examine this, a series of tests were performed, namely: *Acceptance*, *Functional* and *Usability* testing. Furthermore, a survey was carried out with three university students in order to obtain general user opinion regarding varying aspects of the application. The tests performed on the system are going to be discussed first followed by a discussion on how the system compares with a current commercial product in the same field. Finality, the survey responses are examined.

7.1 Acceptance Testing

Acceptance testing is a key part of the software development process. It evaluates weather the software conforms with the client requirements. Consequently, it is determined if the end-product is capable of performing the specified functionality. The following sub-components of the system underwent Acceptance testing, namely: *Goal-Setting Component* (GSC), *Monitoring Component* (MC) and *Feedback Component* (FC).

7.1.1 Goal-Setting Component

The three specifications for this component were all implemented successfully. The list of the implemented requirements can be seen in Table 7.2.

- *"GSC shall provide list of recommended PA goals"* - is implemented by utilising an Android framework component called *Dialog* (Google, 2017). A Dialog window is presented to the user so they can choose their PA goal in minutes (e.g. 10, 20, 30, 60).
- *"GSC shall automatically set the recommended PA goal if the user does not explicitly do so"* is implemented in the *initial setup* set of screens discussed in section 6.3.3. If the user decides to skip the *initial-setup* screens (which offer options such as setting the PA, MCI and the *Sleeping Hours* time interval) default values are set. For example, the PA goal is set to 30 minutes, MCI (or the ST goal) is set to 60 minutes and the *Sleeping Hours* time interval is set to 8pm to 8am.
- *"GSC shall allow the user to set a Maximum Continuous Inactivity (MCI)..."* is successfully implemented using a dialog window. It allows the user to set their ST (or MCI) goal by selecting it from a list of preset values in the dialog. In addition to the *initial setup* screens the user is able to do that in the *Settings* screen.

7.1.2 Monitoring component

Four out of the five initially set features were fully implemented. The list of the implemented requirements can be seen in Table 7.4. First of those features - *HAR component shall recognise 4*

Specification	**Status**	**Notes**
GSC shall provide list of recommended PA goals	Implemented	
GSC shall automatically set the recommended PA goal if the user does not explicitly do that	Implemented	If the user does not set a goal during the initial-setup, the app sets the recommended daily activity levels
GSC shall allow the user to set a Maximum Continuous Inactivity (MCI) before a reminding notification is sent	Implemented	

Table 7.2: GSC acceptance tests

Specification	**Status**	**Notes**
MC component shall classify four physical activities	Implemented	The classifier is trained to recognise *walking*, *running*, *cycling* and *static*
MC shall the increment user's PA or ST value whenever physical activity or sedentary behaviour is recognised	Implemented	
The MC shall personalise HAR's classifier	Not implemented	Due to the time constraints of the project this feature was not implemented.
The system shall allow the user to group the collected data in "Daily" and "Weekly" representation	Implemented	This is implemented in the History screen of the application
The user shall be able to set the *Sleeping Hours Time Interval* (SHTI) during which no measurement of PA or ST should occur	Implemented	

Table 7.4: MC acceptance tests

activities(walking, running, cycling and static) was implemented by gathering activity data from fellow university students and training a classifier to recognise the above-mentioned activities.

As far as *MC shall increment user's PA or ST value whenever physical activity or sedentary behaviour is recognised* is concerned, this feature was implemented in the *ActivityMonitor* class (see section 6.2). This class is responsible for incrementing the activity and inactivity values depending on the recognised class by the classifier.

The system shall allow the user to group the collected data in "Daily" and "Weekly" representation. This feature was implemented in the *History* screen of *ActiveMinutes*. The UI of this screen includes *DAILY* and *WEEKLY* buttons allowing the user to switch between both representations of the collected data, accordingly.

The next requirement - *The user shall be able to set the Sleeping Hours time interval during which no measurement of PA or ST should occur* - was implemented in the *Settings* screen. The UI of this screen provides the user with the option to change their *Sleeping Hours* time interval by tapping on the appropriate menu item and choosing the desirable start and end times.

There were some features of the system that were not implemented since it was determined to be too complex given the project time constraints. One of those feature was *The MC shall personalise HAR's classifier*. This feature was discussed in section 4.3.1 and it has been designed as Android *Service*. It uses the gathered-with-time and filtered by the classifier's confidence level data (e.g. above 80% - to filter only the recognised activities with high confidence). The service

combines this data with the general one and produces a personalised model that is used for further classification. According to Arapakis et al. (2010: 376), that is expected to improve the classifier's accuracy.

7.1.3 Feedback Component

As far as the FC is concerned, a total of three features were successfully implemented (see Table 7.6. One of the initial requirements of the FC was not implement due to the project's time constrains. *FeedbackProvider* (see section 6.2) represents the FC implementation system-wise. The first requirement - *FC shall notify the user when they are being sedentary for 1 hour* allows the user to be notified for a prolonged sedentary time. This feature was implemented by utilising Android's *Notification* API. In addition, this API is used to satisfy another feature as well, namely *FC shall notify the user when a previously set goal is attained.* It allows the user to be notified when they achieve their daily goal (e.g. 30 minutes of PA).

Specification	**Status**	**Notes**
FC shall notify the user when they are being sedentary for 1 hour (or the value set by the user) as recommended by Swartz et al. (2011)	Implemented	
FC shall notify the user when a previously set goal is attained (e.g. being active for 30 minutes)	Implemented	
FC shall not send notifications during the "Sleeping Hours" time interval	Implemented	
FC shall send a notification to the user every Sunday to show goal-performance feedback for the past week	Not Implemented	Although good to have, due to time constraints this feature was not implemented.

Table 7.6: FC acceptance tests

Another feature implemented feature - *FC shall not send notifications during the user set "Sleeping Hours" time interval* makes sure that the system stops monitoring user's PA and ST levels when *Sleeping Hours* time interval is entered.

Last but not least, *FC shall send a notification to the user every Sunday to show goal-performance feedback for the past week* - a feature (or rather a screen) which provides more information to the user regarding their goal-performance in comparison the only receiving notifications. Although "good-to-have", this feature was not implemented due to the project's time constrains rather than lack of technical capability.

7.2 Functional Testing

This is quality assurance (QA) testing that aims to verify that the system is providing the same output as required by the end user. Each test has an input, expected result, actual result and states whether the system passes the test or not.

Every feature of the system was tested to ensure that it works and satisfies the functional requirements of the system. For example, test cases verified that if the navigation of the system works as expected, e.g. when the user presses the *History* menu item of the navigational drawer, that takes the user to this screen. Also, self-management was tested as well, e.g. the goal-setting, monitoring and feedback components. For example, to test the monitoring component of the system the user (one of the project participants) had to perform 5 minutes of PA or SB. The result

was compared with the *Actual result* (if the system measures successfully the above amount). A snippet of the monitoring component functional testing can be seen in Table 7.8.

Step No	Test Cases	Expected result	Actual result	Status
1	The user starts the mobile application and performs 5 minutes of **walking**.	The system should recognise that the user is performing PA and increments *Active* time.	Works as expected e.g., updates the UI by adding 5 minutes to the PA progress bar in *Today* & *History* screens.	Pass
2	The user starts the mobile application and performs 5 minutes of **running**.	The system should recognise that the user is performing PA and increments *Active* time.	Works as expected e.g., updates the UI by adding 5 minutes to the PA progress bar in *Today* & *History* screens.	Pass

Table 7.8: Monitoring activity functional test snippet

A total of 30 test cases in different categories were created to ensure that every implemented feature of the system is tested. The system successfully passed all of the tests. The device used during the tests was *Google Pixel* [1]. All test cases can be seen in Appendix G.

7.3 Usability Testing

In addition to acceptance and functional testing, usability testing has been performed on the system to determine if it is easy to use. The usability test consisting of 16 tasks that the real user (i.e., one of the project participants) performed on the system. Two of the tests can be seen in Table 7.10. The full list of the tasks can be seen in Appendix H.

Task No	Task	Comments	Status
1	Open the application and create an account.	The user was able to understand what was required and complete the process by accessing the Sign-up screen, entering their credentials and selecting the *SIGN UP* button.	Pass
2	Navigate to the *Today* screen and describe what you are seeing.	The user was able to explain correctly the purpose of the screen i.e., the screen shows the current activity and inactivity goals as well as their progress	Pass

Table 7.10: Usability test snippet

The results from the test indicate that the user finds the mobile application easy to use. For example the user was able to understand how to perform all of tasks part of the test. However, when asked to explain the functionality *History* screen, the user spend more time trying to make the connection between the top PA and MCI labels and the two progress bars. This confusion can be easily fixed by adding a text label above each of the progress bars to indicated which is one is displaying PA and MCI.

All of the usability tests indicate that the mobile application is fairly easy to understand and use. The user was able to complete all tests successfully. It is worth mentioning that the user was familiar with the Android platform prior to the test. That means that standard icons such as the

[1] `https://madeby.google.com/intl/en_uk/phone/`

icon used to indicate the presents of a navigational drawer was already familiar to the user which made the task of accessing the different screens of the application a lot easier. Future usability tests should include a wider range of users.

7.4 Comparing the system with commercial solutions

The system has been compared with a commercial mobile application currently available in the market, namely *Human*[2] that offers similar functionality. The two systems have been compared regarding battery consumption, classifier technology used and the main features each of the systems offers. To do the comparison, the two applications have been installed on *Google Pixel* device and evaluated for a full day. The following subsections will discuss the evaluation findings.

7.4.1 Battery consumption

The battery consumption of both application has been evaluated by using an Android an android application called *AccuBattery*[3]. The results show that *ActiveMinutes* uses a considerable amount of battery in comparison to *Human*. According to the *AccuBattery* measurements, *ActiveMinutes* consumed about 150 mAh for the duration of the one day evaluation. That is a lot more then what *Human* consumed - 34 mAh.

7.4.2 Classifier technology

As it was discussed in section 4.2.4, *ActiveMinutes* trains a classifier online whereas *Human* uses *Google Fit* to perform the monitoring of the user offline. Building a classifier online introduces several advantages. Firstly, *ActiveMinutes*'s classifier can be personalised since it is built online. Secondly, the application can work completely offline, meaning that *ActiveMinutes* does not require an Internet connection to function.

Having said that, *ActiveMinutes* has several disadvantages. Since *Human* uses *Google Fit* to perform the monitoring, it uses a combination of sensors such as GPS to perform HAR. That means that the system is less likely to recognise involuntary movements of the device as PA. For example, if the user is sitting on a couch and moving their smartphone up and down, the system would not recognise that as PA. Although, *ActiveMinutes* keeps track of the last three activities to minimise the classification of these involuntary movements as PA it is easy to "fool" the system by going through the above-mentioned example.

7.4.3 Features

Feature-wise, *ActiveMinutes* offers a wider range of goals the user can set in comparison to what *Human* offers. For example, 10, 20, 30, 60, 120 and 30, 60 and 90 minutes are the possible values the user can choose from when setting PA and SB goals, respectively. For comparison, *Human* offers only three values for both PA and SB goals: 30, 60 an 120 (see figure 7.1). By offering a wider range of goal values the user can make a smoother transition from doing 10 minutes of physical activity per day to doing up to 2 hours. In addition to providing a winder range of goal values, *ActiveMinutes* offers the user the option to change their *SleepingHours* interval ("Inactivity nudge" in *Human*) whereas this value in *Human* is pre-set i.e. 9AM - 7PM.

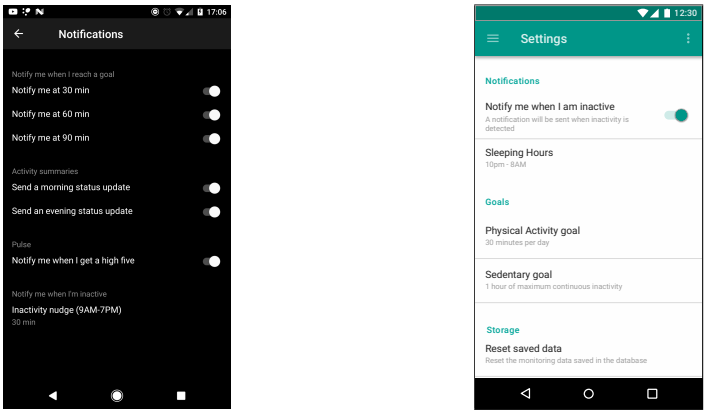
Having said that, *Human* offers several features which are not present in *ActiveMinutes*. For example, the user can compare their goal-performance with the performance of nearby people who are using the app as well. Also, *Human* shows a breakdown of the PA. For example, the user can see how much time they spend walking or cycling.

7.5 Survey results

In order to test *ActiveMinutes*'s performance in a real-life scenario, the mobile application was installed on a total three university students mobile phones. It was used for an entire day before the participants had to share their feedback via a survey. A survey contained a total of 9 questions, a snippet of the survey can be seen in Table 7.12. Each survey question and summary response are going to be discussed next. All questions can be seen in Appendix I.

[2] `http://human.co`

[3] `https://play.google.com/store/apps/details?id=com.digibites.accubattery&hl=en_GB`



(a) *Human* (b) *ActiveMinutes*

Figure 7.1: Comparing the system with other solutions

No	**Question**	**Response summary**
1	Were you more active then usual (e.g. walking, running) as a result of using ActiveMinutes?	Yes (33.33%) No (33.33%) Not sure (33.33%)
2	When notified for prolonged inactivity (e.g. 30 minutes of inactivity), did you try to do at least 5 minutes of physical activity?	Yes (66.67%) No (0%) Sometimes (33.33%)
3	Do you think the application was accurate when measuring your activity levels?	Yes (66.67%) No (0%) Somewhat (33.33%)
4	How easy to understand was the user interface of the application? (0 being very difficult and 100 being very easy)	Response 1 - 100 Response 2 - 80 Response 3 - 75 **Average - 85**

Table 7.12: User survey extract

- *What is your age?* According to the survey, all of the participant's age was between 18 and 24 years. Future evaluations of the system may consider including different age groups to find out how the system works for them.
- *Were you more active then usual (e.g. walking, running) as a result of using ActiveMinutes?* In general, the mobile application had a positive impact on two of the project participants. However, the third participant did not see any improvement in their level of PA.
- *When notified for prolonged inactivity (e.g. 30 minutes of inactivity), did you try to do at least 5 minutes of physical activity?* Overall, the mobile application had a positive effect on the project participants. Two people responded "Yes" and only one responded "Sometimes".
- *Does seeing past days goal-performance (e.g. in the History screen) motivate you to achieve a goal?* All of the survey participants answered "Yes" to this question.
- *Does the visual feedback (i.e. the green progress bar) encourage you to achieve your goal?* All of the survey participants answered "Yes" to this question.
- *Do you think the application was accurate when measuring your activity levels?* Two of the three survey participants responded "Yes" to this question where as the other participant answered "Somewhat".

- *How easy to understand was the user interface of the application (0 being very difficult 100 being very easy)?* All of the answers rated the application UI above 75 on this scale - the average of all responses is 85.
- *Was the mobile application battery-friendly?* Two of the three survey participants responded "Yes" to this question and only one answered "Somewhat".
- *Do you have any suggestions regarding the mobile application in general?* The survey participants gave the following responses to this question:
  1. "Great application, I like the feature that notifies you when you have been inactive for a certain amount of time. The application could maybe have a feature to alert you when you should go to bed to achieve the best amount of sleep. For future development, the application could maybe sync with a watch"
  2. "Slightly wider ranges in terms of sleep time and active and inactive minutes and perhaps a more obvious "Key" informing the users as to what each bar in the history means. Otherwise really easy to use."
  3. "The application could be improved in the future by optimising it to be a bit more battery-friendly."

