Group 08 FallPrevention

Vassdal, Johannes Willumsen
johannes.vassdal@gmail.com

Aamot, Elias
eliasaa@stud.ntnu.no

Larsen Tomren,Filip Andre
filip@tomren.it

Pham,Dat-Danny
datdanny@hotmail.com

Kücükyareli,Tayfun
tayfun.kucukyareli@live.de

April 16, 2013

Contents

Chapter 1

Introduction

Damage from falling is the main cause for hospitalization among the elderly. In addition to the enormous costs incurred to society, falling is also the cause of great personal tragedy. As the use of computational devices becomes more common, it seems natural to examine how such devices can contribute to preventing such accidents.

This project is a cooperation between SINTEF and a group of students at NTNU for the course IT2901. Its goal is to develop a model for risk level of movement based on common sensors found in Android smart phones, and build an API to make the model accessible for third-party developers.

1.1 Group introduction

The group consists of 5 members, four of which were from Norwegian University of Science and Technology, and the remaining student was an exchange student from Germany, Technical University of Dortmund. The members had experience with programming and project development with the programming languages Java and C, but little experience with Android applications.

1.2 Customer introduction

The customer was ”Selskapet for INdustriell og TEknisk Forskning ved norges tekniske hoegskole”, or SINTEF for short. The person representing SINTEF was Babak A. Farshchian, Adjunct Associate Professor at NTNU.

Chapter 2

Requirements

The requirements for the different parts of the project was about the requirements for the application, the underlying model and content provider, and the documentation for the model and content provider.

2.1 Final requirements

Several meetings with the customer and health experts led the initial requirements to be out of date and a little vague which led to new and updated requirements.

2.1.1 Functional requirements

- An Android content provider that stores and makes the data available for other applications through an API.
  - Measure gait speed.
  - Measure gait variability.
  - Measure steps.
  - Filter out noise in terms of bad and unnatural movements.
- An example application that can visualize this data and represent them in a pedagogical manner.

  - Ability to do self-tests
    - Reaction speed
    - Stand up sit down capability
    - Balance tests

2.1.2 Nonfunctional requirements

- 80% of the elderly should manage to use the app within 10 minutes of education given that they have some prior knowledge of smartphones.
- Midly visually impaired should be able to use the application.
- Color blind should be able to use the application.

2.2 Understanding the requirements

The requirements were specified after several meetings with the customer. The understanding the group had of the requirements before meeting the customer was not sufficient to create a set of requirements. The requirements were also subject to change as the project progressed.

The meetings served their purpose, and filled in a lot of holes that was missing from the preliminary description:

- The development process in question uses a iterative approach. This essentially means that the requirements expands as the time goes, and to define the requirements from the beginning is impossible. This decision was made in collaboration with the customer.
- The team at SINTEF got a wide range of experts to help the group with health-specific features, e.g. making the algorithms to recognize movements.
- The customer expected weekly meetings with the whole group present, and that goals had been achieved before each meeting. At each meeting a new goal was set.

2.3 Initial requirements

According to the project description, the following should be developed:

- A model of physical movements based on common movement sensors found in Android smart phones.
- An Android content provider that stores and makes available the data in this model through an Application Programming Interface(API).
- An example application that can visualize this data.

2.4 Testing

The tests were done by the black-box principle of testing, meaning that all the tests consider is, with no attention given to the inner workings of the methods. Because of the team following an agile development method, it was possible to create and test test-cases in short order. The test-cases were therefore written with results included. The tests are corresponding screens for the tests were as follows:

Name Entry is associated with tests 1.x, both when program is started the first time, and when name-change is initiated from settings.

Event Details Screen is associated with tests 2.x.

Statistics Screen is associated with tests 3.x.

Clear history can be initiated from settings screen and is associated with tests 4.x.

Related people Screen is associated with tests 5.x.

Figure 2.1: List of tests for the application

Test number	Input value	Actual Result	Expected Result
1.1	Any valid name	Hello: + name	Hello: + name
1.2	Empty name-box	Hello:	Hello:
2.1	Remove event button clicked	Event removed	Event removed
2.2	Keep event button clicked	Event is kept	Event is kept
3.1	Timeslot set to 10 min or less	Data from correct timeslot displayed	Data from correct timeslot displayed
4.1	Clear History Button Clicked	History Cleared	History Cleared
5.1	Add existing contact button clicked	Contact shows up on appropriate screen	Contact shows up on appropriate screen
5.2	Create New Contact function used	New Contact shows up on appropriate screen	New Contact shows up on appropriate screen

Chapter 3

Testing

3.1 Testdata

Prototype test Subjects: Beatrice & Jorunn
This test was mainly focused on intuitive it is to navigate the application system and how easy it is to understand the data received by the system.

Test result: The data received from the system was easy to spot and easy to understand. The subjects had some difficulty finding the menu button for the first time, but had no problems with using it later on. It was overall relatively easy to navigate the system.

Chapter 4

Project Management

4.1 Terms

Here follows a description of terms that are useful to understand how the project management functioned.

Scrum is an popular agile software development methodology. This means that the work is done in increments called sprints, and daily meetings are held to update the rest of the team on progress and problems. The scrum process also has roles for team members, in particular a team member called scrum master who handles contact between the development team and people outside the team.

Sprint is a period in which work is planned and done. It has a duration of one week. At the end of each sprint the group is updated on progress achieved, and sets goals for the next sprint.

Kanban-board is a visual aid that was used to display work packages, with content, people assigned to each package, and status of work package (to do, currently worked on, finished). The intent was to simplify and organize the work being done and work in need of finishing.

4.2 Development process

The customer favored the use of an iterative approach to the development process, where every sprint added a new layer of functionality, either to the application or the underlying model. Each sprint lasted for a period of 1-2 weeks, and the exact content was worked out in collaboration with the customer. Short term plans were favored over longer plans, due to the flexibility provided. While this made formulating definite goals for the final product difficult, the customer and the group were in agreement that due to the research intensive nature of the project, a high degree of flexibility was required.

