



# Interactive Home Security demos with Engduino and TouchDevelop

Team 10

Robert Sultanaev, Madeleine Whitehall, Hong Hui Yu

Clients: Dr Dean Mohamedally and Microsoft Research

## Abstract

The aim of this project was to create a primitive home security system using the Engduino coupled with the appropriate sensors and Microsoft TouchDevelop. The purpose of the app is to serve as a small demo for a workshop about TouchDevelop and how it can be used to interact with devices. The system should allow the user to connect a series of home security related sensors (e.g. laser tripwire, motion sensor) that operate on a digital output basis (either HIGH or LOW state) to an Engduino and then use an app to configure the system, switch the system on, receive notifications and log the data gained from the sensors.

We built a simple home security system using the Engduino board, Engduino bluetooth module, IR trip sensors, a motion sensor and a buzzer, and an app that allows the user to configure and interact with the system. The communication between the Engduino board and the app happens either through bluetooth (for android version) or through USB (for PC version).

## Table of Contents

|  |    |
|--|----|
| Context.....                                     | 1  |
| Background to project and clients.....           | 1  |
| Problems being solved.....                       | 1  |
| Team Skills and Roles.....                       | 2  |
| Requirements.....                                | 3  |
| Gathering process.....                           | 3  |
| The initial list of questions to the client..... | 3  |
| Notes gathered from the interview .....          | 4  |
| List of requirements in MoSCoW style .....       | 5  |
| Work packages by team member .....               | 6  |
| Glossary.....                                    | 7  |
| Design.....                                      | 8  |
| UI Design.....                                   | 8  |
| Initial mockups .....                            | 8  |
| Finalised app walkthrough .....                  | 14 |
| Technical Design.....                            | 23 |
| The app structure .....                          | 23 |
| Hardware used for the project .....              | 24 |
| Software used for the project .....              | 24 |
| App data .....                                   | 25 |
| Design patterns in the project.....              | 25 |
| User testing procedure guidelines .....          | 26 |
| References .....                                 | 27 |
| Appendix – Biweekly reports .....                | 28 |
| Bi-weekly report #1 .....                        | 28 |
| Bi-weekly report #2 .....                        | 31 |
| Bi-weekly report #3 .....                        | 34 |
| Bi-weekly report #4 .....                        | 37 |
| Bi-weekly report #5 .....                        | 41 |

## Context

### Background to project and clients

Microsoft Research will be holding a TouchDevelop workshop, the aim of which is to show how TouchDevelop can be used to interact with devices. The device that we will be using is the Engduino coupled with sensors related to home security, such as IR tripwires and motion sensors.

UCL's Dr Dean Mohamedally is our main client, and he provides us with the requirements for the app, but we are also working with Peli de Halleux of Microsoft Research who provides us with additional information regarding the TouchDevelop platform.

### Problems being solved

Showing how versatile TouchDevelop is, and how it can be combined with other technologies to create practical and/or interesting applications that use a variety of different devices.

## Team Skills and Roles

### Robert Sultanaev

Good at analytical thinking and problem solving, likes to keep things organised and in order. Good enough people skills. No notable project/work experience. Some programming experience prior to university (familiarity with most of the year 1 programming modules' syllabus, some web development experience and little android experience).

#### Roles assigned:

- Team Leader and Project Manager
- Technology Platform Research Lead
- Lead Interviewer for Requirements
- Programming and Repository Lead

### Madeleine Whitehall

Interested in design, aesthetics and working on the visual components of the app. Has good people skills. Has done a school project that involved designing a product in a team and putting it on the market. Not much programming experience prior to university.

#### Roles assigned:

- Documentation Lead
- Lead UI designer and User Testing

### Hong Hui Yu

Good at analytical thinking and problem solving, is working hard to improve his programming skills. Not the best talker. No notable prior project/work experience, and not much programming experience prior to university.