Overall, the mobile system had a positive effect on the survey participants. They were more active during the day and did try to interrupt prolonged inactivity periods by doing physical activity. Also, they were motivated by their past goal-performance as well as seeing visual feedback in the *History* screen. The majority of the participants think that the system was accurate and that it had an easy to understand UI in addition to having a moderate battery consumption.

Chapter 8

Project Considerations

This chapter addresses the relevant professional, social, ethical, security and legal issues taken into consideration throughout the project development. As a student studying Computing for Graphics and Animations – a course that is accredited by the British Computer Society (BCS) I comply with the BCS Code of Conduct (BCS, 2017). The following sections will discuss how this is achieved.

8.1 Professional

According to point 4 (d) of the BCS code of conduct an individual should "act with integrity and respect in your professional relationships with all members of BCS. . . ". Thus, I acknowledge the work of all external software authors used in this work (see Appendix J).

8.2 Social

Point 3 (b) is especially important for the nature of the project. It states that an individual should "avoid any situation that may give rise to a conflict of interest...". The proposed application avoids promoting any body image stereotypes by focusing more on encouraging the users to be more active rather than on their body characteristics such as weight and height or BMI (Body Mass Index). In addition, the system only shows the recommended levels of physical activity and will not force the users of the application to follow those strictly (some jobs do require seating for extended intervals of time – e.g. drivers). That has been achieved by proving the option to in the *Settings* screen to disable inactivity notifications when necessary.

8.3 Ethical

Since the development of the project required data collection (i.e. recording accelerometer data) involving different people, each participant has been given a consent form to sign. The form ensures that the project participants know the purpose of the project and more importantly the use of their data. The consent forms can be seen in Appendix N. Furthermore, the data collected during the project development such as the names of the project participants and their accelerometer sensor data are used only for the development purposes of the project.

8.4 Security

As the author if this work I intend to respect and comply with point 1 (a) of the conduct that states "you shall have due regard for public health, privacy, security...". Thus, the developed system implements an authorisation component to prevent other people accessing personal data such as total minutes of activity/inactivity collected daily. Furthermore, any relevant or valuable user data is anonymised to protect individual's identity. For example, the system does not use nor asks the user for their name in order to operate i.e. user is only required to create *username* which does not need to be the same as their actual name.

8.5 Legal

To attempt preventing physical harm to the user, the system shows a disclaimer (warning) dialog message to inform the user that they should be in a good physical condition i.e. not suffering from any diseases that could be worsened by using the mobile application such as asthma (see figure C.6). In addition, I intend to fully comply with the terms of conditions of any third party software that I intend to use. As far as the project participant's data is concerned, I intend to keep the data only for the purposes of this project, and not use it for any commercial purpose.

Chapter 9

Concluding Remarks

To sum up, the project initially set seven objectives which help to achieve the high-level project aim. That is to develop a system that allows the user to self-manage SB and PA and provide goal-performance feedback in the form of notifications. The purpose of this chapter is to present the main findings discovered during the project development. Furthermore, a summary of the status of the project objectives will be presented followed by a discussion of the potential future improvements of the system. Finally, a personal reflection regarding the project experience will be discussed.

9.1 Objectives achieved

The project successfully achieved the initially set high-level aim and produced a system that applies self-management for both physical states, namely *Active* and *Static* and it was published on the application store. The system successfully applies behaviour change via three stages - goal-setting, monitoring and providing user feedback. The aim of the project was achieved by attaining the following project objectives:

- *Research the effect of SB as well as investigate behaviour change approaches*
  In order to do this, I have investigated the current literature regarding the behaviour change methods and moderators (i.e. what affects behaviour change, see section 2.3.2).

- *Investigate and gather information about HAR on wearable devices*
  This was done by analysing current and past HAR systems. For example, the literature review suggested that the optimal algorithm to train a classifier on a mobile system is k-Nearest-Neighbour. Furthermore, the optimal window size was found to be 3-second window length and the accelerometer sampling frequency 115 Hz (see section 6.1.1).

- *Develop a fully working HAR system*
  This objective has been achieved by implementing the research findings from the above-mentioned objective and developing a software component (i.e. *HarManager*) integrated as part of *ActiveMinutes* that is responsible for all of the HAR related stages such as data reprocessing & windowing; feature extraction & selection and classification of the user activity (see section 6.1.3).

- *Incorporate the above within an SB self-management system*
  When the HAR system was implemented, it was paired with a self-management system (see section 6.2) that facilitates goal-setting, monitoring and feedback functionality.

- *Design and Implement a mobile application based on the project research findings*
  In order to satisfy this objective successfully, a software architecture design was devised in addition to the UI design of the mobile application.

- *Evaluate the system performance using user feedback and analysing the gathered data*
  A series of tests, namely *Acceptance*; *Functional* and *Usability* have been carried out to evaluate if the system complies with the client requirements, works as expected and is easy

to use. In addition, a user survey has been conducted to gather real user feedback regarding the functionality of the system in general.

- *Release app to application to the Application Store*
  The mobile application was released to the public in the form of a pre-release version (also known as BETA) so users who want to try the system can do so by downloading the mobile application from Google Play Store[1].

9.2 Future versions

Although *ActiveMinutes* implements the key requirements needed by a self-management system (e.g. goal-setting, monitoring and user feedback) there are features that can be added in future iterations of the system. One of the features that may be addressed is classifier personalisation. When implemented, it is expected that the system would improve its classification accuracy since the classifier would adapt to the user's specific motor patterns (i.e. walking). Furthermore, personalisation could be applied to the notification content. For example, the system may keep track of what messages have proven to be successfully (e.g. made the user more active) so that only the selected message content is shown to the user.

One of the features that were not implemented due to the time constraint of the project was social media sharing. Future versions of the application could incorporate such feature allowing for sharing content on social media platforms such as *Facebook* and *Twitter*. For example, the user may share their achieved PA or SB goal. As it was found in the literature review (see section 2.3.2) making a goal public increases the importance of the goal in one's eyes and that is expected to produce better goal-performance results.

The feedback gathered via the survey discussed in Chapter 7 helped to identify other future improvements to the system. For example, integrating a sync feature with a smartwatch platforms such as *Android Wear*[2] would enable the user to self-manage their activity levels even without a smartphone. That would be particularly useful because people who wear watches are more likely to use them most of the time which would lead to more accurate measurements. Furthermore, optimising the system in terms of battery consumption would be another future improvement. That could be done by experimenting with lower sampling rates of the accelerometer sensor. The user feedback gathered from the survey also suggests improving the *History* screen UI so that it is easier to understand i.e. the connection between the legend section at the top of the screen is confusing and it is not that obvious it is associated with the progress bars data representation. That could be improved by adding labels in each of the progress bars so it is clear which one shows the PA and which one shows the MCI progress.

9.3 Reflection

In general, I am satisfied with the effort I have put into the project research and development. I have worked to the best of my abilities to satisfy the (BSc) Honours Computing (Graphics and Animation) degree requirements. As far as the project time management is concerned, I have spent approximately two-thirds of my time on the design and research stage. It allowed me to gather all of the required background information in order to complete the software implementation of the project. Also, the system design ensured that the mobile application implements the specifications required by a typical self-management system and it is feasible to implement given the time constraint of the project.

This undertaking allowed me to apply the university knowledge gathered during my four years of study. Also, I was able to practice and improve my industry experience gained with xDesign. Going into the project, I was interested in evaluating how effective mobile technology is in regards to self-managing people's active and sedentary time. Now that the project is completed, I feel confident that mobile applications such as *ActiveMinutes* can be used to help people improve their lifestyle.

[1] `https://play.google.com/store/apps/details?id=georgikoemdzhiev.activeminutes`
[2] `https://www.android.com/intl/en_uk/wear/`

9.4 Conclusion

This project produced a system that implements self-management approach to reduce sedentary behaviour. The users of the system can self-manage PA and SB. That has been achieved by implementing a goal-setting component allowing the user to change their goals; a monitoring component which monitors the user's activity levels and a feedback component which provides goal-performance notifications to the user for both active and inactive time in the form of notifications. The evaluation of the system suggested that it works as intended and successfully applies behaviour change. The survey participants were positively affected by using the system, i.e. tried to reduce prolonged inactivity intervals and introduce physical activity breaks doing walking, cycling or running.

Bibliography

AliveCor. Alivecor. AliveCor [Online], 2017. URL https://www.alivecor.com/en/#. [Accessed: 15 January 2017].

Android Studio. Android studio the official ide for android. Android Studio [Online], 2017. URL https://developer.android.com/studio/index.html. [Accessed: 22 January 2017].

A. Anjum and M. U. Ilyas. Activity recognition using smartphone sensors. *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 914 – 919, 01 2013. doi: 10.1109/ccnc.2013.6488584.

Ioannis Arapakis, Konstantinos Athanasakos, and Joemon M. Jose. A comparison of general vs personalised affective models for the prediction of topical relevance. *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, 2010. doi: 10.1145/1835449.1835512. URL http://dl.acm.org/citation.cfm?id=1835512.

Atlassian. Kanban the agile coach. Atlassian [Online], 2017. URL https://www.atlassian.com/agile/kanban. [Accessed: 13 February 2017].

Media Anugerah Ayu, Siti Aisyah Ismail, Ahmad Faridi Abdul Matin, and Teddy Mantoro. A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognition. *Procedia Engineering*, 41:224–229, 2012. doi: 10.1016/j.proeng.2012.07.166. URL http://ac.els-cdn.com/S1877705812025520/1-s2.0-S1877705812025520-main.pdf?_tid=8b4f47de-bdf4-11e6-90ab-00000aab0f02&acdnat=1481277048_fb45631774a873b52a538c52155ef29e.

Sulaimon A. Bashir, Daniel C. Doolan, and Andrei Petrovski. The effect of window length on accuracy of smartphone-based activity recognition. *IAENG International Journal of Computer Science*, 43(1):126–136, 2016.

BCS. BCS [Online], 2017. URL http://www.bcs.org/category/6030.

Douglas Bell. *Software engineering for students: A programming approach*. Addison-Wesley Educational Publishers, New York, 4 edition, 03 2005. ISBN 9780321261274.

Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R. Schmidtke, and Michael Beigl. Actiserv: Activity recognition service for mobile phones. *International Symposium on Wearable Computers (ISWC) 2010*, 10 2010. doi: 10.1109/iswc.2010.5665868.

T Bhuvaneswari and S Prabaharan. International journal of computer science and mobile computing a survey on software development life cycle models. *IJCSMC*, 2(5):262–267, 2013. URL http://www.ijcsmc.com/docs/papers/May2013/V2I5201384.pdf.

Aviroop Biswas, Paul I Oh, Guy E Faulkner, Ravi R Bajaj, Michael A Silver, Marc S Mitchell, and David A Alter. Sedentary time and disease incidence, mortality, and hospitalization. *Annals of Internal Medicine*, 162(2):123–132, 01 2015. ISSN 0003-4819. doi: 10.7326/M14-1651. URL https://goo.gl/3bUxGf.

Adam Butler. Moderator mediator. Moderator mediator [Online], January 2017. URL http://www.uni.edu/butlera/courses/org/modmed/moderator_mediator.htm. [Accessed: 14 January 2017].

N. A. Capela, E. D. Lemaire, N. Baddour, M. Rudolf, N. Goljar, and H Burger. Evaluation of a smartphone human activity recognition application with able-bodied and stroke participants. *Journal of NeuroEngineering and Rehabilitation*, 13(5):1–10, 01 2016. ISSN 1743-0003. doi: 10.1186/s12984-016-0114-0.

CERB. The costs of inactivity in europe. *Cebr.com*, June 2015. URL `https://www.cebr.com/reports/the-costs-of-inactivity-in-europe/`.

Denis Campbell. Nhs to offer free devices and apps to help people manage illnesses. *The Guardian*, 06 2016. URL `https://www.theguardian.com/society/2016/jun/17/nhs-to-offer-free-devices-and-apps-to-help-people-manage-illnesses`.

Department of Health. Sedentary behaviour and obesity: Review of the current scientific evidence. Technical report, Department of Health, 2010. URL `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/213745/dh_128225.pdf`.

Department of Health. Start active, stay active a report on physical activity for health from the four home countries' chief medical officers. Technical report, Department of Health, 2011. URL `http://www.bhfactive.org.uk/userfiles/Documents/startactivestayactive.pdf`.

Endomondo. Endomondo features. Endomondo [Online], January 2017. URL `https://www.endomondo.com/features`. [Accessed: 14 January 2017].

FitnessKeeper. Track your runs, walks and more with your iphone or android phone. FitnessKeeper [Online], January 2017. URL `https://runkeeper.com/#where-to-start-module`. [Accessed: 14 January 2017].

Google. Dialogs. Google [Online], 2017. URL `https://developer.android.com/guide/topics/ui/dialogs.html`. [Accessed: 14 April 2017].

Google. Android design principles. Google [Online], 2017a. URL `https://developer.android.com/design/get-started/principles.html`. [Accessed: 5 February 2017].

Google. Activity. Google [Online], 2017b. URL `https://developer.android.com/reference/android/app/Activity.html`. [Accessed: 2 February 2017].

Google. Up and running with material design. Google [Online], 2017c. URL `https://developer.android.com/design/index.html`. [Accessed: 5 February 2017].

Google. Application. Google [Online], 2017d. URL `https://developer.android.com/reference/android/app/Application.html`. [Accessed: 17 March 2017].

Google. Fragments. Google [Online], 2017e. URL `https://developer.android.com/guide/components/fragments.html`. [Accessed: 2 February 2017].

Google. Sensormanager. Google [Online], 2017f. URL `https://developer.android.com/reference/android/hardware/SensorManager.html`. [Accessed: 18 February 2017].

Google. Services. Google [Online], 2017g. URL `https://developer.android.com/guide/components/services.html`. [Accessed: 29 January 2017].

Google. Motion sensors. Google [Online], no date. URL `https://developer.android.com/guide/topics/sensors/sensors_motion.html`. [Accessed: 31 January 2017].

Google Inc. Google fit - fitness tracking – android apps on google play. Google Inc. [Online], January 2017. URL `https://play.google.com/store/apps/details?id=com.google.android.apps.fitness`. [Accessed: 14 January 2017].

Google Trends. Google Trends [Online], January 2017. URL `https://www.google.com/trends/explore?date=all&q=fitness%20app&hl=en-UK`. [Accessed: 15 January 2017].

Harvard TH Chan School of Public Health. Measuring physical activity. Harvard TH Chan School of Public Health [Online], 09 2012. URL `https://www.hsph.harvard.edu/nutritionsource/mets-activity-table/`. [Accessed: 15 January 2017].

Zhen-Yu He and Lian-Wen Jin. Activity recognition from acceleration data using ar model representation and svm. *2008 International Conference on Machine Learning and Cybernetics*, pages 2245 – 2250, 07 2008. doi: 10.1109/icmlc.2008.4620779.

Human. All-day activity & calorie tracker. Human [Online], January 2017. URL `http://human.co/`. [Accessed: 14 January 2017].

Tzu-Ping Kao, Che-Wei Lin, and Jeen-Shing Wang. Development of a portable activity detector for daily activity recognition. *2009 IEEE International Symposium on Industrial Electronics*, pages 115–120, 07 2009. doi: 10.1109/isie.2009.5222001.

Miguel A Labrador and Oscar Lara D Yejas. *Human activity recognition: Using wearable sensors and smartphones*. Taylor & Francis, Boca Raton, 12 2013. ISBN 9781466588271.

Oscar D. Lara and Miguel A. Labrador. A mobile human activity recognition system. *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 38–39, 01 2012. doi: 10.1109/ccnc.2012.6181040.

Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013. ISSN 1553-877X. doi: 10.1109/surv.2012.110112.00192.