It was decided that the developers would have online meetings twice a week and an offline meeting once a week. The working hours were set to not less than 20 hours a week, but the developers were free to choose when to work themselves. This number was the expected workload for the course, and made a reasonable target for work per week. The meetings with the team was set to two times per week instead of once per day, so it could fit with the schedules of the team members. This was different from other agile methods like Scrum, but if was a necessary adaptation to fit meetings into the entire teams schedule. Updates were shared among the team members with email, trello[1], and it's learning as a replacement for daily meetings.

4.3 Plans

Nearing the end of every sprint, the customer and group agreed upon the content of the following sprint. This list has been placed in Figure A.2, and provided a short summary of what was accomplished in a particular time-frame.

[1] See 12 for more info

4.4 Team Roles and Organization

There was not much place for specific roles among the group, as it was a small group, and the project required that all the members were capable and willing to work at all the tasks. This makes a difference from development models with specific roles and tasks assigned. Roles that were set for the group were therefore mainly organizers, so that one person was to keep awareness of what work needed to be completed in a particular domain, and share it with the rest of the group:

Group Leader Elias was made organizer, and got the responsibilities of reminding the group of what to do.

Document-organizer Johannes was tasked with organizing documentation and distributing the work of writing the report to the group.

4.5 Comparison

The development model used by the group was a form of agile development. Other well-known examples of agile methodologies were scrum and extreme programming.

4.6 Management Tools

Trello was a collaboration tool with the ability to create interactive kanban-boards online. This use of this tool was to allow the group to coordinate tasks that were to be done, and the progress on the tasks, and which members were to work on which task.

It's learning was used to distribute information that was not time -critical, with a message board being used. It's learning would not send messages when a new topic or message appeared, so it's use for time-critical messages or making sure that everyone would read it was limited.

Email was used for time-critical communication, and for information that needed feedback swiftly, often within the same day.

Github was used to share code, and to describe and mark issues found in the code when problems were discovered. The relevant issues could then be discussed on the website

4.7 Work Breakdown Structure

With a development methodology that stresses short term goals, it is only possible to plan WBS for the current period. After the customer has specified the goal for the following week, we immediately sat down and defined the WBS for the given period. This was, a WBS was developed incrementally by adjoining the new WBS to the previous. The WBS was made in graphical form, as can be seen in Figure 4.7.

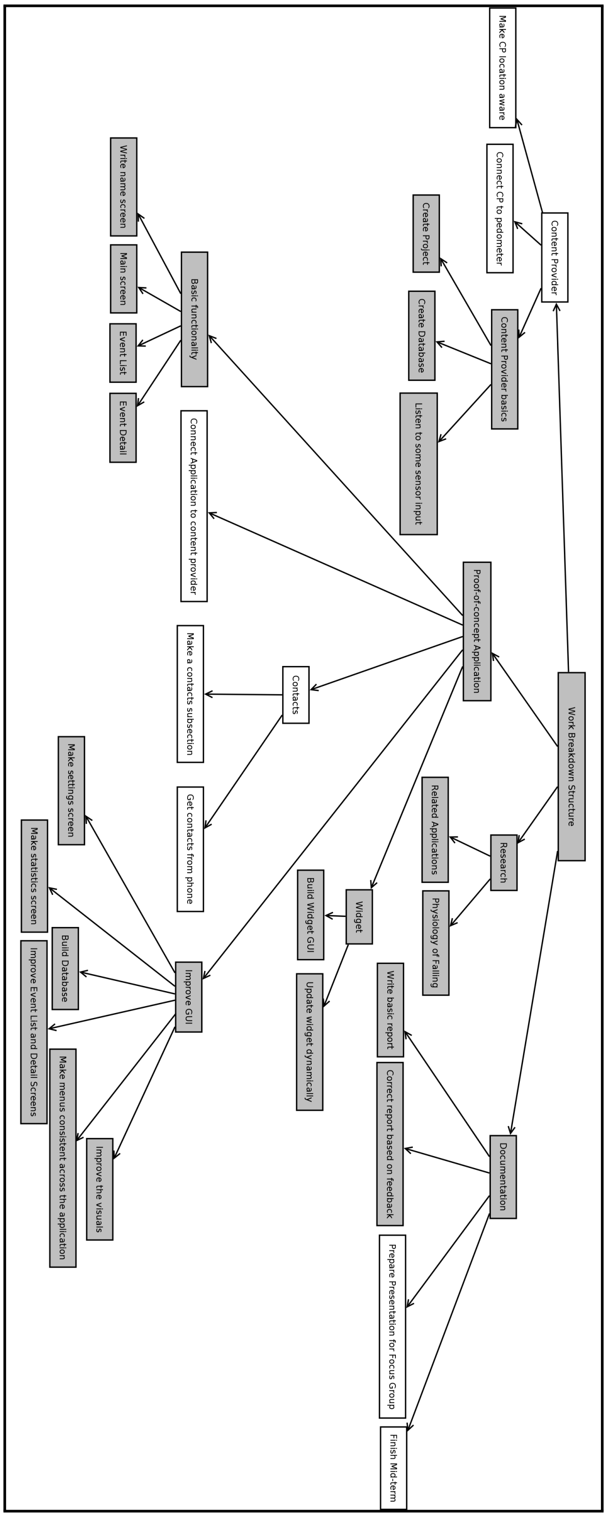

Figure 4.1: The current WBS

Chapter 5

Development Environment

5.1 LaTeX

LaTeX was used to write this report. LaTeX is a typesetting language, with support for varied formatting, including images and including other document. It also has the option allowed for inclusion of multiple LaTeX documents (.tex documents). This leads to improved readability of LaTeX code and productivity, as several chapters could be altered at the same time.This functionality allows a group of multiple users to avoid the problems associated with multiple users writing in the same document.

5.2 Github

It was requested by the customer that the group use the tool Github to share code and perform version control. Github had browser-based interfaces, download-able clients, and a robust command-line interface, meaning all the members of the group could make use of it. There were some problems associated with learning how to use and fix problems with it, as not all group members had experience with these tools.