#### Roles assigned:

- Lead Device/Platform Tester
- Secondary Programmer
- Assistant Technology Researcher

# Requirements

## Gathering process

The initial list of questions to the client

- Do we understand the theme of the project correctly [Explain how we understand the theme]?
- On what kind of platforms should the app run (android, iOS, win phone)? Does any platform take priority?
- Should it be a webapp?
- Any other technology other than touchdevelop that we are expected/recommended to use?
- Details about the functionality of the app and the setup:
  - What we must/should/would want to have?
  - How many sensors
  - What kind of sensors
  - How much should the user be able to configure
  - Is there an alarm? What kind (ringing bell or email or sms)?
- Details about the parameters of our demo:
  - What we must/should/would want to have?
  - General layout of the demo
  - Area
  - # of rooms
  - Stage demo (with informed demonstrators) or interactive (for visitors)
  - If stage, how is it setup?
  - If interactive, # of people concurrently interacting with the demo
- Are there any limitations we should keep in mind?

### Notes gathered from the interview

Since our understanding of the task was not entirely correct before going into the interview, the list of questions that we prepared did not work exactly as planned. So instead we decided to organise our notes from the interview by topics to which they relate, not as answers to specific questions.

- Theme & general outline
  - Downloadable app
  - Using sensor readings & hardware interface of the engduino & touchdevelop
  - "Practical" real world use cases as demonstrations
  - Our demo: a series of experimental tests for home security
- Platforms
  - PC's are a priority, should work on cordova (mobile) as well.
- Other technologies to use
  - Cortana speech recognition.
  - Should work through usb first then also wirelessly (bluetooth)
- Possible sensors to use
  - Door sensors
  - Volume in a room
  - Unusual rise in temperature
  - Tripsensor
  - Pressure pad
- App's responses to when a sensor is triggered
  - Bell alarm
  - Email
  - Facebook notification
  - Sms

## List of requirements in MoSCoW style

Requirements that were successfully completed are highlighted in **green**, and the ones which were failed to achieve are highlighted in **red**.

### Must have:

- App must be working on at least one platform – mobile with bluetooth or PC with USB
- App must be able to send and receive information to and from the Enguino.
- Allow the user to configure the system:
  - Specify which pin connectors are connected to which sensor.
  - Specify which sensors are being used.
- System must work with at least one sensor.
- Notify the user that an intrusion has been detected (the of the Enguino input pin changed) through a pop-up window.

### Should have:

- Save user configurations as “location profiles”
- Allow the user to switch between location profiles.
- System working with more than one sensor.
- Ability to log data and display it.
- Notify the user of an intrusion by sounding a buzzer

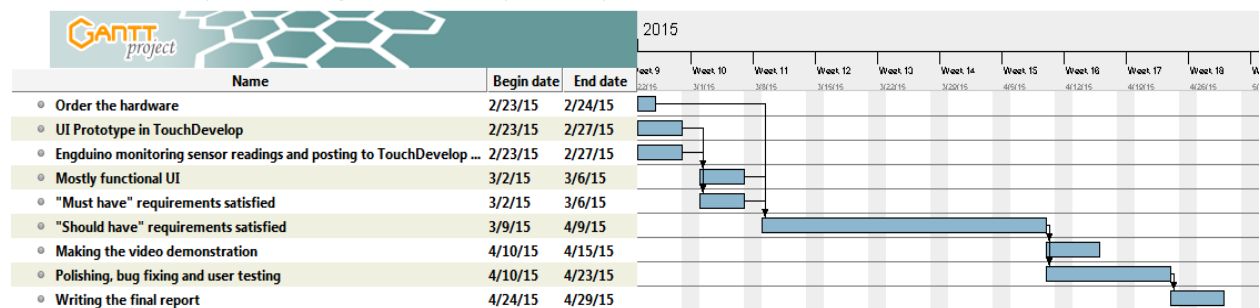
### Could have:

- App working on a second platform (mobile or PC, depending on which was implemented first).
- User input using Cortana.
- Another means of notification, such as e-mail or Facebook.

### Would have:

- Notifications via sms.
- Multiple user interfaces for different platforms.