Óscar D. Lara, Alfredo J. Pérez, Miguel A. Labrador, and José D. Posada. Centinela: A human activity recognition system based on acceleration vital sign data. *Pervasive and Mobile Computing*, 8(5):717–729, 10 2012. doi: 10.1016/j.pmcj.2011.06.004. URL `http://www.sciencedirect.com.ezproxy.rgu.ac.uk/science/article/pii/S1574119211000794`.

Tuan Dinh Le and Chung Van Nguyen. Human activity recognition by smartphone. *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pages 219–224, 09 2015. doi: 10.1109/nics.2015.7302194.

Jin Lee, Jungsun Kim, and Hindawi Publishing Corporation. Energy-efficient real-time human activity recognition on smart mobile devices. *Mobile Information Systems*, 2016, 07 2016. ISSN 1574-017X. doi: 10.1155/2016/2316757. URL `https://www.hindawi.com/journals/misy/2016/2316757/`.

Edwin A. Locke and Gary P. Latham. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American Psychologist*, 57(9):705–717, 2002. doi: 10.1037//0003-066x.57.9.705. URL `http://www-2.rotman.utoronto.ca/facbios/file/09%20-%20Locke%20&%20Latham%202002%20AP.pdf`.

K. Michalak and H. Kwasnicka. Correlation-based feature selection strategy in neural classification. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 1, pages 741–746, Oct 2006. doi: 10.1109/ISDA.2006.128.

Sedentary Behaviour Research Networ. Letter to the editor: Standardized use of the terms "sedentary" and "sedentary behaviours". *Applied Physiology, Nutrition, and Metabolism*, 37(3):540–542, 06 2012. ISSN 1715-5312. doi: 10.1139/h2012-024.

NHS England. Health and high quality care for all, now and for future generations. NHS England [Online], January 2017a. URL `https://www.england.nhs.uk/2016/01/embracing-innovation/`. [Accessed: 15 January 2017].

NHS England. Health and high quality care for all, now and for future generations. NHS England [Online], January 2017b. URL `https://www.england.nhs.uk/ourwork/innovation/test-beds/#`. [Accessed: 15 January 2017].

Oracle. Arraylist (java platform se 7), 2017a. URL `https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html`.

Oracle. Hashtable (java platform se 7), 2017b. URL `https://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html`.

Caroline Parkinson. Hour's activity 'offsets sedentary day'. *BBC Health*, 07 2016. URL `http://www.bbc.co.uk/news/health-36895789`.

Pinky Paul and Thomas George. An effective approach for human activity recognition on smartphone. In *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 1–3. Institute of Electrical and Electronics Engineers (IEEE), 03 2015. ISBN 9781479918546. doi: 10.1109/icetech.2015.7275024.

Realm. Realm for android. Realm [Online], 09 2014. URL `https://realm.io/news/realm-for-android/`. [Accessed: 29 January 2017].

Realm. Realm [Online], 2017. URL `https://realm.io/products/realm-mobile-database/`. [Accessed: 19 February 2017].

Daniele Riboni and Claudio Bettini. Cosar: Hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 08 2010. ISSN 1617-4909. doi: 10.1007/s00779-010-0331-7.

Jean Roberts. Pervasive health management and health management utilizing pervasive technologies: Synergy and issues. *Pervasive health management and health management utilizing pervasive technologies: Synergy and issues*, 12(1):6–14, 2006. URL `http://www.jucs.org/jucs_12_1/pervasive_health_management_and/jucs_12_01_0006_0014_roberts.pdf`.

James P Sanders, Adam Loveday, Natalie Pearson, Charlotte Edwardson, Thomas Yates, Stuart JH Biddle, and Dale W Esliger. Devices for self-monitoring sedentary time or physical activity: A scoping review. *J Med Internet Res*, 18(5):e90, 2016. doi: 10.2196/jmir.5373. URL `https://www.jmir.org/article/viewFile/jmir_v18i5e90/2`.

Sadiq Sani, Nirmalie Wiratunga, Stewart Massie, and Kay Cooper. Selfback - activity recognition for self-management of low back pain. *Research and Development in Intelligent Systems XXXIII*, pages 281–294, 2016. doi: 10.1007/978-3-319-47175-4_21.

Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 01 2015. ISSN 1424-8220. doi: 10.3390/s150102059.

SL Shuger, VW Barry, X Sui, A McClain, GA Hand, S Wilcox, RA Meriwether, JW Hardin, and SN Blair. Electronic feedback in a diet- and physical activity-based lifestyle intervention for weight loss: A randomized controlled trial. *The international journal of behavioral nutrition and physical activity.*, 8(41):1–8, 05 2011. URL `https://www.ncbi.nlm.nih.gov/pubmed/21592351`.

Haroon Siddique. One hour of activity needed to offset harmful effects of sitting at a desk. *The Guardian*, 07 2016. URL `https://www.theguardian.com/lifeandstyle/2016/jul/27/health-risk-one-hour-activity-offset-eight-hours-sitting-desk`.

Ian Sommerville. *Software engineering*. Addison-Wesley Educational Publishers, Boston, 9 edition, 03 2010. ISBN 9780137035151.

V. J. Strecher, G. H. Seijts, G. J. Kok, G. P. Latham, R. Glasgow, B. DeVellis, R. M. Meertens, and D. W. Bulger. Goal setting as a strategy for health behavior change. *Health Education & Behavior*, 22(2):190–200, 07 1995. ISSN 1090-1981. doi: 10.1177/109019819502200207.

Ann M Swartz, Leah Squires, and Scott J Strath. Energy expenditure of interruptions to sedentary behavior. *International Journal of Behavioral Nutrition and Physical Activity*, 8(69), 2011. ISSN 1479-5868. doi: 10.1186/1479-5868-8-69.

Steven Swinford. Britain becomes 'sedentary nation' as people give up walking. *The Telegraph*, 07 2014. URL `http://www.telegraph.co.uk/finance/newsbysector/transport/10998536/Britain-becomes-sedentary-nation-as-people-give-up-walking.html`.

Artem Syromiatnikov and Danny Weyns. A journey through the land of model-view-design patterns. *2014 IEEE/IFIP Conference on Software Architecture*, pages 21 – 30, 04 2014. doi: 10.1109/wicsa.2014.13.

Emmanuel Munguia Tapia, Stephen S. Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. *2007 11th IEEE International Symposium on Wearable Computers*, 2007. doi: 10.1109/iswc.2007.4373774. URL `http://www.ccs.neu.edu/home/intille/papers-files/MunguiaTapiaIntilleETAL07.pdf`.

Cesar Torres-Huitzil and Andres Alvarez-Landero. Accelerometer-based human activity recognition in smartphones for healthcare services. In *Springer Series in Bio-/Neuroinformatics*, pages 147–169. Springer Nature, 2015. ISBN 9783319128160. doi: 10.1007/978-3-319-12817-7_7.

Cesar Torres-Huitzil and Marco Nuno-Maganda. Robust smartphone-based human activity recognition using a tri-axial accelerometer. In *2015 IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS)*, pages 1 – 4. Institute of Electrical and Electronics Engineers (IEEE), 02 2015. ISBN 9781479983322. doi: 10.1109/lascas.2015.7250435.

Oscar Torres-Reyna. Time series. Technical report, Princeton University, 2012. URL `https://www.princeton.edu/~otorres/TS101.pdf`.

Nick Townsend, Kremlin Wickramasinghe, and Julianne Williams. *Physical activity statistics 2015*. British Heart Foundation, United Kingdom, 01 2015. ISBN 9781899088270.

Chi Wang and Wei Zhang. Activity recognition based on smartphone and dual-tree complex wavelet transform. *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, pages 267 – 270, 12 2015. doi: 10.1109/iscid.2015.51.

Lisa Whitehead and Philippa Seaton. The effectiveness of self-management mobile phone and tablet apps in long-term condition management: A systematic review. *Journal of Medical Internet Research*, 18(5):97, 2016. doi: 10.2196/jmir.4883. URL `https://www.jmir.org/2016/5/e97/`.

Rhiannon Williams. Android roars back in strongest growth in two years, as ios shrinks. *The Telegraph*, 05 2016. URL `http://www.telegraph.co.uk/technology/2016/05/17/android-roars-back-in-strongest-growth-in-two-years-as-apple-shr/`.

SL Williams and DP French. What are the most effective intervention techniques for changing physical activity self-efficacy and physical activity behaviour–and are they the same? *Health education research.*, 26(2):308–22, 02 2011. URL `https://www.ncbi.nlm.nih.gov/pubmed/21321008`.

E. G. Wilmot, C. L. Edwardson, F. A. Achana, M. J. Davies, T. Gorely, L. J. Gray, K. Khunti, T. Yates, and S. J. H. Biddle. Sedentary time in adults and the association with diabetes, cardiovascular disease and death: Systematic review and meta-analysis. *Diabetologia*, 55(11): 2895–2905, 08 2012. ISSN 0012-186X. doi: 10.1007/s00125-012-2677-z.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008. ISSN 0219-1377. doi: 10.1007/s10115-007-0114-2.

Gwenllian Wynne-Jones, Jemma Cowen, Joanne L Jordan, Olalekan Uthman, Chris J Main, Nick Glozier, and Danielle van der Windt. Absence from work and return to work in people with back pain: a systematic review and meta-analysis. *Occupational and Environmental Medicine*, 2013. ISSN 1351-0711. doi: 10.1136/oemed-2013-101571. URL `http://oem.bmj.com/content/early/2013/11/01/oemed-2013-101571`.

Yang Zhang and Yanjing Luo. An architecture and implement model for model-view-presenter pattern. *2010 3rd International Conference on Computer Science and Information Technology*, 8:532 – 536, 07 2010. doi: 10.1109/iccsit.2010.5565090.

Appendix A

Use cases

Log in

The user opens the application. The system shows the log-in screen. The user types in their username and password. If the provided credentials are valid, the system shows the main screen of the application.

Sign up

The user opens the application and go to the Sign-up screen. The system shows the Sign-up screen and opening the virtual keyboard awaiting user input. The user enters their username and password. The system verifies that the entered information is correct and creates a new user account. The user is taken to the main screen of the application.

Set PA goal

The user opens the application. The system takes the user to the main screen. The user opens the navigation drawer and selects the *Settings* menu item. The system opens the Settings screen. The user selects the "Physical Activity goal" settings item. The system shows a dialog window with list of options. The user selects the desired option. The system accepts the user input and updates the system accordingly.

Change SB remind interval

The user opens the application. The system takes the user to the main screen. The user opens the navigation drawer and selects the *Settings* menu item. The system opens the Settings screen. The user selects the "*Sedentary goal*" settings item. The system shows a dialog window providing list of options. The user selects the remind interval and confirms their choice by pressing the "OK" button. The system accepts the user input and updates the system accordingly.

Change *"Sleeping Hours"*

The user opens the application. The system opens the main screen. The user opens the navigation drawer and selects the *Settings* menu item. The system opens the Settings screen. The user selects the "*Sleeping Hours*" settings item. The system shows a date picker dialog. The user selects the preferred time interval and confirms their choice by pressing the "OK" button. The system accepts the user input and updates the system accordingly.

Check goal progress

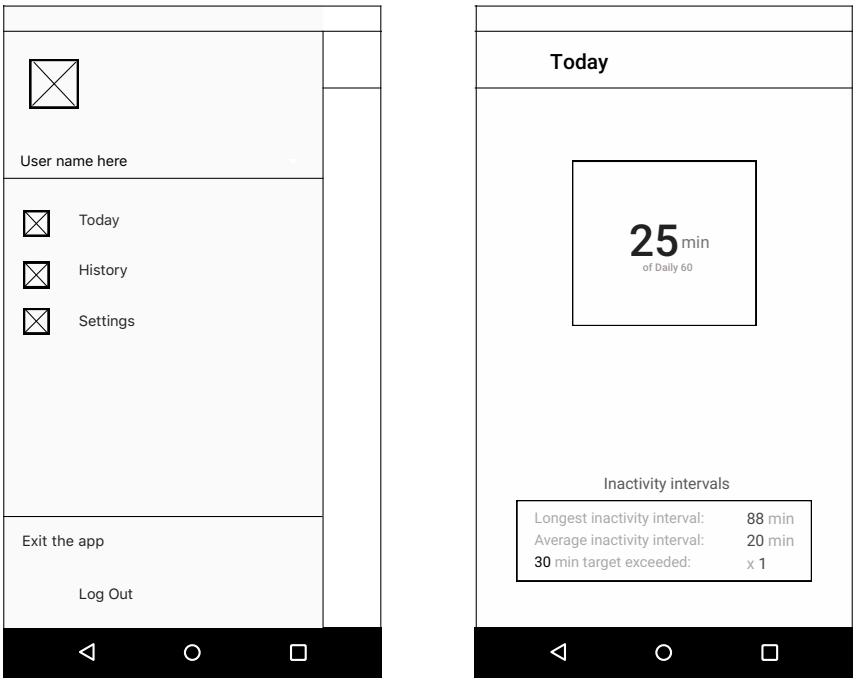
The user opens the application. The system opens the main screen (i.e. *Today* screen). The user sees the amount of PA in minutes as well as the previously set goal. The user closes the application.

Check activity history

The user opens the application. The system shows the main screen. The user opens the navigation drawer and selects the *"History"* settings item. The system opens the History screen. The user sees the history of activities sorted either *"Daily"* or *"Weekly"*. The user closes the application.

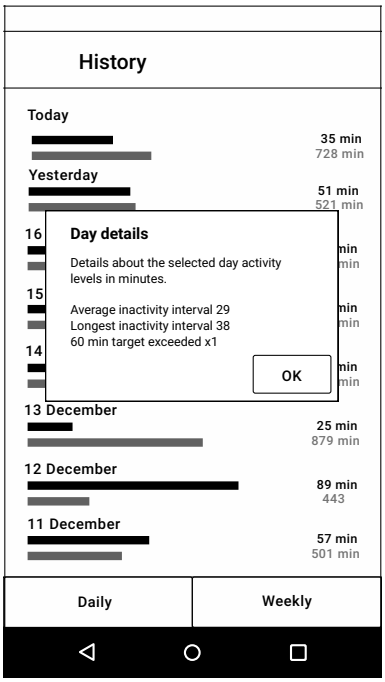
Appendix B

Screen Wireframes

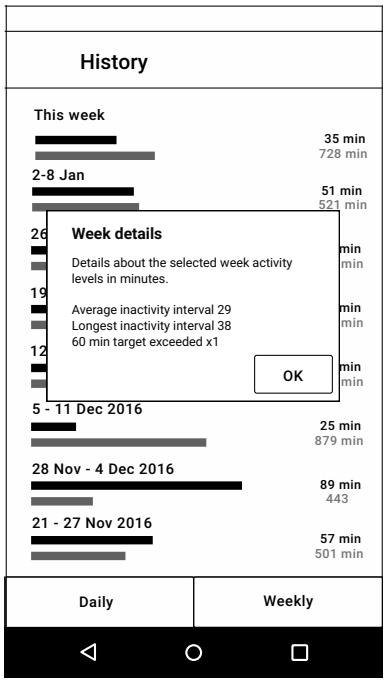


(a) Navigation Drawer opened (b) Main Screen

Figure B.1: Main Screen. Navigation drawer opened.