5.3 Eclipse with Android Development Tools Plugin

The demands from the Integrated Development Enviroment(IDE) was as following:

- Had support for programming applications for Android
- Was understood by at least some members of the team

To fill these requirements, and because there was plenty of tutorials that could be found, the group decided to use the Eclipse IDE, with add-on's to more easily code and deploy towards Android.

Chapter 6

Implementation

6.1 Overall Architecture

The product has been implement as a system of independent component applications where each component performs a well-defined task. The motivation behind the domular architecture is two-fold: Primarily, it was desirable that the underlying model for movement and fall risk be independent of the proof-of-concept application, in order to facilite development of possible future applications employing the model. Secondly, the standards of Android application development state that a Content Provider should in itself solely provide a interface to the data. This implies that the sensor model, as well as any derivation of secondary data, should be performed in components independent of the Content Provider. [ADD INFO ABOUT INDEPENDENCE SENSOR MODEL / CP AS WELL AS INDEPENDENCE CP / GUI?]

The system that has been designed according to these principles therefore consists of five components: The content provider emphValens Content Provider, the proof-of-concept application emphValens Health Helper, the sensor model emphValens Step Detector and a service of deriving secondary data emphValens Content Feeder. Each of these components fulfills a well-defined role in the total application system: The content provider provides an interface to the data model. The sensor model listens to the sensor data, and feeds the content provider with the timestamps of any detected steps.

The Content Feeder listens continually to changes to the data in the content provider. When the data changes, it uses the new data to calculate secondary data, such as gait speed and variability. Finally, the proof-of-concept application demonstrates one simple way in which the data provided by the content provider can be used as a part of a health-promoting app.

6.2 Architecture of the App

The application is made in a way that is common for all android applications. This means that user interface is described in xml layout files that is called in java code. Strings and resources is placed in a separate folder and file, to be accessed by the code as needed. This is to separate content and layout in the UI. The separation of layout and strings enables different localizations, so that the application can provide a user interface in different languages easily.

As is common in android applications, classes that inherit from the class Activity define a separate screen in the GUI. Because of this, a significant part of all the classes in the application are activity classes. These classes simply define the behavior of their GUI screen. The flow for the GUI can be seen in Figure 6.2

Classes that are not activities can be roughly divided into 4 types: Data structure classes, auxiliary classes, connectivity classes, widget classes. Data structure classes simply define a useful data structure in the program. The data structure classes in the app are Event, ContactPerson and RiskStatus. Auxiliary classes help the activities perform their tasks. Most common among these are the adapters that define how elements should be shown in a list view. Connectivity classes are responsible for communicating with the database and content providers. There are two connectivity classes: DatabaseHelper, which does all the work, and DatabaseContract, which specifies the table and column names in the database. While a DatabaseContract might seem redundant, it is common android application practice to use one. Finally, there are the widget classes. WidgetProvider creates the widget and gives it a GUI. WidgetUpdateService is started by the WidgetProvider