## Gantt chart representing the development process





## Work packages by team member

### Robert Soultanaev

- Building the hardware part of our project, most notably constructing the IR trip sensor from an IR receiver, transceiver and a number of other components.
- Sorting out all technical aspects of the communication between the app and the Engduino via the Engduino protocol.
  - Via bluetooth
  - Via USB
- Programming the app functionality
- Testing the app functionality
- Communication with our client and contact at Microsoft

### Madeleine Whitehall

- Designing the UI
  - Drawing mock-up sketches
  - Programming the UI elements
- Designing the app icon
- Production of the video demonstration
  - Filming
  - Editing

### Hong Hui Yu

- Aiding with programming the app functionality
- Researching the possible technology additions to our app and determining whether it is reasonable to attempt to include them in the project.
  - Facebook notifications
  - E-mail notifications
  - Communication between the app and Engduino via USB
- Writing the user testing procedure for our app

## Glossary

- Page – A segment of the UI that occupies an entire screen.
- Profile – A configuration saved by the user, that specifies what sensors occupy which pins on the Engduino, the sensor names and whether the sensors trigger a buzzer. The user can also define a name for a profile. Semantically a profile would usually represent a location, so for example a user might have profile for a home environment and one for the work environment.
- Intrusion – When a sensor attached to the engduino reacts to the environment and outputs a HIGH state at its output pin.
- Log – Stored information regarding detected intrusions.
  - Log entry – Name of the sensor that detected the intrusion, time at the intrusion and the selected profile.

# Design

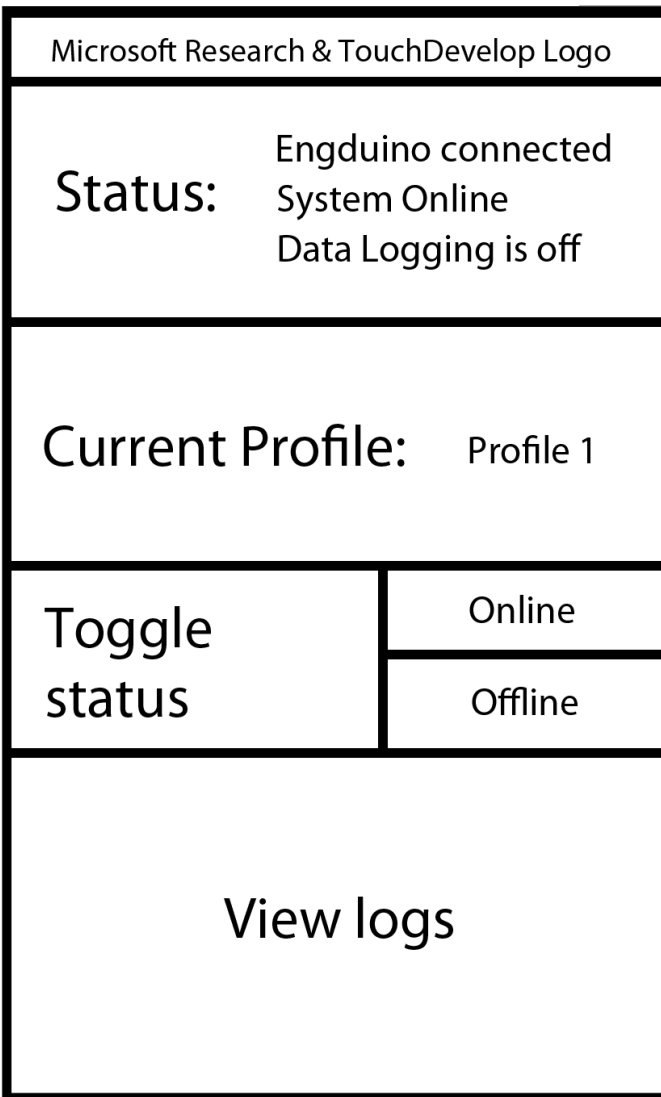
## UI Design

### Initial mockups

The following are the initial mock ups of our planned user interface in a wireframe style. These were made to guide us in the process of UI design.

The drawings were made using Adobe Photoshop.

Figure 1: Home Page



This is the Home screen of our app. Here the user can see whether the Engduino has been connected, whether the system is on or off, whether the data is being logged, and the current profile being used.