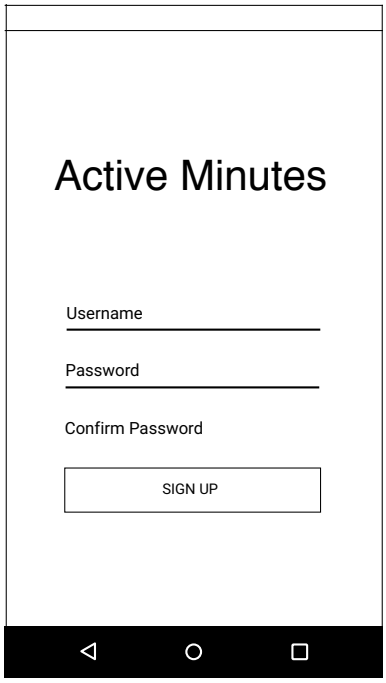
(a) History Screen, mode daily



(b) History Screen, mode weekly

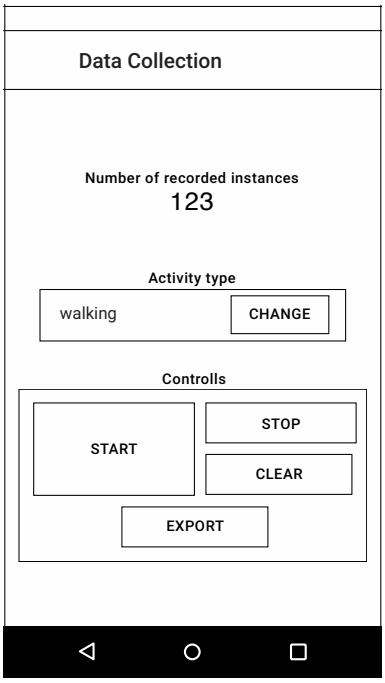
Figure B.2: History Screens

(a) Login Screen

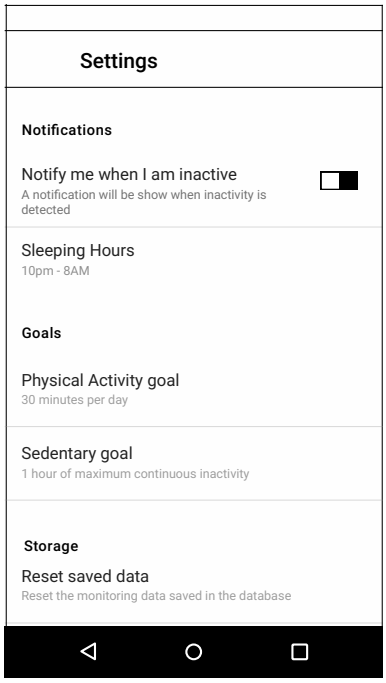


(b) Signup Screen

Figure B.3: Authentication Screens

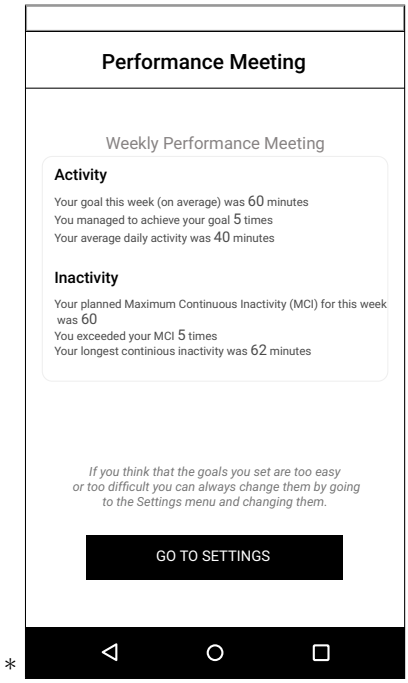


(a) Data Collection Screen



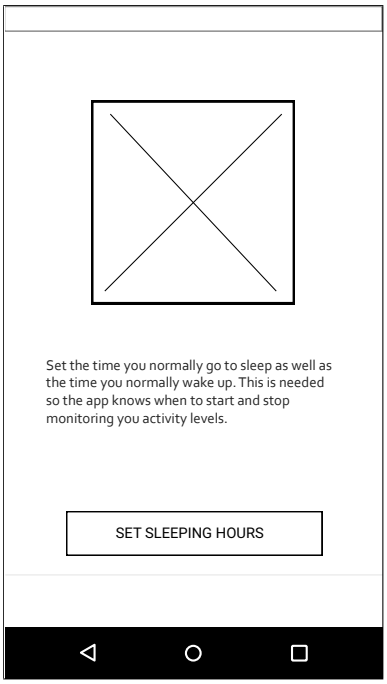
(b) Settings Screen

Figure B.4: Utility Screens

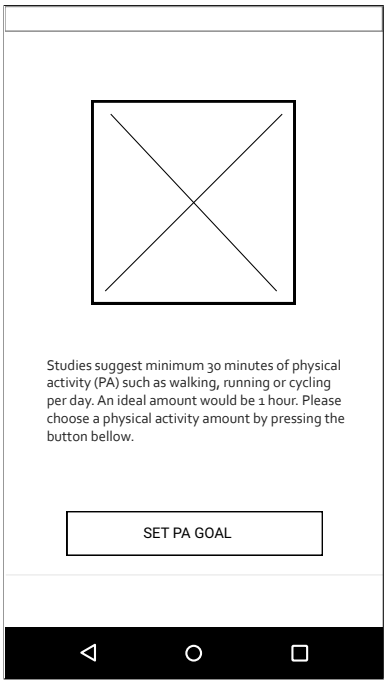


*

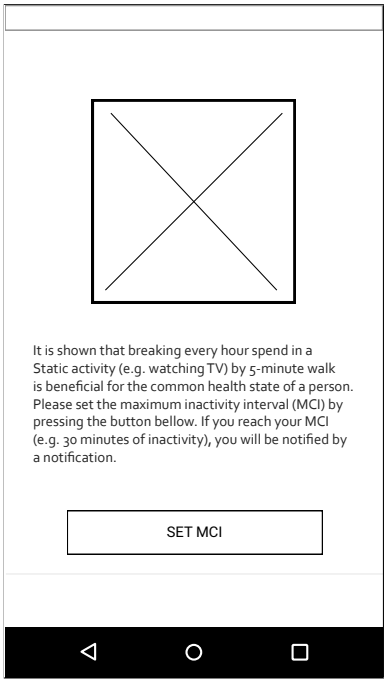
Figure B.5: Performance Meeting Screen



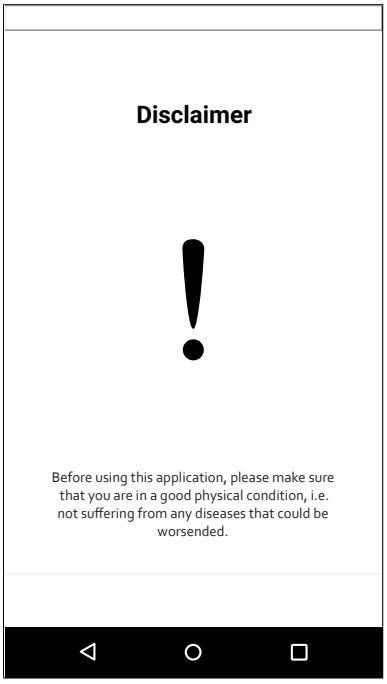
(a) Set Sleeping Hours Screen



(b) Set PA Screen



(c) Set MCI Screen

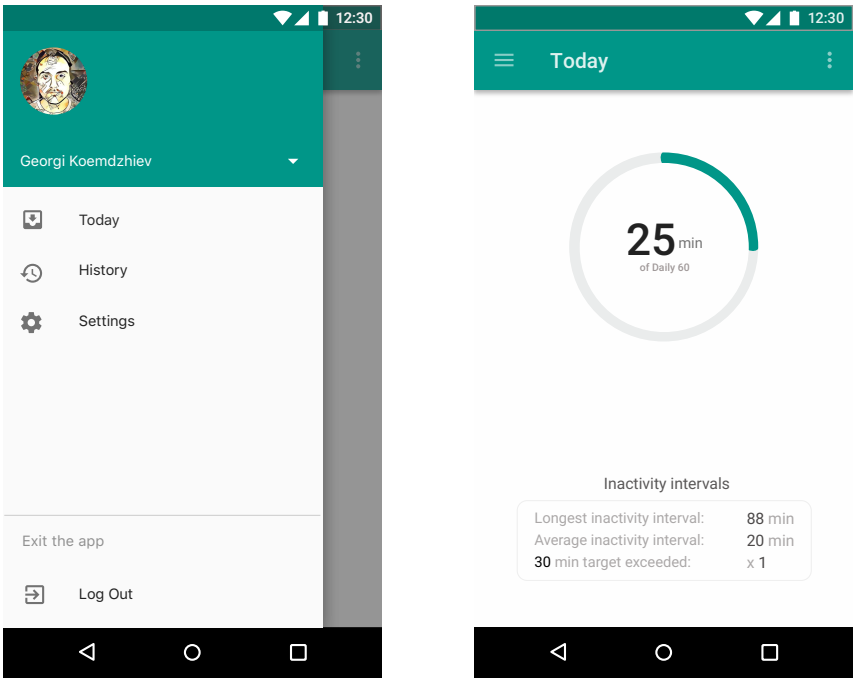


(d) Disclaimer Screen

Figure B.6: Initial setup screens

Appendix C

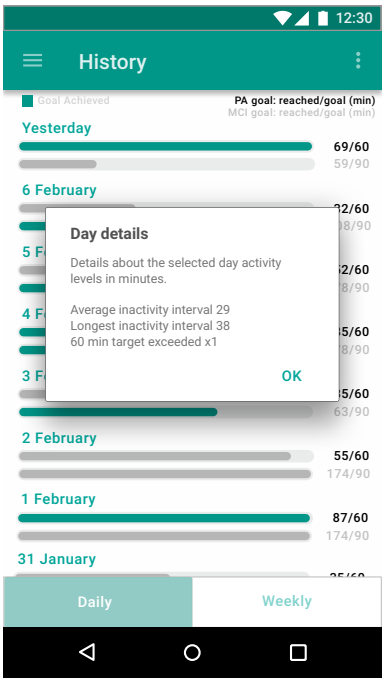
Screen Designs



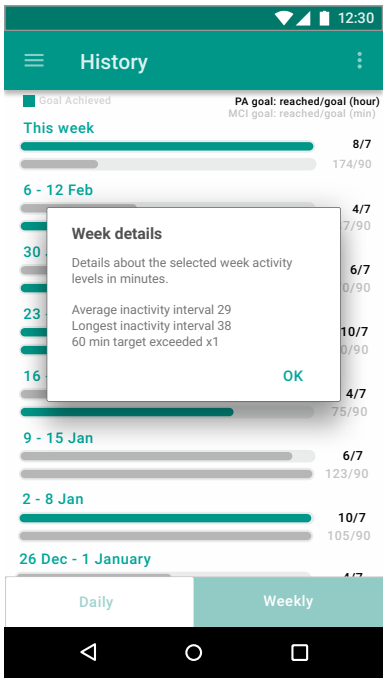
(a) Navigation Drawer opened

(b) Today Screen

Figure C.1: Today Screen. Navigation drawer opened

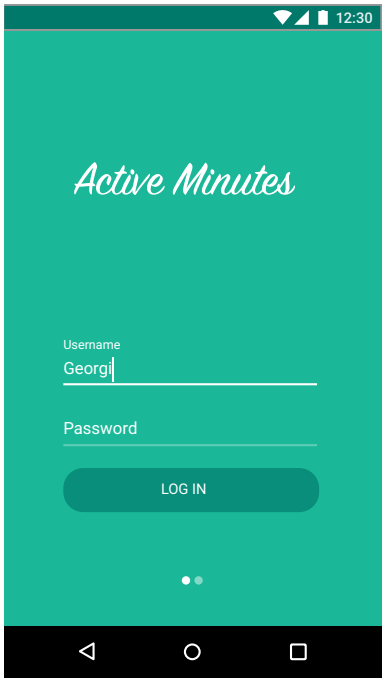


(a) History Screen, mode daily

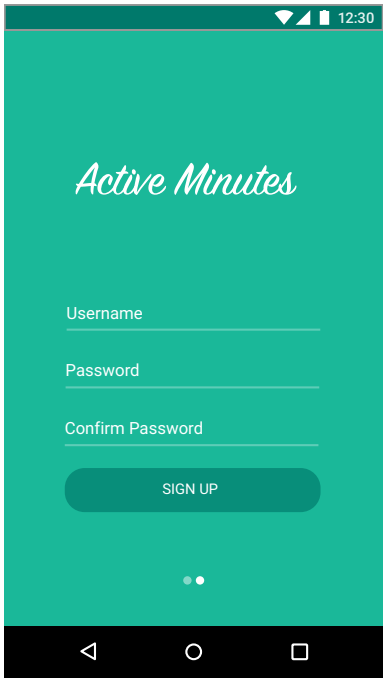


(b) History Screen, mode weekly

Figure C.2: History Screens

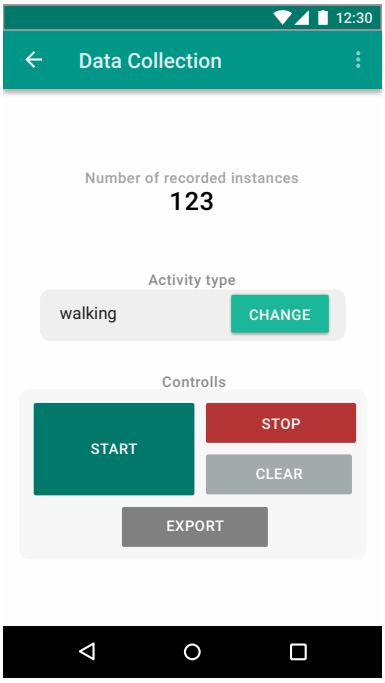


(a) Login Screen

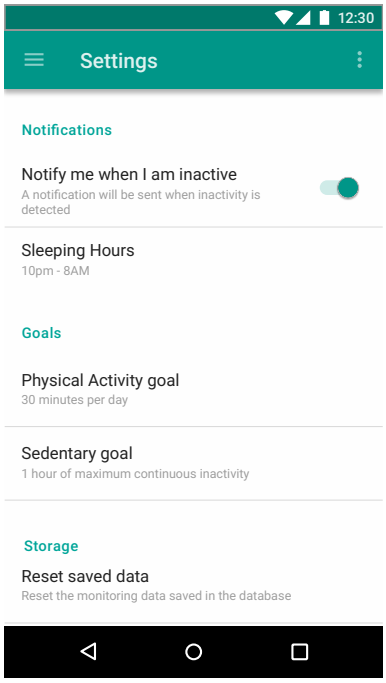


(b) Signup Screen

Figure C.3: Authentication Screens



(a) Data Collection Screen



(b) Settings Screen

Figure C.4: Utility Screens

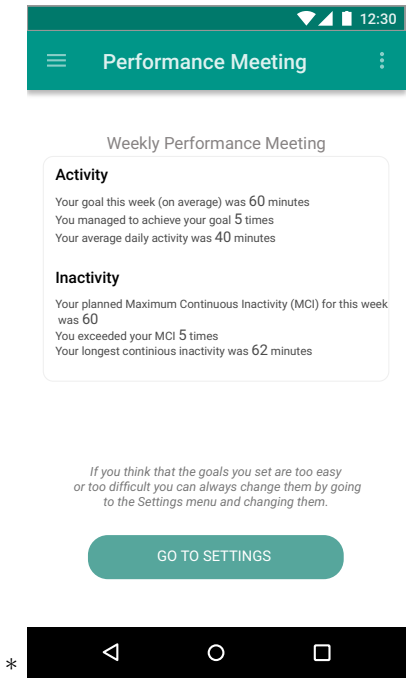
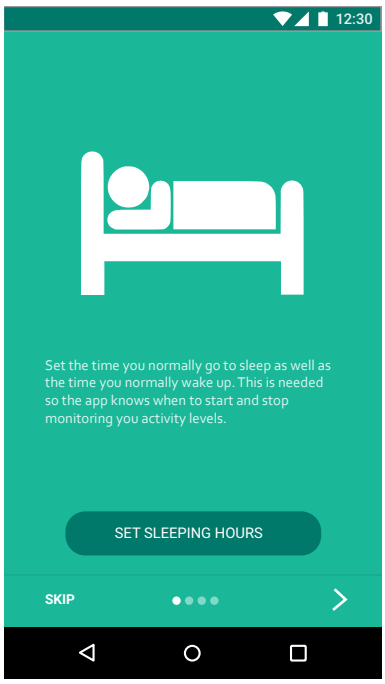
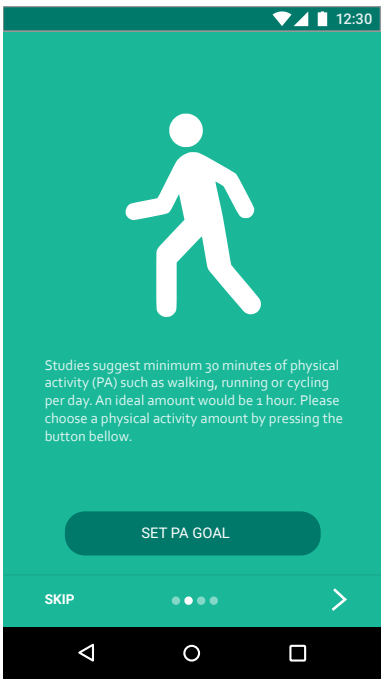