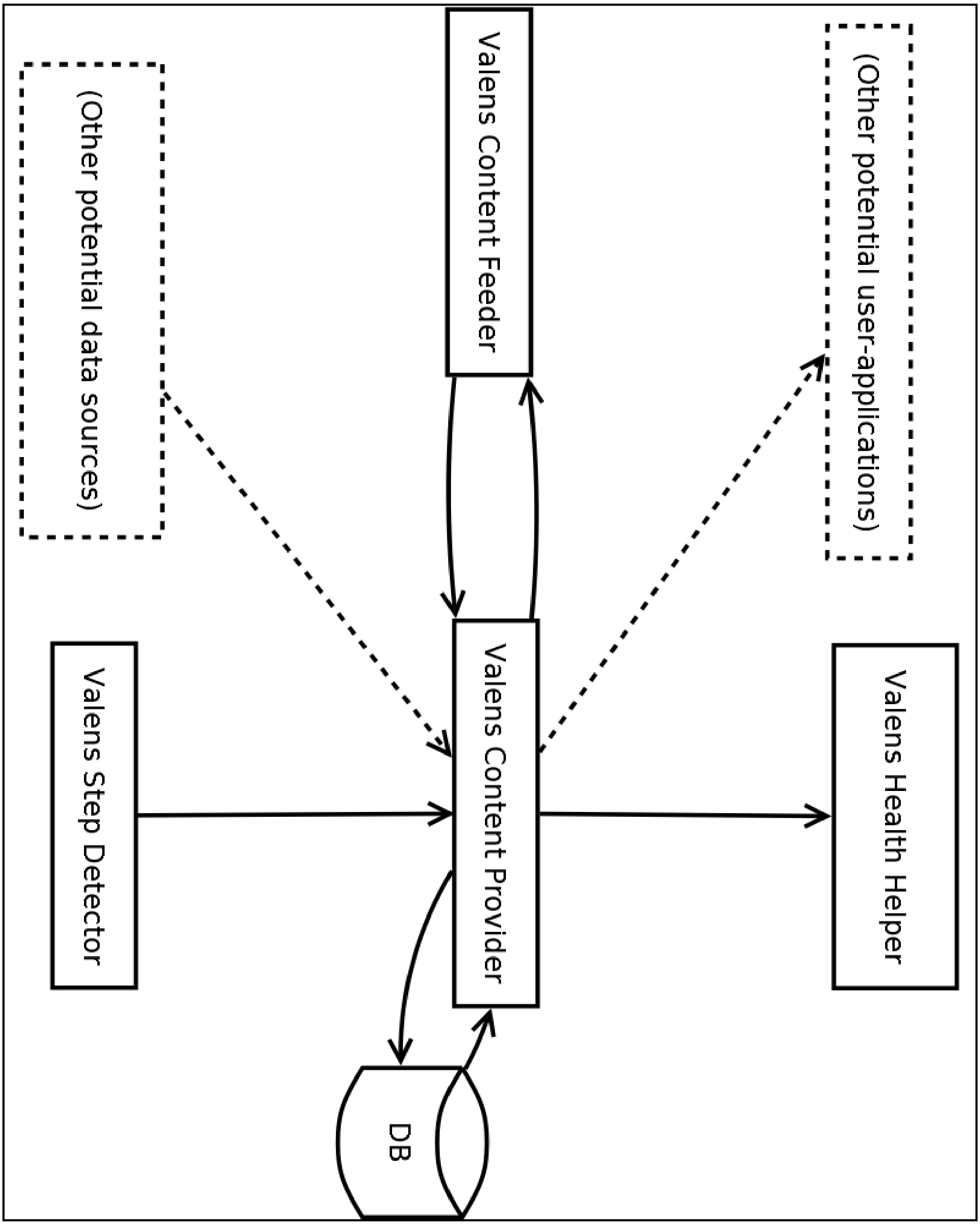

Figure 6.1: The over-all system architecture

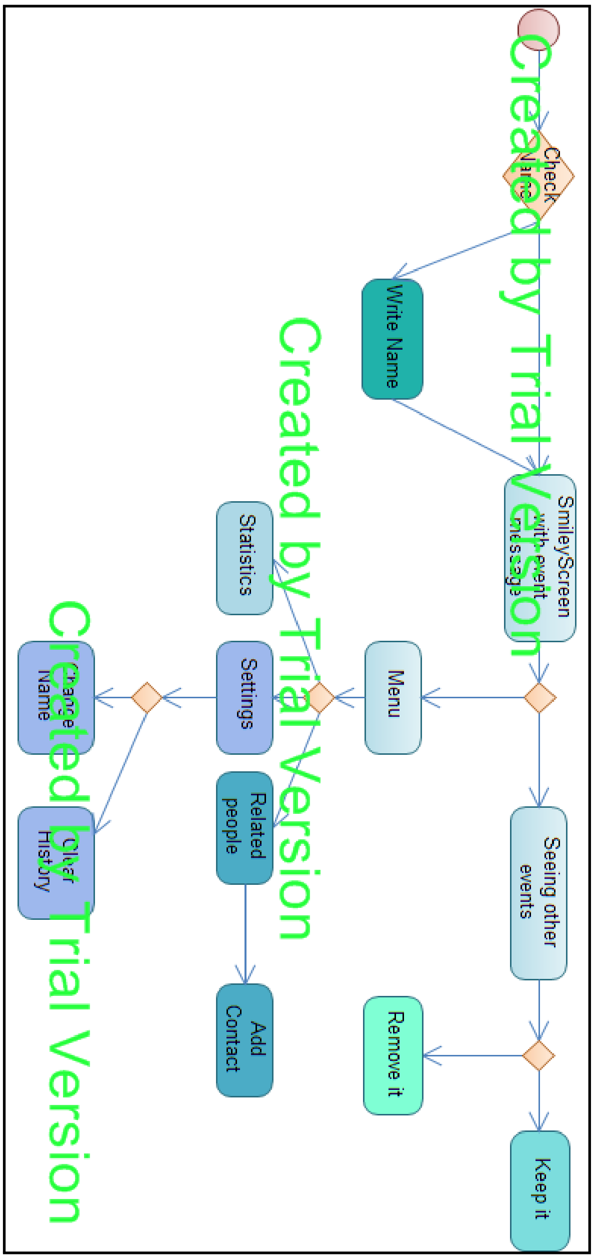

Figure 6.2: Flow chart describing the Graphical User Interface

and takes care of updating the widget when changes happen in the content provider.

6.3 Architecture of the Content Provider

In android, Content Providers provide an interface to structured data of some sort. Access to for instance the list of contacts in the phone, or the phone's calendar is managed through standard content providers. Content Providers provide methods for inserting, updating and deleting relevant data. A major part of the assignment was to implement a Content Provider that gives developers access to structured movement data.

The Content Provider component is conceptually very simple, and consists of four classes:

CPValensDB a subclass of SQLiteOpenHelper, the standard android class for handling database communication. Specifies the procedure for creating the database, as well as upgrading and downgrading the database version. In the current implementation, upgrading and downgrading the database version clears all the data and tables, and creates the database according the details specified in the new database version. In the current version, creating the database consists of creating the tables given in the data model, without feeding any data into the database.

DBSchema defines the database model and provides string values for the names of the tables and fields. Employing a database schema to demonstrate the structure of the database is standard in android applications.

ValensDataProvider a subclass of ContentProvider, which is the standard android class for implementing content providers. This is the core of the content provider, and describes how queries to the content provider are handled.

Main provdides the content provider with a basic GUI.

6.4 Class description and diagram

The class structure of the Applications changed regularly, and the most recent description is in Figure 6.4. Older diagrams and descriptions was placed in the appendix.

6.5 Architecture of the Valens Step Detector

Despite being a relatively simple application in terms of lines of code and GUI screens, the architecture of Valens Step Detector has a certain complexity.

6.6 Research reports

The reports in this section represent research and information-gathering done by the group when it was thought necessary.