The user can enter the profile page by clicking on the “Current profile” box. The user can also toggle the system on (online) or off (offline), and enter the logs page by clicking “View logs”.

The Microsoft Research and TouchDevelop logos will be displayed at the top of every page and if clicked, will take the user to this home page.

Figure 2: Profiles Page

|  |      |
|--|------|
| Microsoft Research & TouchDevelop Logo |      |
| Current Profile: Profile 1             |      |
| Profile 1                              | Edit |
| Profile 2                              | Edit |
| Profile 3                              | Edit |
|  |      |
| Create new profile                     |      |
| ← Back                                 |      |

This is the profiles page of the app, where the user can see the current profile being active, choose a different profile (by clicking on the appropriate profile name), edit existing profiles and create new profiles.

Figure 3: The Edit/Add Profile Page



|  |        |
|--|--------|
| Microsoft Research & TouchDevelop Logo |        |
| Edit/Add profile                       |        |
| Profile name:                          |        |
| Sensor 1                               | Edit   |
|  | Remove |
|  | on/off |
| Sensor 2                               | Edit   |
|  | Remove |
|  | on/off |
| Add sensor                             |        |
| Save profile                           |        |
| ← Back                                 |        |

This is the page for editing existing profiles or adding new profiles. The title at the top will say “Edit profile” or “Add profile” depending on which action the user is taking, otherwise the page looks identical in both cases.

A profile consists of a name and the information about which sensors are connected to which pin connectors on the engduino and which sensors are active.

On this page the user can change the profile name, add or remove sensors from the profile and then save the changes.

Figure 4: Edit/Add Sensor Page

|   |  |
|---|--|
| Microsoft Research & TouchDevelop Logo  |  |
| Edit/Add sensor   |  |
| Sensor name:  |  |
| Pins occupied:  |  |
|    |  |
| Save sensor configuration   |  |
|  |  |

This is the page for editing existing sensors or adding new sensors. The title at the top will say “Edit sensor” or “Add sensor” depending on which action the user is taking, otherwise the page looks identical in both cases.

On this page the user can specify the name of the sensor and which pin connectors it occupies. The grid below “Pins occupied” represents the pins on the Engduino and the user can click on a cell to make that pin count as occupied by the sensor (the crossed out cells represent the selected pins in this illustration).

Figure 5: Logs Page

|  |               |
|--|---------------|
| Microsoft Research & TouchDevelop Logo             |               |
| Toggle data logging                                | On            |
|  | Off           |
| Start: 12.01.2015, 15:23<br>End: 12.01.2015, 20:21 | View Log data |
| Start: 14.01.2015, 13:02<br>End: 14.01.2015, 17:33 | View Log data |
|  |               |
| ← Back   |               |

This is the logs page, where the user can turn data logging on or off, and view the logs that have been recorded previously. Clicking “View Log data” will take the user to a page for that particular log.

Figure 6: Log Data Page

|  |   |
|--|---|
| Microsoft Research & TouchDevelop Logo |   |
| Intrusions detected:                   | 2 |
| Sensor 1 detected intrusion at 15:43   |   |
| Sensor 2 detected intrusion at 16:02   |   |
|  |   |
| ← Back                                 |   |

This is the Log Data Page. Here the user can see how many intrusions (sensor readings going over threshold value) have been detected in total during the logging interval, as well as which were the sensors that detected the intrusions and at what times the intrusions occurred.