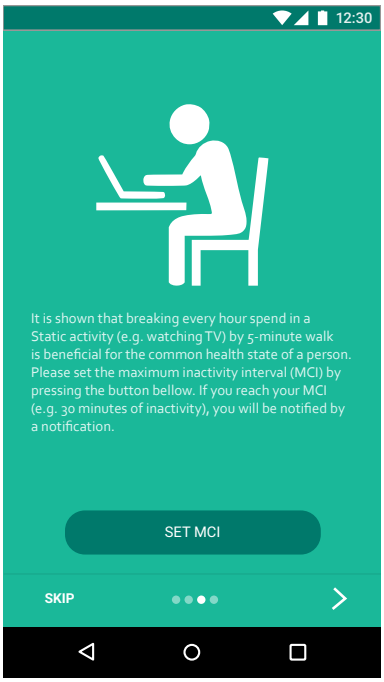
Figure C.5: Performance Meeting Screen



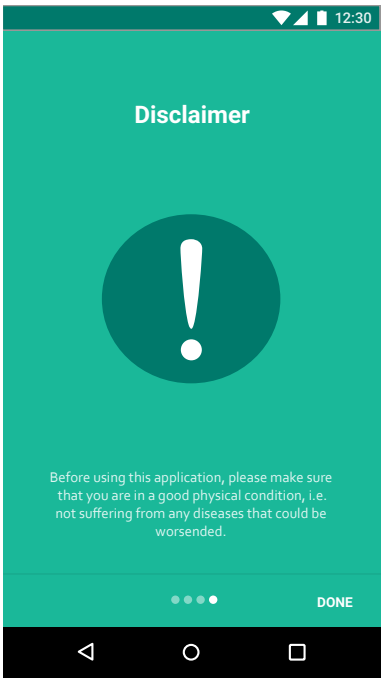
(a) Set Sleeping Hours Screen



(b) Set PA Screen



(c) Set MCI Screen



(d) Disclaimer Screen

Figure C.6: Initial setup screens

Appendix D

Dagger Component and Module

```
@ActivityScope
@Subcomponent(modules = AuthModule.class)
public interface AuthComponent {

    void inject(LoginFragment loginFragment);

    }
}
```

Listing D.1: Dagger Component example

```
@Provides
@ActivityScope
ILoginModel provideLoginModel(IAuthDataManager dataManager) {
    return new LoginModel(dataManager);
}

@Provides
@ActivityScope
ILoginPresenter provideLoginPresenter(ILoginModel model) {
    return new LoginPresenter(model);
}
```

Listing D.2: Dagger Module example

```
public class LoginFragment extends Fragment implements ILoginView {
@Inject
ILoginPresenter mPresenter;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        satisfyDependencies();

        mPresenter.setView(this);
    }

    private void satisfyDependencies() {
        ((ActiveMinutesApplication) getActivity()
                .getApplication()).buildAuthComp().inject(this);
    }

}
```

Listing D.3: Dependency injection example

Appendix E

ARFF dataset file sample

```
@relation HAR

@attribute accX__fft1 numeric
@attribute accX__fft2 numeric
@attribute accX__fft3 numeric
@attribute accX__fft4 numeric
@attribute accX__fft5 numeric
@attribute accY__fft1 numeric
@attribute accY__fft2 numeric
@attribute accY__fft3 numeric
@attribute accY__fft4 numeric
@attribute accY__fft5 numeric
@attribute accZ__fft1 numeric
@attribute accZ__fft2 numeric
@attribute accZ__fft3 numeric
@attribute accZ__fft4 numeric
@attribute accZ__fft5 numeric
@attribute accM__fft1 numeric
@attribute accM__fft2 numeric
@attribute accM__fft3 numeric
@attribute accM__fft4 numeric
@attribute accM__fft5 numeric
@attribute class {walking,running,static,cycling}

@data
9,-76,-834,25,22,4,-1,-401,-190,-3,3,5,-213,-94,-45,2,-50,-49,-15,18,walking
-0,-184,140,4,-34,-2,13,45,63,-8,0,51,-68,-30,-72,1,-52,15,-51,63,walking
-1,-13,-30,3,2,3,-18,-2,3,-21,-1,2,-4,-9,-1,-2,33,3,16,20,walking
0,-16,-17,-22,6,-3,-6,-8,-16,-3,-15,-2,12,-5,9,8,8,37,38,-18,walking
-6,-22,14,-54,1,14,8,27,9,60,-22,-31,-11,-11,4,-7,1,0,-3,-39,running
-3,-11,-35,-3,-53,-13,-45,23,-57,17,-2,52,13,58,15,22,48,-36,25,-50,running
3,7,-11,27,-41,-19,-31,-53,-107,10,0,-42,22,-42,3,20,34,-26,76,-104,running
-3,37,34,9,58,0,30,97,-74,83,4,-13,-39,40,26,-4,15,-10,27,-26,cycling
-1,-91,-131,-121,-58,-4,6,-66,22,70,3,-30,-76,-63,-30,-1,-39,-10,-39,-76,
    cycling
1,-54,-41,-180,114,-3,4,-21,78,-188,3,8,-56,-19,70,0,-38,22,-83,103,cycling
0.3,0,-1,1,-1.4,0,1,-1,0,0,0,-1,0,0,0,0,0,0,0,0,static
0,-6,48,-8,0,0,-6,-21,2,5,0,-1,42,-6,0,0,0,-1.5,0,-1.6,static
-0.3,8,-4,3,-4,0.1,6,-4,2,-3,0,2,-2,2,-1,0,0.2,1,-0.4,0.2,static
```

Listing E.1: WEKA ARFF file extract

Appendix F

Project Log

W/C	Activities
03/10/2016	• Investigating a research topic • Devising the Project Proposal document structure and content • Researching how the project could make a contribution to the society
10/10/2016	• Devising overall project structure • Devising the content of the Project's ethics form
17/10/2016	• Working on the Ethical form document • Improving the content of the project proposal document
24/10/2016	• Researching Sedentary Behaviour consequences • Investigating self-management approaches • Investigating HAR on mobile devices
31/10/2016	• Implementing supervisor Project Proposal feedback • Improving the Project report structure • Devising the content and structure of the Literature Review Chapter
07/11/2016	• Researching Dependency Injection design principle • Researching LaTeX and BibTex
14/11/2016	• Continue reading about Dependency Injection in Software development • Continue learning the basis of LaTeX
21/11/2016	• Discussing the structure of the literature review document as well as ways how to improve it • Continue learning the basis of LaTeX • Learn the basics of Dagger 2 library on Android
28/11/2016	• Implemented suggested feedback changes to the Literature Review document • Continue learning the basis of LaTeX • Experimenting and testing Dagger 2 on Android
05/12/2016	• Continue researching how HAR could be implemented on mobile device
12/12/2016	• Implemented suggested feedback changes to the literature review document • Start learning version control (GitHub) basics
19/12/2016	• Discussing ways of improving the literature review document • Discussing the first draft of the System Specifications Chapter • Learn Astah Professional

W/C	Activities
26/12/2016	• Finalising the Literature Review deliverable • Start learning Sketch 3 • Investigating system overall design
02/01/2017	• Continue learning Sketch 3 • Research appropriate software architectural model • Drafting use case diagram for the system using Astah Professional
09/01/2017	• Research User interface design • Draft the design of the mobile application • Research necessary data features to be extracted • Learn how to implement MVP architectural pattern on the Android platform • Research software Architectural patterns
16/01/2017	• Reading about Software architecture • Researching Software Architecture patterns • Produce use case diagram of the system • Designing the system's database • Read about Component diagrams
23/01/2017	• Producing first draft of the system architecture (Component diagram) using Astah Professional • Finishing the system architecture diagram
30/01/2017	• Adding the Architectural Design section content to the LaTeX document • Refining the UI of the mobile application • Creating first design prototype with Flinto • Prepare system design for implementation
06/02/2017	• Implementing system prototype feedback from client • Started the implementation of the mobile app • Data collection screen UI and partial functionality produced after the first iteration of the development
13/02/2017	• Improving the History's daily and weekly mode screen designs • Improving the Data Collection Screen functionality • Gathering data from the project's participants • Training a general classifier • Implementing the Log-in screen functionality • The second iteration produced full log in & Signup screens functionality
20/02/2017	• Improving the Sign-up screen UI & functionality • Start implementing the navigation drawer UI • Start the implementation of the Monitoring component
27/02/2017	• Improved the UI of the Today screen as part of the third iteration of the development • Improved the feedback functionality. Now the app notifies the user for both the activate time and inactive time
6/03/2017	• Started the implementation of the History Daily screen
13/03/2017	• Started & finished the implementation of the History Weekly screen • Improved the UI of the History Daily screen • Implemented Settings screen
20/03/2017	• Researched Android Notifications API • Implemented the Self-management component of the application

W/C	**Activities**
27/03/2017	• Due to technical difficulties no software development was done this week
03/04/2017	• Work on the Introduction chapter of the document • Write the initial Evaluation chapter version • Applying supervisor report feedback • Improving the UI of the History screen
10/04/2017	• Prepare a questionnaire to help evaluate the effectiveness of ActiveMinutes • Perform Acceptance tests on the implemented mobile application • Discuss the results of the above-mentioned test in the Evaluation chapter
17/04/2017	• Gather user feedback via a survey • Present *"ActiveMinutes"* at the Future Developed 2017 computing show • Finishing the Evaluation and Conclusion chapter
24/04/2017	• Devising Usability and Functional testing plans • Carrying out the above tests and gather user feedback • Presenting and demonstrating the project • Report content improvements & submission

Table F.1: Project Log

Appendix G

Functional Testing

G.1 Navigational

Step No	Test Cases	Expected result	Actual result	Status
1	The user opens the application for the first time	The system should should the authentication screen to the user.	The system shows correctly the authentication screen allowing the user to register with the system.	Pass
2	The user navigates to the *Login* screen and performs a right-to-left swipe gesture.	The system should shows the *Signup* screen to the user.	The system works as expected.	Pass
3	The user presses the "Next" (e.g. the arrow symbol) button located in the bottom right corner of the *Initial setup* screens.	The system should advance the user to the next *Initial setup* screen.	The system correctly updates the screen transfers the user to the next screen.	Pass
4	The user performs a right-to-left swipe gesture during the *Initial Setup* process.	The system should show the next setup screen.	The system correctly updates the screen and transfers the user to the next screen.	Pass
5	The user finishes the *Initial Setup* process by pressing the *"DONE"* button.	The system should show the *Today* screen to the user.	The system works as expected and transfers the user to the *Today* screen.	Pass
6	The user navigates to the *Today* screen and performs a left-to-right swipes gesture from the edge of the screen.	The system should show the *Navigational drawer* to the user.	The system navigational drawer is correctly shown to the user.	Pass
7	The user taps on the "Hamburger icon" at the top left corner of the screen.	The system should show the *Navigational drawer* to the user.	The navigational drawer is correctly shown to the user.	Pass
8	The user opens the *Navigational drawer* of the application and selects the *Today* menu item.	The system should show the "Today" screen to the user.	Works as expected.	Pass

Step No	Test Cases	Expected result	Actual result	Status
9	The user opens the *Navigational drawer* of the application and selects the *History* menu item.	The system should show the "History" screen to the user.	Works as expected.	Pass
10	The user opens the *Navigational drawer* of the application and selects the *Settings* menu item.	The system should show the "Settings" screen to the user.	Works as expected.	Pass
11	The user opens the *Navigational drawer* of the application and selects *Log Out* menu item.	The system should show the "Login" screen to the user.	Works as expected.	Pass

G.2 Goal-setting (Initial Setup)

The following functional tests are performed during the *Initial Setup* process (e.g. upon first sign-up of a new user).

Step No	Test Cases	Expected result	Actual result	Status
1	The user presses the "Set PA Goal" button and selects a value from the list.	The system should accept the user's input and show a confirmation message.	Works as expected.	Pass
2	The user presses the "Set MCI" button and selects a value from the list.	The system should accept the user's input and show a confirmation message.	Works as expected.	Pass
3	The user presses the "Set Sleeping Hours" button	The system should show a date picker dialog.	Works as expected.	Pass
4	The user selects a *From* and *To* value from the *Sleeping Hours* date picker and presses the "OK" button.	The system should accept the user input and show a confirmation message.	Works as expected.	Pass
5	The user finishes the *Initial Setup* by pressing the "DONE" button.	The system should show the values set during the *Initial Setup* throughout the application i.e., *Today* and *Settings* screens.	Works as expected.	Pass

G.3 Goal-setting (Settings screen)

The following functional tests are performed in the *Settings* screen.

Step No	Test Cases	Expected result	Actual result	Status
1	The user presses the "Physical Activity goal" menu item and selects a value from the list.	The system should accept the user's input and show a confirmation message.	Works as expected.	Pass

Step No	Test Cases	Expected result	Actual result	Status
2	The user presses their presses the "Physical Activity goal" menu item and selects a value from the list.	The system should update the UI of the *Today* screen to reflect the change.	Works as expected.	Pass
3	The user presses the "Sedentary goal" menu item and selects a value from the list.	The system should accept the user's input and show a confirmation message.	Works as expected.	Pass
4	The user presses their presses the "Physical Activity goal" menu item and selects a value from the list.	The system should update the UI of the *Today* screen to reflect the change.	Works as expected.	Pass
5	The user presses the "Sleeping Hours" menu item.	The system should show a date picker dialog.	Works as expected.	Pass
6	The user selects a *From* and *To* value from the *Sleeping Hours* date picker and presses the "OK" button.	The system should accept the user input and show a confirmation message.	Works as expected.	Pass

G.4 Monitoring Activity

The following functional tests check if the system monitors PA correctly.

Step No	Test Cases	Expected result	Actual result	Status
1	The user starts the mobile application and performs 5 minutes of **walking**.	The system should recognise that the user is performing PA and increments *Active* time.	Works as expected e.g., updates the UI by adding 5 minutes to the PA progress bar in *Today* & *History* screens.	Pass
2	The user starts the mobile application and performs 5 minutes of **running**.	The system should recognise that the user is performing PA and increments *Active* time.	Works as expected e.g., updates the UI by adding 5 minutes to the PA progress bar in *Today* & *History* screens.	Pass
3	The user starts the mobile application and performs 5 minutes of **cycling**.	The system should recognise that the user is performing PA and increments *Active* time.	Works as expected e.g., updates the UI by adding 5 minutes to the PA progress bar in *Today* & *History* screens.	Pass

G.5 Monitoring Inactivity

The following functional tests check if the system monitors ST correctly.