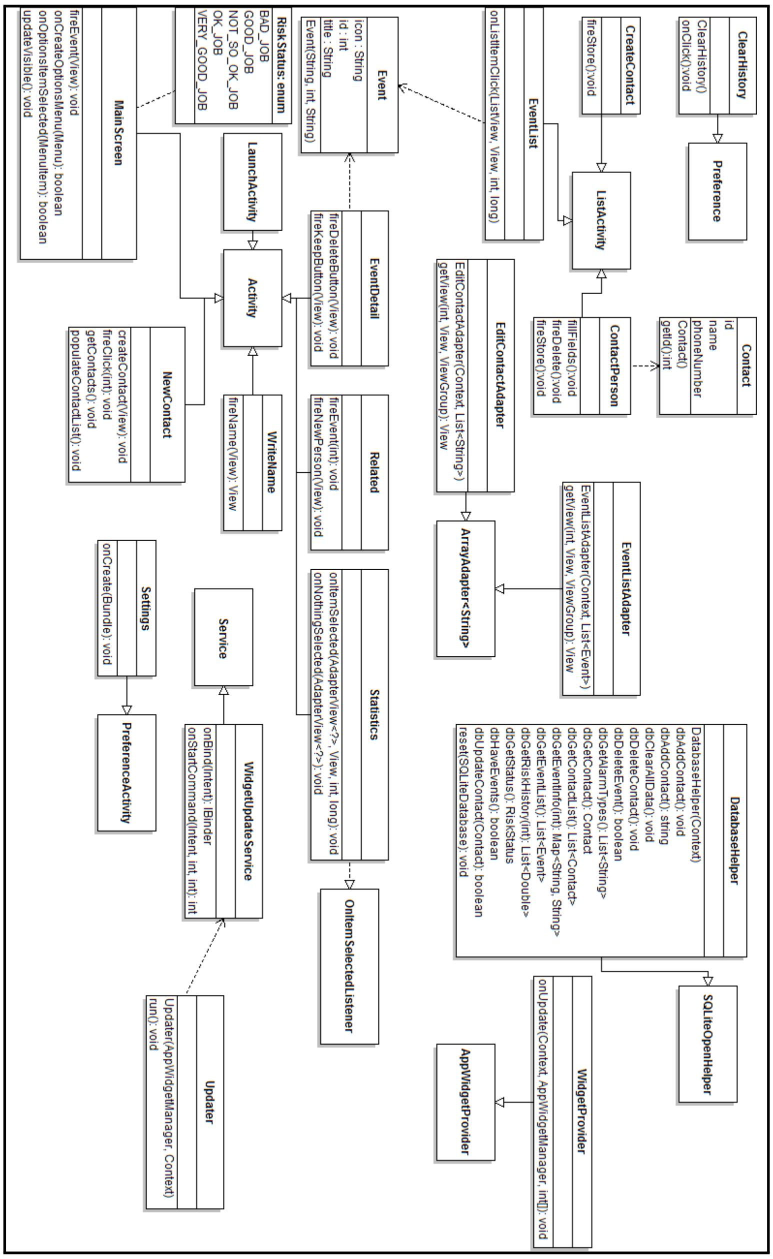

Figure 6.3: A class diagram of the current application

Chapter 7

Research and Alternate Solutions

The group had to do research on real-world problems and solutions to similar problems. The reports produced by the group will be reproduced in this chapter.

7.0.1 Falling: Causes, Consequences and how to avoid it

This report collected information from several sources on the subject of falls among the elderly. Particular focus is given to risks and avoidance strategies.

Risk factors

There are several risk factors associated with falling. Typical risk factors are:

- **Biological risk factors (sorted by order of significance in [1]):**
  - Muscle weakness
  - Balance deficit
  - Gait deficit
  - Visual deficit

  - Mobility limitation
  - Cognitive impairment
  - Impaired functional status
  - Postural hypertension

- **Behavioral risk factors(Not in sorted order):**
  - Inactivity
  - Side effects from medication
  - Alcohol use
  - Living alone
  - Staying at home

- **Home/environmental risk factors(not in sorted order)**
  - Dangers in the house or in public places.
  - Unfamiliarity with walking aids such as canes, crutches, walking chairs.

A fall is normally caused by the interaction of two or more of these risk factors. Home or environmental risk factors are involved in less than half of all falls. What this means is that more than half of all falls happen without the influence of environmental factors. The second and third most common single causes of falling are gait/balance disorders and dizziness/vertigo, followed by drop attacks, sudden falls without loss of consciousness or dizziness. Many older adults are unaware of their risk factors, and therefore unable to take preventive actions. Even older adults with a history of falling have been given little education about the potential risk factors. Risk assessment, even self-administered, can be very beneficial, especially when the results are discussed with a healthcare provider. It has been showed that many of the biological risk factors can be reduced by effectively. Specifically strength, gait and balance has been shown to be improvable in a number of studies. 60 % of falls happen in or around the house. While this sounds like a lot, older adults spend a lot of time at home, so one would expect this number to be higher.

Avoiding falls

One can easily reduce the risk of falling greatly by taking certain measures. A list of measures normally recommended by the authorities is given below:

- Begin a regular exercise program.
  - Exercises that improve balance and coordination are the most helpful.
  - Is the only measure that by itself reduces the risk of falling independently of individual circumstances.
- Have your health care provider review your medicines, even over-the-counter ones.
  - Avoid medicines that can make you dizzy or sleepy.
- Have your vision checked.
- Make your home safer.
  - Remove or fasten small rugs and carpets.
  - Remove wires and cords away from commonly taken paths. Tape wires to the walls, and if necessary install an additional power outlet.
  - Keep items where you can reach them without having to climb. If you have to use a step stool, get a stable one with handrails.
  - Remove loose items that can make you trip (books, papers, clothes, etc.) from the floor and stairs.
  - Add grab bars and non-slip mats to the bathroom.
  - Make your home brighter.
  - Repair broken or uneven steps and handrails in the staircase.
  - Avoid using the staircase more than necessary, for instance by installing a light switch at the top as well as the bottom of the stairs.

- Miscellaneous:
    - Wear shoes, even inside. Avoid walking barefoot or wearing slippers.
    - Get up slowly after you sit or lie down.
    - Keep yourself updated on your own medical conditions.
    - Learn how to properly use your walking aids.
    - Keep a healthy diet, drink enough water. Dehydration and malnutrition weaken your reflexes and attention.

Medical personnel are recommended by researchers to prevent falling by treating medical conditions that cause falls, as well as reducing the risk factors in the individual. When a fall happens, medical personnel is advised to spend time inquiring the details related to the accident in order to understand the causes of falling for the individual, and then try to counteract these risks.