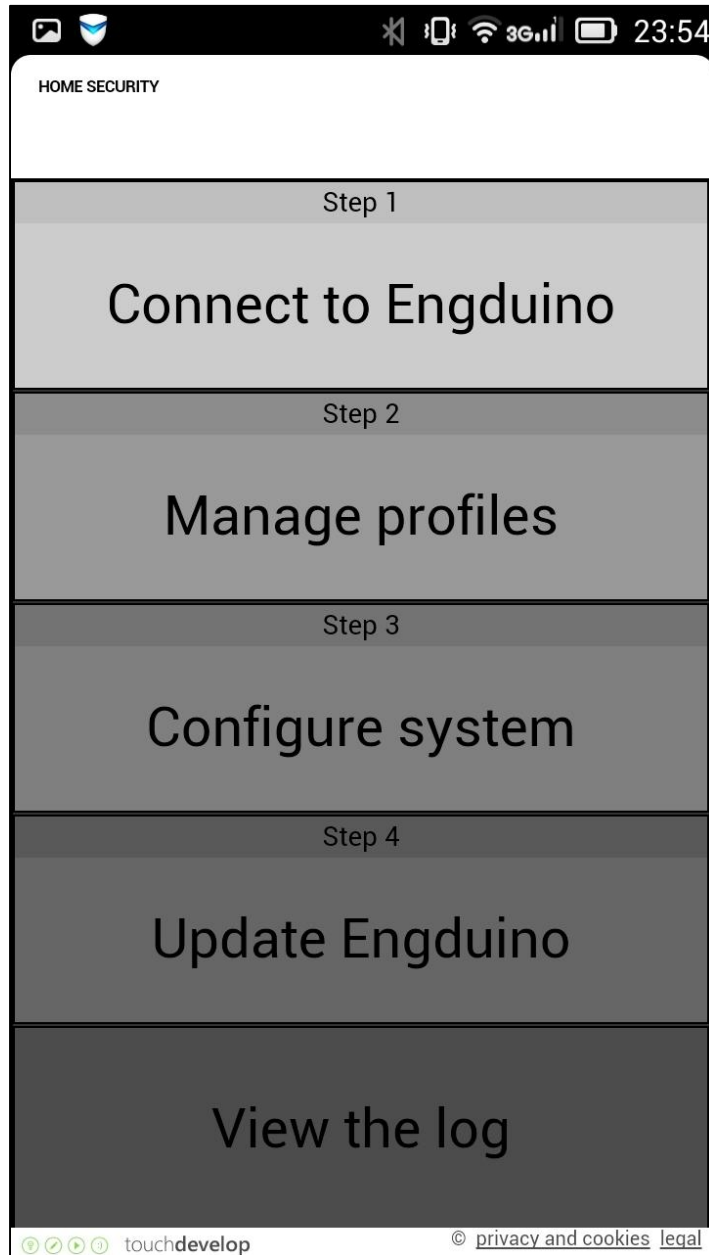


## Finalised app walkthrough

The following are the screenshots of the final app running on an android phone.

The UI of the PC version is nearly identical and is therefore not worth showing.