Step No	Test Cases	Expected result	Actual result	Status
1	The user starts the mobile application and deletes all data in the database. Next, the user sits on a chair for 5 minutes.	The system should recognise that the user is static and increment *Static* time.	Works as expected e.g., updates the UI by adding 5 minutes to the *"Longest inactivity interval"* text view located in the *Today*.	Pass

G.6 Notifications

This section of the functional testing checks if the system sends notifications to the user correctly.

Step No	Test Cases	Expected result	Actual result	Status
1	The user achieves their goal of 30 minutes of PA	The system should send a notification to the user at 80% achieved goal to motivate the user as well as when reaching the goal.	Works as expected. The user receives the above-mentioned notifications.	Pass
2	The user reaches their target of 30 minutes of ST (or MCI)	The system should send a notification to the user at 80% of the reached target to notify the user as well as when reaching the target.	Works as expected. The user receives the above-mentioned notifications.	Pass

G.7 Other

Step No	Test Cases	Expected result	Actual result	Status
1	The user navigates to the *Settings* screen and selects the *"Reset saved data"* menu item.	The system should a dialog window asking the user to confirm the action	Works as expected.	Pass
2	The user confirms the above-mentioned action by pressing the *"OK"* button.	The system should delete the saved data in the database and that should be reflected by the *Today* and *History* screens	Works as expected.	Pass

Appendix H

Usability Testing

Task No	Task	Comments	Status
1	Open the application and create an account.	The user was able to understand what was required and complete the process by accessing the Sign-up screen, entering their credentials and selecting the *SIGN UP* button.	Pass
2	Navigate to the *Today* screen.	The user was able to navigate to the screen by accessing the navigational drawer and selecting the *Today* text label.	Pass
3	Navigate to the *Today* screen and describe what you are seeing.	The user was able to explain correctly the purpose of the screen i.e., the screen shows the current activity and inactivity goals as well as their progress	Pass
4	Navigate to the *History* screen.	The user was able to navigate to the screen by accessing the navigational drawer and selecting the *History* text label.	Pass
5	Navigate to the *History* screen and describe what you are seeing.	The user was able to explain correctly the purpose of the screen i.e., the screen shows past goal-performance in both representations *Daily* and *Weekly*. **Note:** It took the user a bit more time to associate the PA and MCI text labels with the two progress bar for each list item.	Pass
6	Navigate to the *History* screen; switch to *Daily* view of the list and select a day from the list.	The user was able to perform the task without any signs of confusion.	Pass
7	After selecting a list item in task 6, explain what you are seeing on the screen.	The user was able to correctly explain what information the dialog window was showing.	Pass
8	Navigate to the *History* screen, switch to *Weekly* view of the list and explain what you are seeing.	The user was able to correctly explain the information shown on the screen.	Pass
9	When task 8 is completed, select an item from the list and explain what you are seeing.	The user was able to correctly explain what information the dialog window was showing.	Pass

Task No	**Task**	**Comments**	**Status**
10	Navigate to the *Settings* screen and explain what you are seeing.	The user was able to navigate to the screen without any signs of confusion and correctly explain the purpose of the screen (i.e. provide access to personalising the functionality of the application).	Pass
11	Navigate to the *Settings* screen and change the physical activity goal.	The user was able to successfully change the goal by selecting the *Physical Activity goal* from the list and selecting a value from the dialog window.	Pass
12	Navigate to the *Settings* screen and change the sedentary goal.	The user was able to successfully change the goal by selecting the *Sedentary goal* from the list and selecting a value from the dialog window.	Pass
13	Navigate to the *Settings* screen and change the *Sleeping Hours* time interval.	The user was able to successfully change the time interval by setting the appropriate *From* and *To* values from the dialog date picker.	Pass
14	Navigate to the *Settings* screen and switch off the inactivity notifications.	The user was able to successfully turn off the inactivity notifications by tapping on the check box next to the *Notify me when I am inactive* settings item.	Pass
15	Log out of the application.	The user was able to log out of the system by accessing the navigational drawer and selecting the *Log Out* button.	Pass
16	Log in to the application	The user was able to log in to the system by opening the application, typing their credentials and selecting the *LOG IN* button.	Pass

Appendix I

Survey results

No	**Question**	**Response summary**
1	What is your age?	18 to 24 (100%)
2	Were you more active then usual (e.g. walking, running) as a result of using ActiveMinutes?	Yes (33.33%) No (33.33%) Not sure (33.33%)
3	When notified for prolonged inactivity (e.g. 30 minutes of inactivity), did you try to do at least 5 minutes of physical activity?	Yes (66.67%) No (0%) Sometimes (33.33%)
4	Does seeing past days goal-performance (e.g. in the History screen) motivate you to achieve a goal?	Yes (100%) No (0%) Not sure (0%)
5	Does the visual feedback (i.e. the green progress bar) encourage you to achieve your goal?	Yes (100%) No (0%) Somewhat (0%)
6	Do you think the application was accurate when measuring your activity levels?	Yes (66.67%) No (0%) Somewhat (33.33%)
7	How easy to understand was the user interface of the application? (0 being very difficult and 100 being very easy)	Response 1 - 100 Response 2 - 80 Response 3 - 75 **Average - 85**
8	Was the mobile application battery-friendly?	Yes (66.67%) No (0%) Somewhat (33.33%)

No	**Question Response**	**Response summary**
9	Do you have any suggestions regarding the mobile application in general?	(1) "Great application, I like the feature that notifies you when you have been inactive for a certain amount of time. The application could maybe have a feature to alert you when you should go to bed to achieve the best amount of sleep. For future development, the application could maybe sync with a watch." (2) "Slightly wider ranges in terms of sleep time and active and inactive minutes and perhaps a more obvious "Key" informing the users as to what each bar in the history means. Otherwise really easy to use." (3) "The application could be improved in the future by optimising it to be more battery-friendly."

Appendix J

Third-party software used in this work

Weka for Android

Author RJ

Usage A stripped down Java API for Machine Learning on Android

License Copyright (c) 2017 University of Waikato

> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

MaterialDrawer

Author Mike Penz

Usage Navigational drawer implementation

License Copyright 2016 Mike Penz

> Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
>
> http://www.apache.org/licenses/LICENSE-2.0
>
> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Butter Knife

Author Jake Wharton

Usage Simplifying the Android view-binding process while reducing the amount of boilerplate code

License Copyright 2013 Jake Wharton

> Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
>
> http://www.apache.org/licenses/LICENSE-2.0
>
> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Dagger 2

Author Google (forked from square)

Usage A Dependency Injection library

License Copyright 2012 The Dagger Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JTransforms

Author Piotr Wendykier

Usage Calculating the FFT algorithm coefficients

License Copyright (c) 2007 onward, Piotr Wendykier Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Material Dialogs

Author Aidan Follestad

Usage Implementation of Android dialogs

License Copyright (c) 2014-2016 Aidan Michael Follestad

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

EventBus

Author Markus Junginger

Usage Communication between Activity and Service

License Copyright (C) 2012-2016 Markus Junginger

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AppIntro

Author APL Developers

Usage Implementation of *ActiveMinutes*'s intro screens as well as Authentication screens

License Copyright 2015 Paolo Rotolo Copyright 2016 Maximilian Narr

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Material Date and Time Picker with Range Selection

Author Supratim Chakraborty

Usage *SleepingHours* time dialog picker

License Copyright 2015-2016 Supratim Chakraborty

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Android-Job

Author Evernote

Usage Scheduling task library. Used to schedule the *ActiveMinutesService* to start and stop at the set *SleepingHours*

License Copyright (c) 2007-2016 by Evernote Corporation, All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Appendix K

ActivityMonitor class

```
    public class ActivityMonitor implements Observer, IActivityMonitor {
    // increment time is 3 seconds
    private final int INCREMENT_AMOUNT = 3;
    ...

    public ActivityMonitor(IActivityDataManager dataManager, IFeedbackProvider feedbackProvider) {
        mDataManager = dataManager;
        correctionList = new LimitedSizeList<>(3);
        this.mFeedbackProvider = feedbackProvider;
    }

    @Override
    public void update(Observable observable, Object activity) {
        monitorActivity(activity);
    }

    @Override
    public void monitorActivity(Object activityClass) {
        int activityInt = (int) activityClass;
        int currentPa = 0;
        int currentSt = 0;
        // Get the user's PA goal...
        int paGoal = mDataManager.getUserPaGoal();
        int stTarget = mDataManager.getMaxContInacTarget();

        correctionList.add(activityInt);
        int correctResult = applyCorrection();
        // check of the correctedResult contains activity of type ACTIVE
        if (correctResult == correctionList.size()) {
            // It does, so incrementActive time.
            currentPa = mDataManager.incActiveTime();
            // clear current inactive interval since activity is detected...
            mDataManager.clearCurrentInacInterval();
        } else {
            currentSt = mDataManager.incCurrentInacInterval();
        }
        // Check if the user's goal is achieved and make sure that
        if (currentPa > paGoal && currentPa <= paGoal + INCREMENT_AMOUNT) {
            // is achieved, so notify for goal achieved..
            mFeedbackProvider.provideGoalAchievedFeedback(paGoal);
        }
        // Check to see if the PA goal is 80% achieved
        if (currentPa > paGoal * ENCOURAGEMENT_PERCENTAGE &&
                currentPa <= paGoal * ENCOURAGEMENT_PERCENTAGE + INCREMENT_AMOUNT) {
            // is achieved, so notify for goal achieved..
            int leftMinutesTillGoal = (int) (paGoal - (paGoal * ENCOURAGEMENT_PERCENTAGE));
            mFeedbackProvider.provideEncouragingFeedback(leftMinutesTillGoal);
        }

        if (currentSt % stTarget == 0 && currentSt != 0) {
            mFeedbackProvider.provideProlongedInactivityFeedback(currentSt);
            System.out.println("currentSt % stTarget == 0 currentSt: " + currentSt);
        }
        // Check to see if the ST target is 80% reached
        if (currentSt > stTarget * WARNING_PERCENTAGE &&
                currentSt <= stTarget * WARNING_PERCENTAGE + INCREMENT_AMOUNT) {
            mFeedbackProvider.provideWarningProlongedInactivityFeedback(currentSt);
        }
    }

    @Override
    public void setUser(User user) {
        this.mDataManager.setUser(user);
    }

    private int applyCorrection() {
        int count = 0;
        // iterate through the correctionList containing the last 3 recognised classes (i.e. 0,1,3)
        for (int item : correctionList) {
            // check if the current item is of class "cycling" - 3; "running" - 1;"walking" - 0
            if (item == 0 || item == 1 || item == 3) {
                // if it is, increment the counter
                count++;
            }
        }

        return count;
    }
}
```

Listing K.1: ActivityMonitor class

Appendix L

ActiveMinutesService

```
    public class ActiveMinutesService extends Service
    implements SensorEventListener {
    ...
    @Inject
    @Named("classify")
    IHarManager mHarManager;

    @Inject
    IActivityMonitor mActivityMonitor;
    ...
    @Override
    public void onCreate() {
        super.onCreate();
        satisfyDependencies();
       ...
        setUpActivityMonitor();


    }

    private void setUpActivityMonitor() {
        // set the logged in user...
        mActivityMonitor.setUser(mAuthDataManager.getLoggedInUser());
        // add observable mHarManager to the observer ActivityMonitor
        ((HarClassifyManager) mHarManager)
        .addObserver((Observer) mActivityMonitor);
    }
    ...
    private void satisfyDependencies() {
        ((ActiveMinutesApplication) getApplication()).getComponent()
        .inject(this);
    }

    private void runAsForeground() {
        ...
    }

    @Override
    public void onSensorChanged(SensorEvent sensorEvent) {
        mHarManager.feedData(sensorEvent.values, sensorEvent.timestamp);
    }
   ...
}
```

Listing L.1: ActiveMinutesService class (simplified)

Appendix M

Ethics Form

THIS PAGE INTENTIONALLY LEFT BLANK

STUDENT PROJECT ETHICAL REVIEW (SPER) FORM

ROBERT GORDON UNIVERSITY•ABERDEEN

Student Name	Georgi Koemdzhiev
Supervisor	Dr Stewart Massie
Project Title	A Mobile Application that applies a Self-Managing Approach to Reducing Sedentary Behaviour
Course of Study	Computing For Graphics and Animations
School/Department	The School of Computing Science and Digital Media

Part 1 : Descriptive Questions			
1	Does the research involve, or does information in the research relate to:	Yes	No
	(a) individual human subjects	-	
	(b) groups (e.g. families, communities, crowds)		-
	(c) organisations		-
	(d) animals?		-
	Please provide further details:		
	The project will require collecting activity data (accelerometer data from a smartphone) from project participants. Data will be collected from the following activities - walking, running, cycling, standing, sitting, laying and ascending and descending stairs.		
2	Will the research deal with information which is private or confidential?	Yes	No
			-
	Please provide further details:		

Part 2: The Impact of the Research			
3	In the process of doing the research, is there any potential for harm to be done to, or costs to be imposed on	Yes	No
	(a) research participants?		-
	(b) research subjects?		-
	(c) you, as the researcher?		-
	(d) third parties?		-
	Please state what you believe are the implications of the research:		
4	When the research is complete, could negative consequences follow:	Yes	No
	(a) for research subjects		-
	(b) or elsewhere?		-
	Please state what you believe are the consequences of the research:		

Part 3: Ethical Procedures			
5	Does the research require informed consent or approval from:	Yes	No
	(a) research participants?		-
	(b) research subjects		-
	(c) external bodies		-
	If you answered yes to any of the above, please explain your answer:		
6	Are there reasons why research subjects may need safeguards or protection?	Yes	No
		-	
	If you answered yes to the above, please state the reasons and indicate the measures to be taken to address them:		
	The participants, during the data collection process, will perform several activities, some of which (e.g. cycling) may require safety equipment. However, participants are only being asked to undertake an activity they would normally be doing and so will use their existing equipment where required.		
7	Has PVG membership status been considered?		-
	(a) PVG membership is not required.		-
	(b) PVG membership is required for working with children.		-
	(c) PVG membership is required for working with protected adults.		-
	(d) PVG membership is required for working with both children and protected		-
	If you answered yes to (b), (c) or (d) above, please give details:		
8	Are specified procedures or safeguards required for recording, management, or storage of data?	Yes	No
		-	
	If you answered yes to the above, please outline the likely undertakings: Data will be stored in an anonymised form.		

Part 4: The Research Relationship			
9	Does the research require you to give or make undertakings to research participants or subjects about the use of data?	Yes	No
			-
	If you answered yes to the above, please outline the likely undertakings:		
10	Is the research likely to be affected by the relationship with a sponsor, funder or employer?	Yes	No
			-
	If you answered yes to the above, please identify how the research may be affected:		

Part 5: Other Issues			
11	Are there any other ethical issues not covered by this form which you believe you should raise?	Yes	No
			-

Statement by Student I believe that the information I have given in this form is correct, and that I have addressed the ethical issues as fully as possible at this stage.			
Signature		Date	06.10.2016

If any ethical issues arise during the course of the research, students should complete a further Student Project Ethical Review (SPER) form.