Physiological aspects

The list below explains the physiological factors that contribute to stability. "A marked deficit in any one of these factors may be sufficient to increase the risk of falling; however, a combination of mild or moderate impairments in multiple physiological domains also may increase the risk of falling. By directly assessing an individual's physiological abilities, intervention strategies can be implemented to target areas of deficit." [2]

- Reaction time
    - Hand
    - Foot
- Vision
    - Contrast sensitivity
    - Visual acuity

- Vestibular function
    - Visual field dependence
- Peripheral sensation
    - Tactile sensitivity
    - Vibration sense
    - Proprioception
- Muscle force
    - Knee flexion
    - Knee extension
    - Ankle dorsiflection

There exists an array of tests that can measure the performance on these aspects. Lord et al. [2] provides one possible set of tests. Some of these are likely possible to implement as a part of our application, but none of them are based on hip movement. The test developed by Lord et al. [2] was used to classify older adults into fallers or non-fallers, and has an accuracy of 75 - 80 % in different experiments. If a test which disregards hip movement patterns performs well, it indicates that hip movement is only a minor factor in detecting fall risk.

Walking behaviour seems to be generally associated with falling. Lord et al. [3] shows that there is a negative correlation between falling and steps per minute, stride length and stride velocity. A positive correlation was shown between falling and stance duration and stance percentage. However, these physiological traits are all associated with old age, and old age is associated with falling, so the relation might be quite indirect.

“In old age, the ‘strategy’ for maintaining balance after a slip shifts from the rapid correcting ‘hip strategy’ (fall avoidance through weight shifts at the hip) to the ‘step strategy’ (fall avoidance via a rapid step) to total loss of ability to correct in time to prevent a fall” [1].

7.0.2 Summary of interview with medical professionals

This is a summary of the information gathered by a meeting with two medical professionals and the customer. **Present:** All group members, Babak Farshchian, Jorunn L. Helbostad, Beatrice Vereijken
Time: 14.15-15.37
Place: St.Olavs Hospital
The plan was to have a short presentation, a short GUI demonstration, and then discuss and interview with the interviewees.

Feedback on Interface and Presentation of the program

- Immediate and summarized feedback was desired, as it would be more helpful than feedback later.
- A focus on falling is not a selling point, try health or lifestyle instead
- Menu is not obvious to people without android experience
- Interface seems too complex for target groups
- Text might be too small
- Graph needs more contrast

Feedback on program functionality

- Feedback should be tailored to particular groups
- Self-testing can be useful for reducing risk and keeping awareness,
- Small tests to measure reaction time, for example reaction to light or sound
- Researchers could also be interested in data gathering
- Measuring transitions between sitting/lying and standing up

- Compare behavior on a weekly basis will give an overview of whether things are good or not

Feedback on medicinal stuff

- Preventing inactivity over time is useful to reduce risk
- People with stability problems increase risk when moving much
- Variable gait pattern is useful for predicting risk
- Step time is more robust (and hopefully easy to measure)
- A common exercise is to take a step forwards, sideways, backwards, sideways (measuring a box)
- People in risk group are those who cut out walking, taking the bus, want help from others around the house
- Irregularity in speed and length is important to look for
- Appropriate movement amount for gender and age group is usually known
- Time needed to turn in place

7.1 Alternate solutions

There were several other open source programs and commercial programs used to solve related problems. In some cases the team could import and make use of the open source programs, and in others the team could be inspired by design and well-made interfaces and functions. Here is that report, as written:

7.1.1 Endomondo Sports Tracker

This application is used to track routes, times and progress in training and activities. It uses GPS (and accelerometer) to find position on maps and track the location. Very easy to use, and automatically syncs up to endomondo.com. Has an inspiring design and sync-functionality. The main part of Endomondo is the inspirational part to get people to move; you can compare yourself to others, e.g. friends, and you have a really easy way to see what an exercise is worth in amount of calories and amount of burgers burned by an exercise.

A lot of aspects from this app can be used to design our own app in terms of labeling and representation of data.

7.1.2 GPS Status

This application is used to view detailed status about the GPS system on the phone. Has a very advanced interface with lots of numbers to show information, but for a rookie user the data rarely translates into information.

The app succeeds to use the sensors very heavily.

7.1.3 Pedometer

[1] An open source app that has a GPLv3 license that is compatible with our Apache 2.0 license. The app uses the same sensors that we should focus on and opens up the opportunity for us to take advantage of it and skip leaps in the "how to do this on Android"-process. We can freely use code from this software that already works.

From this we can use very much. **How does the step detection algorithm work?** Basically, it aggregates the sensor values, finds the maximum and minimum, and if the difference is bigger than a value (which depends on the sensitivity setting) then it counts it as a step. There's is some additional optimization, which I arrived to through experimentation.

[1] application can be found at https://code.google.com/p/pedometer/

7.1.4 GPS Tracker

[2] This program adds the capability to store and review where you and your Android device have been. Basically you press record at the start of your trip and your phone stores the route you take. This route is drawn real-time on the Maps functionality of Android or in the background with an idle device. The route is stored on your phone for review and further use. The applications tracks location by GPS, hence the name. An accurate description of the program type would be a GPS logger.

It contains the ability to log location, something that could be useful for our application. Since it is open source and has a compatible license (GNU v3), we can use large part of it.

[2] application can be found at: https://code.google.com/p/open-gpstracker

Chapter 8

Step Detection Algorithm

8.1 Development Process

As the group had little experience in sensor use and signal processing, a series of experiments were conducted to better understand which sensors provide useful information for step detection, and which patterns predicate steps. To this end, a simple app was developed to observe sensor input and store the raw data as a file, which then could be transferred to a computer. A set of python scripts were written to plot the data time series. Using python, the group experimented with different step detection algorithms, and a prototype was developed. Finally, the algorithm was implemented in the Valens Step Detector app, modified to work with live data.

8.2 The Algorithm

The current section will explain the algorithm used for step detection, formulated as an off-line algorithm. The following section will explain what changes was made to make the algorithm work for live data streams.