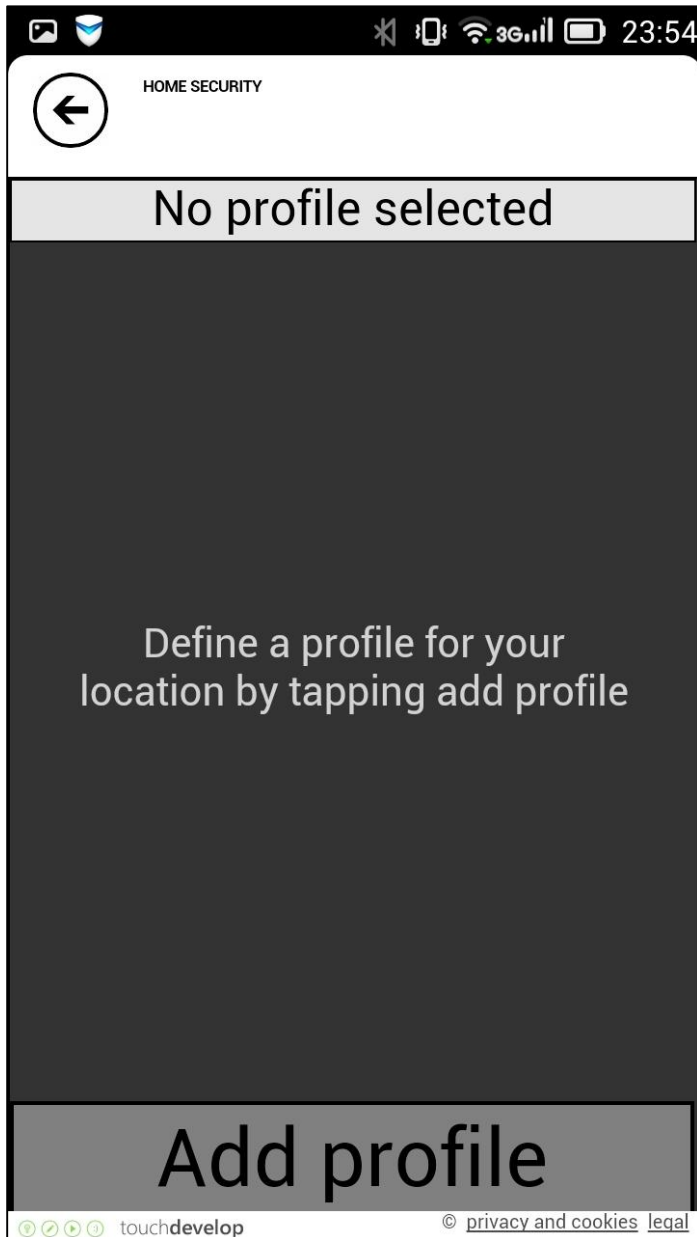
Figure 7: Home page



This is the home page of our app. By tapping the respective boxes, the user can connect to the Engduino and update the configuration on the Engduino, navigate to the manage profiles page, the configure system page and logs page.

We also included step numbers to make it clearer how to use the app, because the order of these steps is important.

Figure 8: Manage profiles page without profiles

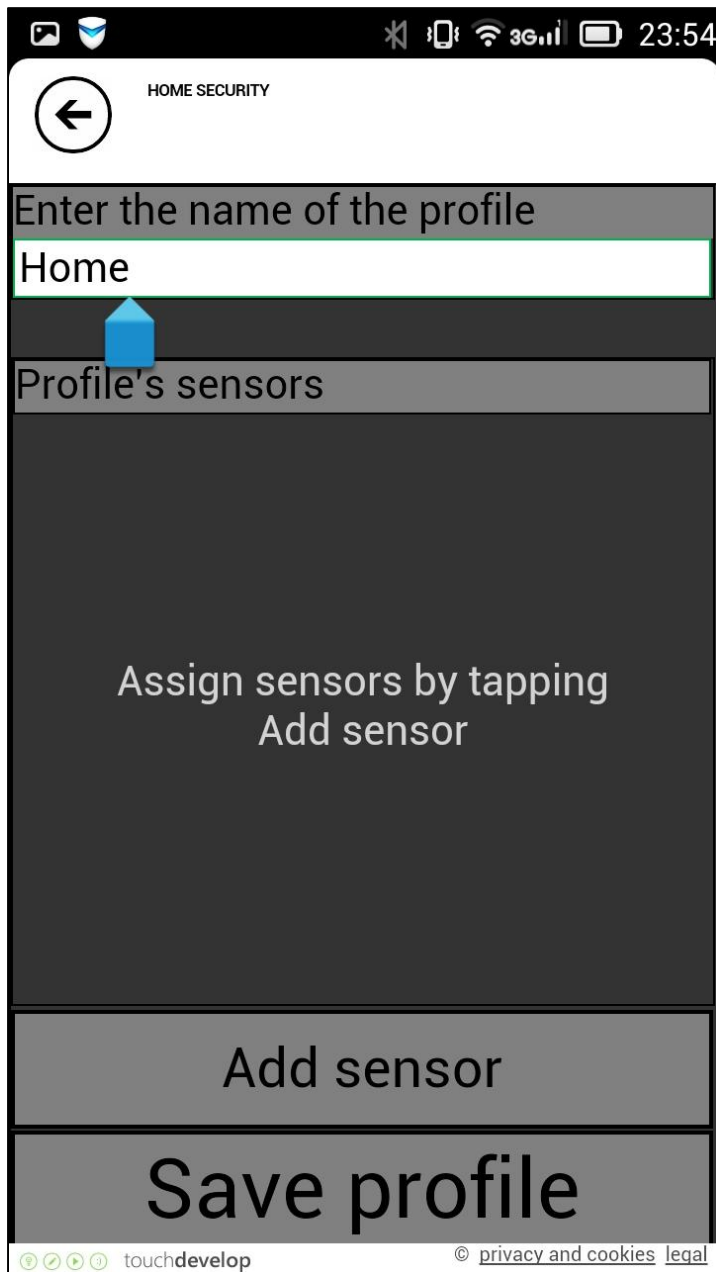


This is the manage profiles page. Currently the user has not created any profiles, so no profile is selected and there is a tip in the empty space, prompting the user to create a new profile.