The *Research Ethics Policy* is available at www.intranet.rgu.ac.uk/credo/staff/page.cfm?pge=7060

Part 6: To be completed by the supervisor			
12	Does the research have potentially negative implications for the University?	Yes	No
			X
	If you answered yes to the above, please explain your answer:		
13	Are any potential conflicts of interest likely to arise in the course of the research?	Yes	No
			X
	If you answered yes to the above, please identify the potential conflicts:		
14	Are you satisfied that the student has engaged adequately with the ethical implications of the work? [In signifying agreement, supervisors are accepting part of the ethical responsibility for the project]	Yes X	No
	If you answered no to the above, please identify the potential issues:		
15	**Appraisal:** Please select one of the following		
	The research project should proceed in its present form – no further action is required	X	
	The research project requires ethical approval by the School Ethics Review Panel		
	The research project needs to be returned to the student for modification prior to further action		
	The research project requires ethical review by an external body. If this applies please give details		
	Title of External Body providing ethical review		
	Address of External Body		
	Anticipated date when External Body may consider project		

Affirmation by Supervisor			
I have read the student's responses and have discussed ethical issues arising with the student. I can confirm that, to the best of my understanding, the information presented by the student is correct and appropriate to allow an informed judgement on whether further ethical approval is required.			
Signature	Stewart Massie	**Date**	06/10/16

Appendix N

Consent Forms

THIS PAGE INTENTIONALLY LEFT BLANK

A Mobile Application that applies a Self-Managing Approach to Reducing Sedentary Behaviour

Nowadays, with the rapid advance of technology people spend a lot of their time in a static position. This project will develop a smartphone application that will help inactive people reduce sedentary behaviour by monitoring their inactivity during the day, notifying the user when extensive inactivity times are detected. What is more, the application will give feedback upon goal achievements when the user has accomplished the previously set goals in order to motivate and encourage activity.

By signing this form, you consent to be a participant in the research project.

Your responses will be kept strictly confidential. If at any time during the session you feel unable or unwilling to continue, you are free to leave - participation is voluntary. If you not wish to answer any particular question or questions, you are free to decline. Your name will not be linked with the research materials, and will not be made public in any report produced by the researcher.

If you have any questions about this research, these will be answered as best as possible. After todays study, if you require further information, please contact

Georgi Koemdzhiev
g.koemdzhiev@rgu.ac.uk

RL	I confirm that I have read and understand this form. I have had the opportunity to consider the information, ask questions, and have had these answered satisfactorily
RL	I understand that my participation is voluntary and that I am free to withdraw from the study at any time without giving any reason and without penalty.
RL	I understand that data collected during the study may be looked at by individuals from the research team where it is relevant to my taking part in this research. I give permission for these individuals to have access to my data.
RL	I understand that in the event of a publication no data will be identifiable as my own
RL	I agree that an audio/video recording of this session can be created for research purposes
RL	I agree to take part in this study

RORY LENG		16.02.2017
Participant Name	Participant Signature	Date
Georgi Koemdzhiev		16.02.2017
Researcher Name	Researcher Signature	Date

A Mobile Application that applies a Self-Managing Approach to Reducing Sedentary Behaviour

Nowadays, with the rapid advance of technology people spend a lot of their time in a static position. This project will develop a smartphone application that will help inactive people reduce sedentary behaviour by monitoring their inactivity during the day, notifying the user when extensive inactivity times are detected. What is more, the application will give feedback upon goal achievements when the user has accomplished the previously set goals in order to motivate and encourage activity.

By signing this form, you consent to be a participant in the research project.

Your responses will be kept strictly confidential. If at any time during the session you feel unable or unwilling to continue, you are free to leave - participation is voluntary. If you not wish to answer any particular question or questions, you are free to decline. Your name will not be linked with the research materials, and will not be made public in any report produced by the researcher.

If you have any questions about this research, these will be answered as best as possible. After todays study, if you require further information, please contact

Georgi Koemdzhiev
g.koemdzhiev@rgu.ac.uk

CJR	I confirm that I have read and understand this form. I have had the opportunity to consider the information, ask questions, and have had these answered satisfactorily
CJR	I understand that my participation is voluntary and that I am free to withdraw from the study at any time without giving any reason and without penalty.
CJR	I understand that data collected during the study may be looked at by individuals from the research team where it is relevant to my taking part in this research. I give permission for these individuals to have access to my data.
CJR	I understand that in the event of a publication no data will be identifiable as my own
CJR	I agree that an audio/video recording of this session can be created for research purposes
CJR	I agree to take part in this study

Corrie Robb		16/2/2017
Participant Name	Participant Signature	Date
Georgi Koemdzhiev		16.02.2017
Researcher Name	Researcher Signature	Date

A Mobile Application that applies a Self-Managing Approach to Reducing Sedentary Behaviour

Nowadays, with the rapid advance of technology people spend a lot of their time in a static position. This project will develop a smartphone application that will help inactive people reduce sedentary behaviour by monitoring their inactivity during the day, notifying the user when extensive inactivity times are detected. What is more, the application will give feedback upon goal achievements when the user has accomplished the previously set goals in order to motivate and encourage activity.

By signing this form, you consent to be a participant in the research project.

Your responses will be kept strictly confidential. If at any time during the session you feel unable or unwilling to continue, you are free to leave - participation is voluntary. If you not wish to answer any particular question or questions, you are free to decline. Your name will not be linked with the research materials, and will not be made public in any report produced by the researcher.

If you have any questions about this research, these will be answered as best as possible. After todays study, if you require further information, please contact

Georgi Koemdzhiev
g.koemdzhiev@rgu.ac.uk

R.C	I confirm that I have read and understand this form. I have had the opportunity to consider the information, ask questions, and have had these answered satisfactorily
R.C	I understand that my participation is voluntary and that I am free to withdraw from the study at any time without giving any reason and without penalty.
RC	I understand that data collected during the study may be looked at by individuals from the research team where it is relevant to my taking part in this research. I give permission for these individuals to have access to my data.
R.C	I understand that in the event of a publication no data will be identifiable as my own
R.C	I agree that an audio/video recording of this session can be created for research purposes
RC	I agree to take part in this study

R. Clarke	R. Clarke	16/02/17
Participant Name	Participant Signature	Date
Georgi Koemdzhiev	[signature]	16.02.2017
Researcher Name	Researcher Signature	Date

Appendix O

Presentation Slides

THIS PAGE INTENTIONALLY LEFT BLANK

5/2/2017

A Mobile application that applies a Self-Management Approach to Reduce Sedentary Behaviour

STUDENT: GEORGI KOEMDZHIEV
SUPERVISOR: DR. STEWART MASSIE

Sedentary Behavior as a threat to our physical wellbeing

Sedentary Behaviour causes:

- Back pain
- Type 2 Diabetes
- Cardiovascular disease
- Cancer
- All-cause mortality
- Leading cause of absence from work
- Annually coasts the EU 80 billion euro

5/2/2017

Project aim

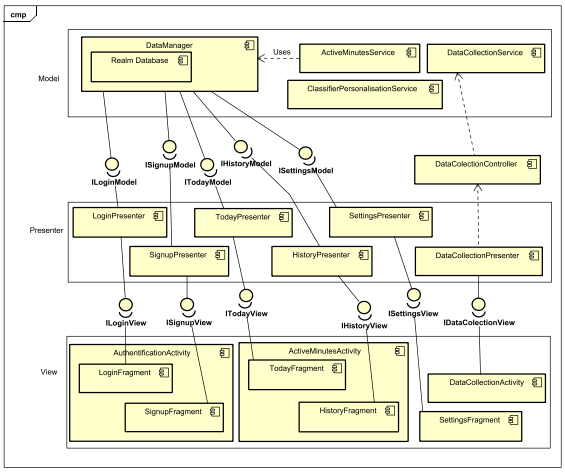
Active Minutes

Objectives Achieved

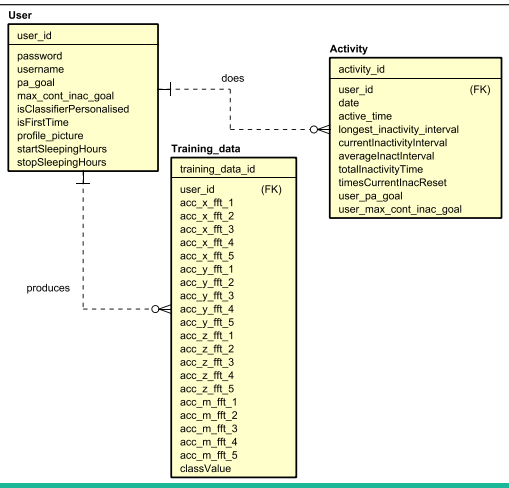
- Research Sedentary Behaviour (SB) and behaviour change
- Investigate how to implement Human Activity Recognition (HAR) on wearable devices
- Develop a fully working HAR system
- Incorporate the above within a SB self-management system
- Design and Implement the mobile application
- System evaluation
- Release app to application to the Application Store

5/2/2017

System Architecture

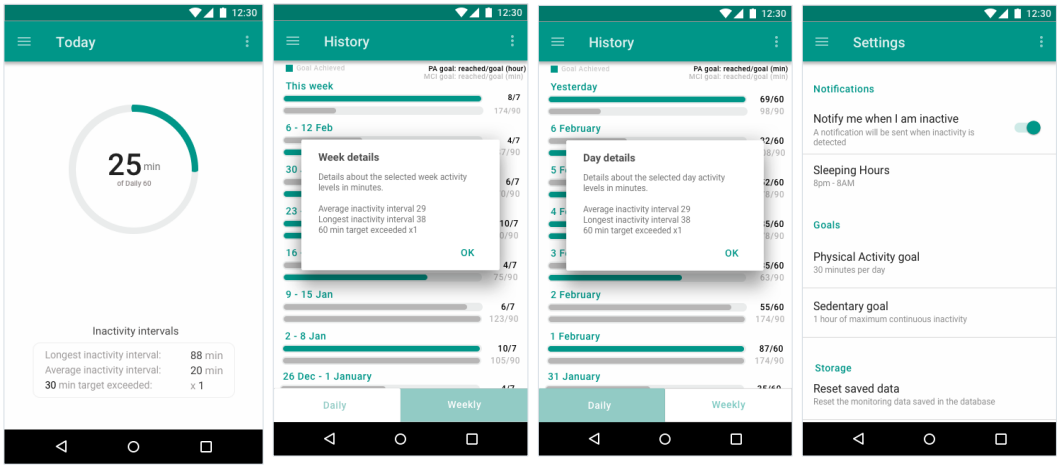


Database Design



5/2/2017

System UI



System demo

Demo of system using Android Studio's. Showcase how the process of registering a new user as well as all of the system screens

5/2/2017

Survey responses

No	Question	Response
1	What is your age	18 to 24 (100%)
2	Were you more active then usual (e.g. walking, running) as a result of using ActiveMinutes?	Yes (33.33%) No (33.33%) Not sure (33.33%)
3	When notified for prolonged inactivity (e.g. 30 minutes of inactivity), did you try to do at least 5 minutes of physical activity?	Yes (66. 67%) No (0%) Sometimes (33.33%)
4	Does seeing past days goal-performance (e.g. in the History screen) motivate you to achieve a goal?	Yes (100%) No (0%) Not sure (0%)
5	Does the visual feedback (i.e. the green progress bar) encourage you to achieve your goal?	Yes (100%) No (0%) Not sure (0%)

Survey responses (Continued)

No	Question	Response
6	Do you think the application was accurate when measuring your activity levels?	Yes (66. 67%) No (0%) Somewhat (33.33%)
7	How easy to understand was the user interface of the application? (0 being very difficult and 100 being very easy)	Response 1 – 100 Response 2 – 80 Response 3 - 75 **Average - 85**
8	Was the mobile application battery-friendly?	Yes (66. 67%) No (0%) Somewhat (33.33%)

5/2/2017

Survey responses (Continued)

1. "Great application, I like the feature that notifies you when you have been inactive for a certain amount of time. The application could maybe have a feature to alert you when you should go to bed to achieve the best amount of sleep. For future development, the application could maybe sync with a watch."

2. "Slightly wider ranges in terms of sleep time and active and inactive minutes and perhaps a more obvious "Key" informing the users as to what each bar in the history means. Otherwise really easy to use."

3. "The application could be improved in the future by optimising it to be more battery-friendly."

Future work

- Implement a personalisation of classifier logic
- Incorporate for social media sharing support
- Integrate Smartwatch sync support
- Improve the feedback process by adding a "End of the week goal-performance" reports
- Implemented the user feedback gathered via the project survey

Appendix P

Project Proposal

THIS PAGE INTENTIONALLY LEFT BLANK

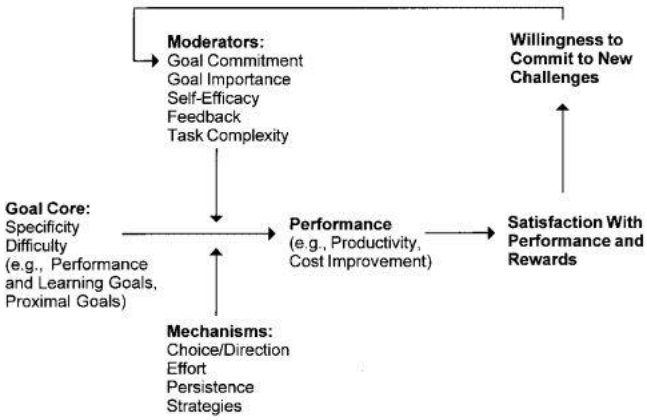
1.1 . Detailed research question/problem

How can Human Activity Recognition be implemented on a mobile device in order to help inactive people self-manage sedentary behaviour?

The aim of this project is to develop an application that will support inactive people change their behaviour and become more active. The application will facilitate behaviour change by providing the users with a self-management framework such as the one Locke and Latham (2002) suggested – (see figure 1). They (ibid.) showed that goal setting, monitoring and giving feedback are very important elements when it comes to motivating one's behaviour change. In this project the goals will be set by the user but informed by medical guidelines, the monitoring will be achieved by accelerometer sensor in a smartphone and real-time feedback of achievement via phone notifications will be shown to the user. In order to answer the research question, the following has to be undertaken: -

- Literature review to investigate current mobile device behaviour change approaches and further find out the current state-of-the-art in Human Activity Recognition (HAR)
- Design and build a system that will:
  - Gather accelerometer sensor data, extract features of the data, and classify and log user's activity as "active" or "static" over time – HAR component implementation.
  - Implement self-management component
  - Ensure usability by allowing the user to set goals and show notifications to the user when prolonged inactivity intervals are detected and show achievement accomplishments.
- Evaluate the approach by testing the proposed mobile application under real circumstances (e.g., the system will be evaluated by volunteers, and a questionnaire will be conducted to gather information about the effectiveness of the application).

Figure 1. Essential Elements of Goal-setting (Locke and Latham, 2002)



1.2 Keywords

Human Activity Recognition; machine learning; data mining; self-care; self-management

1.3 Project title

A Mobile Application that applies a Self-Managing Approach to Reducing Sedentary Behaviour

1.4 Client, Audience and Motivation:

Nowadays, with the rapid advance of technology people spend a lot of their time in a static position. In a statistical report Townsend et al. (2015) concluded that people in England spend their leisure time sitting- on average 4.9 hours and 4.7 hours per weekend day for men and women respectively. The majority of the adults in Great Britain, for example, spend a lot of their time watching TV, browsing the Internet or playing video games (Lader, Short, and Gershuny, 2005). The current shift of technology and work becoming more sedentary will potentially cause many health problems. According to Wilmot et al. (2012) spending prolonged periods of sedentary time increases the chances of all-cause mortality by 49%. In addition, people who lead non-active lifestyle are 112% more likely to get diabetes, 147% more likely to experience cardiovascular events, and 90% more likely to die due to cardiovascular events (ibid.). In order for a person to reduce the chances of developing the above diseases they should to follow the recommended levels of activity per day. According to the Department of Health (2011) and Townsend et al. (2015) the guidelines of activity per day is that adults (19-64 years) should do 30 minutes on at least five days a week. Parkinson (2016) and Siddique (2016) claim that an hour of activity per day would be ideal. Furthermore, lead scientist Professor Ulf Ekelund,

from Cambridge University and the Norwegian School of Sports Sciences, cited in Siddique (2016) said that every hour spent sitting should be broken by a five-minute break.