Experiments showed that the accelerometer consistently gave the most predicative feedback patterns for step detection[1]. Furthermore, it was found

[1] None of the android phones available to the group at the time had a working gyroscope. It is suspected that a gyroscope can replace or supplement the accelerometer for more precise step detection, but this should tested empirically.

that the total acceleration was a better predicator of steps than acceleration along any of the single axes individually. To exploit this, a pre-processing step calculates the a vector length in euclidian space of the raw acceleration vector.

Detecting steps based on a time series of accelation vector lengths reduces to peak detection in the time series graph. The step detection algorithm is therefore essentially a generic algorithm for peak detection in noisy data. The noise in the data originates from noise in the sensor readings. To cope with the noise, the data is first smoothed using a moving average[2].

Subsequently the program runs the main peak detection algorithm, which consists of the following steps:

Calculate peak strength

Every data point in the time series has its *peak strength* calculated. The peak strength for a point x_i is calculated as $\frac{(x_i - x_{i-1} + x_i - x_{i-1} + \ldots + x_i - x_{i-k})}{k}$

SKRIV OM KORLEIS EIN FINN VERDIANE SOM WINDOW_SIZE etc. a

[2] That is, every datapoint has its value set to the average of itself and the x neighbours before and after it, where x is called the *window size* of the smoothing.

Chapter 9

Step Detection Algorithm

9.1 Development Process

As the group had little experience in sensor use and signal processing, a series of experiments were conducted to better understand which sensors provide useful information for step detection, and which patterns predicate steps. To this end, a simple app was developed to observe sensor input and store the raw data as a file, which then could be transferred to a computer. A set of python scripts were written to plot the data time series. Using python, the group experimented with different step detection algorithms, and a prototype was developed. Finally, the algorithm was implemented in the Valens Step Detector app, modified to work with live data.

9.2 The Algorithm

The current section will explain the algorithm used for step detection, formulated as an off-line algorithm. The following section will explain what changes was made to make the algorithm work for live data streams.

Experiments showed that the accelerometer consistently gave the most predicative feedback patterns for step detection[1]. Furthermore, it was found

[1] None of the android phones available to the group at the time had a working gyroscope. It is suspected that a gyroscope can replace or supplement the accelerometer for more precise step detection, but this should tested empirically.

that the total acceleration was a better predicator of steps than acceleration along any of the single axes individually. To exploit this, a pre-processing step calculates the a vector length in euclidean space of the raw acceleration vector.

Detecting steps based on a time series of acceleration vector lengths reduces to peak detection in the time series graph. The step detection algorithm is therefore essentially a generic algorithm for peak detection in noisy data. The noise in the data originates from noise in the sensor readings. To cope with the noise, the data is first smoothed using a moving average[2].

Subsequently the program runs the main peak detection algorithm, which consists of the following steps:

Calculate peak strength

Every data point in the time series has its *peak strength* calculated. Informally, the peak strength of a data point is a measurement of how much that point is worthy of being considered a peak. A wide range of functions can map the raw vector lengths into a peak strength graph, but an additional desideratum of the peak strength function is that the output values should be normalized around 0, where only actual potential peaks are given sub-zero values. The peak strength for a point x_i is calculated as $\frac{\frac{(x_i - x_{i-1} + x_i - x_{i-1} + \ldots + x_i - x_{i-k})}{k} + \frac{(x_i - x_{i+1} + x_i - x_{i+1} + \ldots + x_i - x_{i+k})}{k}}{2}$, where k is a constant referred to as the *peak strength window.*

SKRIV OM KORLEIS EIN FINN VERDIANE SOM WINDOW_SIZE etc.

[2] That is, every data point has its value set to the average of itself and the x neighbours before and after it, where x is called the *window size* of the smoothing.

Appendix A

Appendix

Appendix content goes here.

A.1 Mockups

After the second meeting with the customer, the group had a sketch that was to be used as a starting point for the mock-up application.

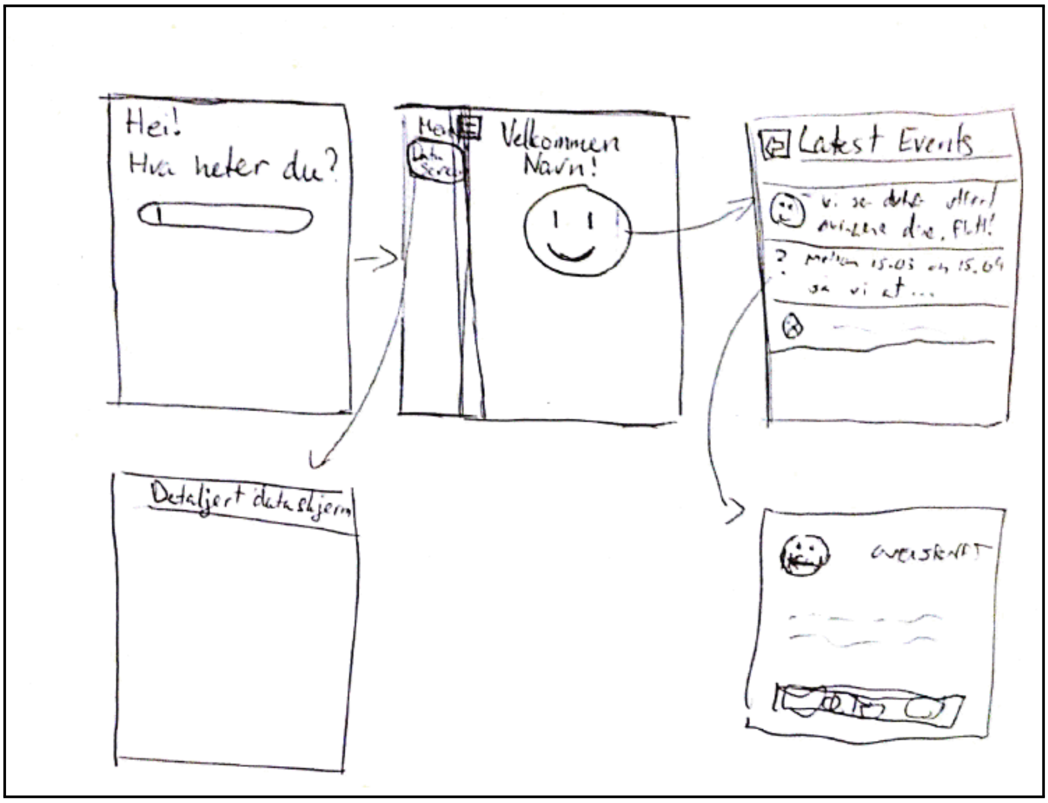

Figure A.1: A mockup of the program flow, text is just scribbling

A.2 Reports

The work done was described as short periods called sprints.