This project will develop a smartphone application that will help inactive people reduce sedentary behaviour by monitoring their inactivity during the day, notifying the user when extensive inactivity times are detected. What is more, the application will give feedback upon goal achievements when the user has accomplished the previously set goals in order to motivate and encourage activity. As part of the self-management logic of the application, the user would be able to set an exercise goal (e.g., 30 minutes of activity and maximum inactivity period of one hour per day) – thus trying to prevent or minimise the occurrences of diseases associated with sedentary lifestyle. The primary audience for the above application is the general public - anyone who is spending most of their time in a static position and being inactive in general.

1.5 Project Plan

Project plan for the project includes six stages – Literature review and application analysis, Gathering application requirements, Application design, Implementation, User Evaluation and Report preparation (see also Project Plan as a Gantt Chart in **figure 2**). As for the implementation stage, I have decided to follow the waterfall software development modal (see figure 3) since I do not have a particular client to exchange information/feedback. The waterfall software development modal perfectly fits the linear execution of my project (see the six stages bellow) as well as it is simple to understand. (S. L. Osmani et.al. 2013).

1. **Literature review and application analysis** – it includes researching:
   1.1 Self-management in the health care sector
   1.2 Human activity recognition
   1.3 Analysing existing mobile applications.
2. **Gathering application requirements** – includes:
   2.1 Identifying what the application will do in terms of functional and non-functional requirements.
3. **Application design** – includes:
   3.1 Defining use case scenarios to allow design of prototype user interface
   3.2 Defining the data storage and the application logic
4. **Implementation** – includes the following:
   4.1. Implementing HAR system
      4.1.1 Preparing documents such as participant's consent forms
      4.1.2 Collecting user data
      4.1.3 Training and exporting classifier
   4.2 Implementing self-management logic (e.g. recommended activity amounts according to health guidelines)
   4.3 Implementing application User Interface (UI)
5. **User evaluation** – the proposed application will be evaluated by real users. They will use the application for a whole day and will give feedback regarding how effective (if successfully recognised if they were active or not as well as if it reminded them when being static for a long time) the application has been throughout the day. The feedback will be gathered in the form of a questionnaire
6. **Report preparation –** gathering and summarising findings and results.
7. **Demo presentation** – prepare a demo version of the mobile application

ID		Task Mode	Task Name	Duration	Start	Finish
1			**Honours Project**	**154 days**	**September 30, 2016**	**May 3, 2017**
2			**Literature review and application analysis**	**1 mon**	**September 30, 2016**	**October 27, 2016**
3			Self-management in health care sector	1 wk	September 30, 2016	October 6, 2016
4			Human Activity Recognition	2 wks	October 7, 2016	October 20, 2016
5			Analysing existing mobile applications	1 wk	September 30, 2016	October 6, 2016
6			**Gathering Application Requirements**	**3 days**	**October 28, 2016**	**November 1, 2016**
7			Identifying what the application will do - functional and non-functional requirements	3 days	October 28, 2016	November 1, 2016
8			**Application Design**	**7 days**	**November 2, 2016**	**November 10, 2016**
9			Defining use case scenarios to allow design of prototype user interface	3 days	November 2, 2016	November 4, 2016
10			Defining the data storage and the application logic	3 days	November 7, 2016	November 9, 2016
11			**Implementation**	**40 days**	**November 13, 2016**	**January 5, 2017**
12			**Implementing HAR system**	**7 days**	**November 13, 2016**	**November 22, 2016**
13			Preparing documents such as participant's consent forms	1 day	November 13, 2016	November 13, 2016
14			Collecting user data	6 days	November 14, 2016	November 21, 2016
15			Training and exporting classifier	1 day	November 22, 2016	November 22, 2016
16			Implementing self-management logic	15 days	November 23, 2016	December 13, 2016
17			Implementing application User Interface (UI)	1 wk	December 14, 2016	December 20, 2016
18			User evaluation	1 wk	January 6, 2017	January 12, 2017
19			**Report preparation**	**2 wks**	**January 12, 2017**	**January 25, 2017**
20			Gathering and summarising findings and resul	2 wks	January 12, 2017	January 25, 2017

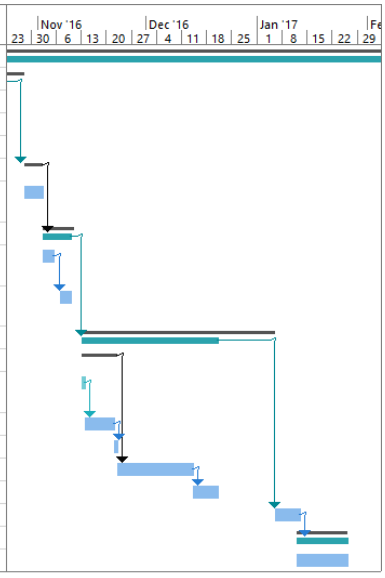


Figure 2. Gantt chart of the project plan.

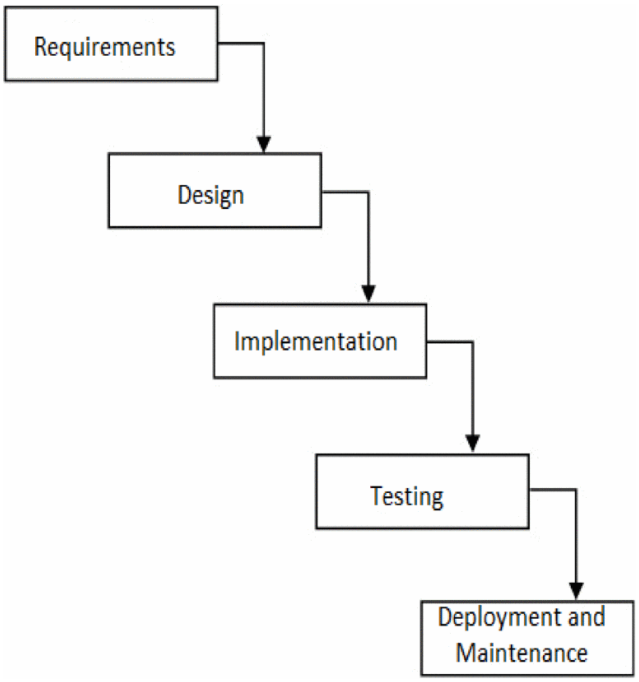


Figure 3. Waterfall Software development modal (S. L. Osmani et.al., 2013)

This is the end of section one.

2.1 Abstract

It is known that many people nowadays spend most of their time in front of a computer screen, both at work and in their leisure time. It is evident that a lot of time is spent in a static position (e.g., writing an article on a PC or watching TV). This means that if people do not follow the recommended physical activities guidelines they can easily develop health issues such as lower back pain, diabetes and cardiovascular diseases, in addition to having an increased chance of early age mortality. The aim of this project is to develop an application on the Android platform to help inactive people reduce the time they spend in a static position. The main objective of the project is to change user behaviour via 3 step process - (i) set goal (the user sets a goal – e.g., being active for 30 minutes per day), (ii) tracking and (iii) analysing goal achievements. Furthermore, the application will be designed so that when a prolonged inactivity interval is detected, it will notify the user via a notification in order for them to take action such as going for a walk. Another objective of the project is to design logic for personalising the classifier (previously trained from the accelerometer sensor data on the mobile device) to adapt to the user's movement patterns. The application will detect when the user is sleeping to prevent unnecessary notifications. The application's effectiveness will be evaluated by a small number of research participants. The evaluation will consist of research participants using the application for a certain amount of time and giving feedback via a questionnaire.

2.2 Initial/Mini Literature Review

One of the main challenges in this work is to develop an effective Human Activity Recognition (HAR) component that will support self-management logic for reducing sedentary behaviour. HAR has been and still is a hot research topic due to its many existing and potential application areas across a wide range of health and fitness applications (e.g. Google fit. no date, Apple Inc., 2016). There are many good examples of HAR systems implemented on mobile devices, for example Shoaib et al. (2015) proposed a HAR system that used both smartphone and smartwatch sensors to detect 'bad habits' such as smoking or not eating on time. This information could be used to gain insight into how people addicted to smoking progress over time.

One of the most important characteristics of a HAR system is to be accurate, unobtrusive and energy efficient. As far as accuracy is concerned, "*Vigilante*", a mobile application that implements HAR (Ó D. Lara and M. A. Labrador) achieves overall accuracy of 96.8% by utilising a C.45 decision tree classifier. The application uses a sensor strap attached on the user's chest to collect sensor data such as acceleration, heart rate and skin temperature. *ActiServ* (Berchtold et al., 2010) uses only the accelerometer sensor on a mobile device (NEO FreeRunner) to recognise activities with high accuracy. Compared to the system that Ó D. Lara et.al proposed, it achieves a slightly better recognition accuracy of 98%.

Another important characteristic of a HAR system is being unobtrusive to the user or not requiring too much interaction with the application (Labrador and Lara Yejas, 2013). "*Vigilante*" could be improved by eliminating the need for the user to wear a sensor strap on their chest and

using only the smartphones sensors – thus minimising its obtrusiveness. As for ActiServ, it is less obtrusive, compared to "*Vigilante*" due to the fact that it uses the built-in accelerometer sensor on the mobile device. On the other hand, it requires the user to complete several activities in order for the application to personalise its classifier which could introduce human error when the classifier is being personalised (e.g. incorrectly labelling of the data).

Least but not last a HAR system should be battery friendly, Kalic et.al. (no date) showed the Bluetooth communication consumes a considerable amount of energy - 10% per hour (for constant Bluetooth download) which means that "*Vigilante*" is spending valuable energy to send the raw data from the sensor strap to the phone. What is more, Parkka et. al. (2010) showed that when data is transferred wirelessly (e.g. via Bluetooth) there is a known percentage of samples that is lost - 0.01% to 2.02% (depending on the sampling rate). As for *ActiServ*, additional energy is spent making network calls to the application's server to select the best Activity Classification Module Set (ACMS). Kalic et. al. (no date) showed that Wi-Fi upload data transmissions could use a lot of smartphone's battery - about 4.2 hours (with constant Wi-Fi communication).

The system proposed in this paper will use only the accelerometer sensor on smartphone and does not require any external sensors (*Vigilante*) - thus its level of obtrusiveness is low. In addition, a personalised classifier will be trained without the requirement of the user to perform any necessary activities. This is done via selectively gathering newly labelled data and training a new classifier based on the new data. This autonomous process allows for minimising human error. The main purpose of the application proposed with this work is to identify prolonged inactivity intervals and to remind the user when those occur.

2.3 Relevant professional, social, ethical, security and legal issues to the project

The project has considered the following issues:

Professional

As a student studying CGA (Computing for Graphics and Animations) – a course that is accredited by the British Computer Society (BCS) I intend to comply with the BCS Code of Conduct. Specifically point 4 (d) (BCS Code of Conduct, 2016) - "act with integrity and respect in your professional relationships with all members of BCS…" I intend to give credits to the authors of any third party software I use in this project.

Social

The proposed application will try to avoid promoting any body image stereotypes by focusing more on encouraging the users to be more active rather than on their body characteristics such as weight and height or BMI (Body Mass Index). In addition, the application will only show what the recommended activity intervals per day are and will not force the users of the application to follow those strictly (some jobs do require seating for extended intervals of time – e.g. Pilots).

Ethical

The data collected during the project development such as names of the project participants and their accelerometer sensor data will be stored on disk only for the development purposes of the project.

Security

The mobile application, proposed in this work, will offer an authorization component to prevent other people accessing personal data such as total minutes of activity/inactivity collected daily. What is more, any relevant or valuable user data will be anonymised in order to protect individual's identity.

Legal

To prevent any harm done to the user, the mobile application will show a disclaimer (warning) dialog message to inform the user that they should be in a good physical condition (not suffering from diseases that could lead to worsening the condition of the user) before they use the application. In addition, I intend to fully comply with the terms of conditions of any third party software that I intend to use (my main goal is to use primarily Open Source software). As far as the project participant's data is concerned, I intend to keep the data only for the purposes of this project, and the data will not be used for any commercial purpose.

2.4 Bibliography (key texts for your literature review)

Apple Inc. (2016) *IOS - health*. Available at: http://www.apple.com/uk/ios/health/ (Accessed: 14 October 2016).

Berchtold, M., Budde, M., Gordon, D. and Beigl, M. (2010) 'ActiServ: Activity Recognition Service for mobile phones', Seoul, Korea: Karlsruhe Institute of Technology. pp. 1–8.

BCS Code of Conduct (2016) Available at: http://www.bcs.org/category/6030#profession (Accessed: 18 October 2016).

Shoaib, M., Bosch, S., Scholten, H., Havinga, P.J.M. and Incel, O.D. (2015) 'Towards detection of bad habits by fusing smartphone and smartwatch sensors', *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, . doi: 10.1109/percomw.2015.7134104.

Siddique, H. (2016) *One hour of activity needed to offset harmful effects of sitting at a desk*. Available at: https://www.theguardian.com/lifeandstyle/2016/jul/27/health-risk-one-hour-activity-offset-eight-hours-sitting-desk (Accessed: 17 October 2016).

S. L. Osmani, G. Arifi and F. Idrizi, "Choosing the Most Suitable Model for Developing a Software," *Computational Intelligence, Communication Systems and Networks (CICSyN), 2014 Sixth International Conference on*, Tetova, 2014, pp. 83-88

Google fit (no date) Available at: https://www.google.com/fit/ (Accessed: 14 October 2016).

Kalic, G., Bojic, I. and Kusek, M. (no date) *Energy consumption in Android phones when using wireless communication technologies*. Available at: http://ieeexplore.ieee.org.ezproxy.rgu.ac.uk/document/6240745/ (Accessed: 14 October (Accessed: 14 October 2016).

Labrador, M. and Lara Yejas, O. (2013) 'Human Activity Recognition', in *Human activity recognition: Using wearable sensors and smartphones*. Boca Raton: Taylor & Francis Group, pp. 14–14.

Lader, D., Short, S. and Gershuny, J. (2005) *The time use survey, 2005 how we spend our time A report on research using the ONS omnibus survey produced on behalf of the economic and social research council (ESRC), department of culture, media and sport (DCMS), department for education and skills*. Available at: http://www.timeuse.org/sites/ctur/files/public/ctur_report/1905/lader_short_and_gershuny_2005_kight_diary.pdf (Accessed: 10 October 2016).

Locke, E.A. and Latham, G.P. (2002) 'Building a practically useful theory of goal setting and task motivation: A 35-year odyssey', *American Psychologist*, 57(9), pp. 705–717. doi: 10.1037//0003-066x.57.9.705.

Parkinson, C. (2016) *Hour's activity 'offsets sedentary day'*. Available at: http://www.bbc.co.uk/news/health-36895789 (Accessed: 17 October 2016).

Pärkkä, J., Cluitmans, L. and Ermes, M. (2010) 'Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree', IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society., 14(5), pp. 1211–1215.

Ó D. Lara and M. A. Labrador, "A mobile platform for real-time human activity recognition," *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2012, pp. 667-671.

The Department of Health (2011) *Start active, stay active A report on physical activity for health from the four home countries' chief medical officers.* Available at: http://www.bhfactive.org.uk/userfiles/Documents/startactivestayactive.pdf (Accessed: 10 October 2016).

Townsend, N., Wickramasinghe, K., Williams, J., Bhatnagar, P. and Rayner, M. (2015) Physical Activity Statistics 2015. London: British Heart Foundation.

Wilmot, E.G., Edwardson, C.L., Achana, F.A., Davies, M.J., Gorely, T., Gray, L.J., Khunti, K., Yates, T. and Biddle, S.J.H. (2012) 'Sedentary time in adults and the association with diabetes, cardiovascular disease and death: systematic review and meta-analysis', 55(11), pp. 2895–2905. doi: 10.1007/s00125-012-2677-z.