Figure A.2: Short summary of work done sorted by sprint

Sprint nr.	Date	Summary
Sprint 1:	03.02.13 - 08.02.13	Developing user stories and paper prototypes of the GUI.
Sprint 2:	08.02.13 - 15.02.13	Developing a mock-up application demonstrating the GUI.
Sprint 3:	15.02.13 - 22.02.13	Improving UI and functionality for the prototype, researching medical factors.
Sprint 4:	22.02.13 - 01.03.13	Improving secondary functionality for the prototype (settings, statistics, relatives screens),researching content-providers and alternative solutions.
Sprint 5:	01.03.13 - 07.03.13	Creating content-provider, creating presentation, finishing secondary screens (as described above), including pedometer, writing test-cases.
Sprint 6:	08.01.13 - 1603.13	

A.3 Meeting summaries

Regular status reports was a part of necessary documentation. Of particular importance was reports to the supervisor, and reports done to the other group members. This status report was written during sprint 4. The reports were included in chronological order.

Project FallPrevention Group 08

Status report for week 09

1: Progress summary

Prototype:

- The mock-up application has been developed further, in particular with a better-looking GUI and several more screens to display data.
- A database has been build with the application to store all the required data.
- Enabled localization for the application, and included Norwegian and English text. File for the german version are avaliable, but not yet translated.

Research:

- The group has learnt about ContentProviders, which Android uses to encapsulate data from outside of the application.
- A short summary on consequences of, and what factors results in, falling among the elderly, has been written and read by the entire group.

2: Open/Closed problems

- **Group members out of town:** Some group members had trips out-of-town, resulting in less work being done. This was solved by not planning more work to be done in those time frames, than what the remainder of the group could accomplish. With group members remembering to inform the rest of the group in advance, it does seem like this problem is closed.
- **Customer taking vacation:** The group had to skip one meeting with the customer, as the customer took some time off for vacation. As the product is still in a very primitive state, and any further work need to be discussed with the customer, this caused a delay of almost a week, as there was limited work to do in the meantime.

3: Planned activity for next period

The plans for this week are as follows:

- With the graphical and persistence-related parts of the prototype mostly implemented, the details and implementation of the ContentProvider handling interaction between the application and the motion detector seems like it will need more work.
- Complete the implementation of the GUI, and polish it by making it more readable, and adding the rest of the icons necessary to display all the intended permutations.
- Prepare a demonstration for the experts the customer will put the group in touch with, as the expert is not experienced with application development and might need a more user-friendly demonstration.
- Research applications that use sensor technology.

The plans for the period beginning this friday will need to be discussed with the customer and any medical expert the group has yet to meet.

4: Updated risks analysis

While we will not discuss updating the risk analysis until the meeting this wednesday, it is evident that some risks need to be updated:

- “Customer not able to meet with the group” will need to be added to the risk list, although it does not seem very likely to happen more that once. As the group has some understanding of what parts of the project needs working on, it will not become a large delay unless it happens several times in a row.
- “Computer problems” has had a larger influence than we have expected.
- A new risk, “participant not able to work as much as required” will need to be added.

Project FallPrevention group 08

Status report for week 11

1:Progress summary

Prototype:

- Content Provider implemented, but not yet integrated with prototype application.
- Content Provider now has database functionality.
- Integrated pedometer functions
- Application can now add contacts for the purpose of automatically contacting the contact in particular circumstances.
- Application now has settings screen with a few functions.

Research:

- Research into Android applications Edomondo Sports Tracker, Pedometer, and GPS Status has been done, as these apps use and display sensor data in ways that the group might find very useful.
- Pedometer is also open source and has an compatible licence, so it has been integrated in our project and application
- Made presentation and prepared questions for meeting with medical experts.

Documentation:

- Feedback from the preliminary report has been fixed
- Requirements partially updated to current needs
- Testing plan researched and prepared

2: Open/Closed problems

There has mainly been three problems since last report. They are: Unpredictable illness for group members, e-mail still being inefficient at communicating and getting responses swiftly, and group member not being equally experienced with the tools being used, slowing down productivity.

3:Planned activity for next period

- Finish filling testing plan with tests
- Update requirements in the report
- Update architecture in the report to reflect the current state of the application
- Have meeting with medical experts, and gather information about what the application should focus on.
- Integrate Pedometer into the Content-Provider, and get the application ready to detect
- Take a few passes with refactoring and documenting over all the code, with an aim to improve readability and quality of code.

4: Updated risks analysis

The risks as described above is already included in the risk list. With the topmost item on the list being implementation difficulties(mainly people not having the programming competencies the tasks demand). Documenting and refactoring the code, something which everyone will take part in, will likely improve the competency of the group, and their understanding of the code, to such a level that the likelihood of this problem will be reduced.

Bibliography

[1] ”Falls in older people: epidemiology, risk factors and strategies for prevention”. Laurence Z. Rubenstein. Age and Ageing 2006; 35-S2:ii37-ii41.

[2] ”A phsycological Profile Approach to Falls Risk Assessment and Prevention”, Stephen R. Lord,Hylton B. Menz and Anne Tiedemann, Journal of the American Physical Therapy Association.

[3] ”Sensori-motor Function, Gait Patterns and Falls in Community-dwelling Women”. Stephen R. Lord, David G. Lloyd, Sek Keung Li. Age and Ageing 1996:25:292-299.

[4] ”What you can do to prevent falls”, Centers for Disease Control and Prevention