Tapping “Add profile” will lead the user to the add profile page.

Also there is a “back button” at the top left of the screen, which will in all cases take the user to the previous page.

Figure 9: Add/edit profile page

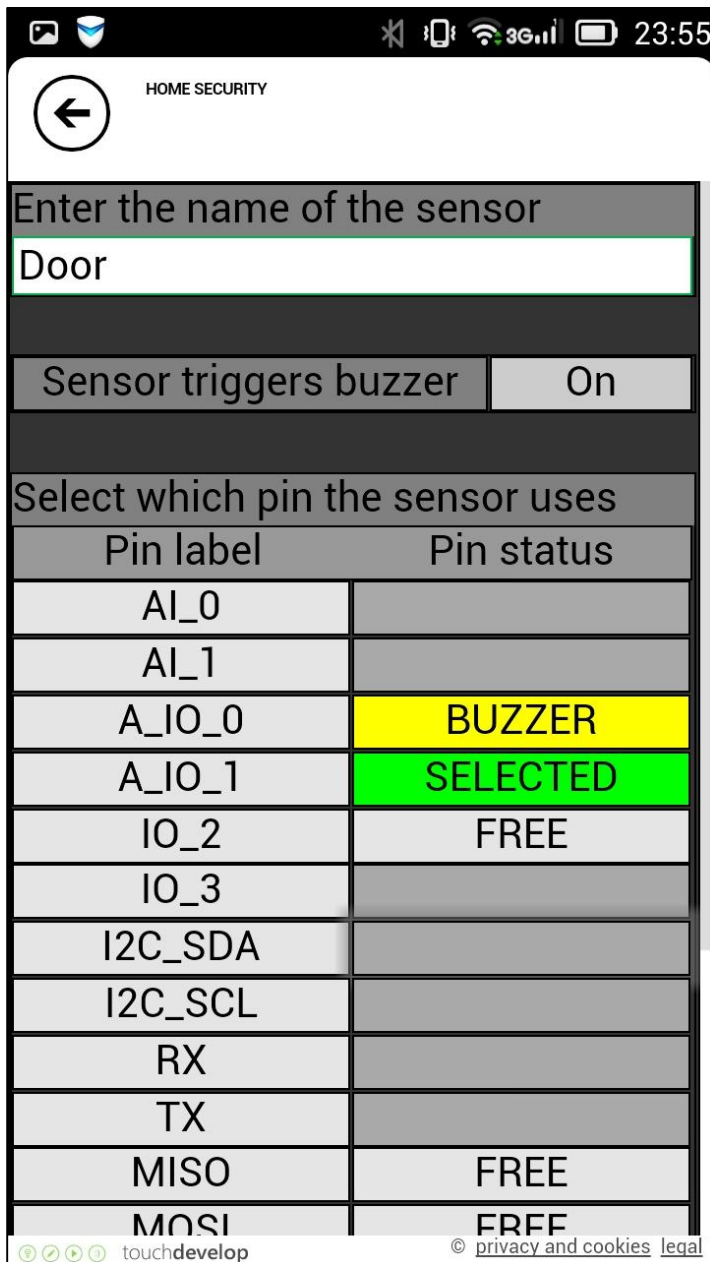


This is the add/edit profile page. Here the user can specify the name as well the sensors of the profile that the user wishes to create.

Tapping "Add sensor" will take the user to the add sensor page.

Tapping "Save profile will" save the profile with the given configuration.

Figure 10: Add sensor page



This is the add sensor page. Here the user can specify the name of the sensor, whether it will sound the buzzer, and to which pin on the Engduino the sensor is connected to.

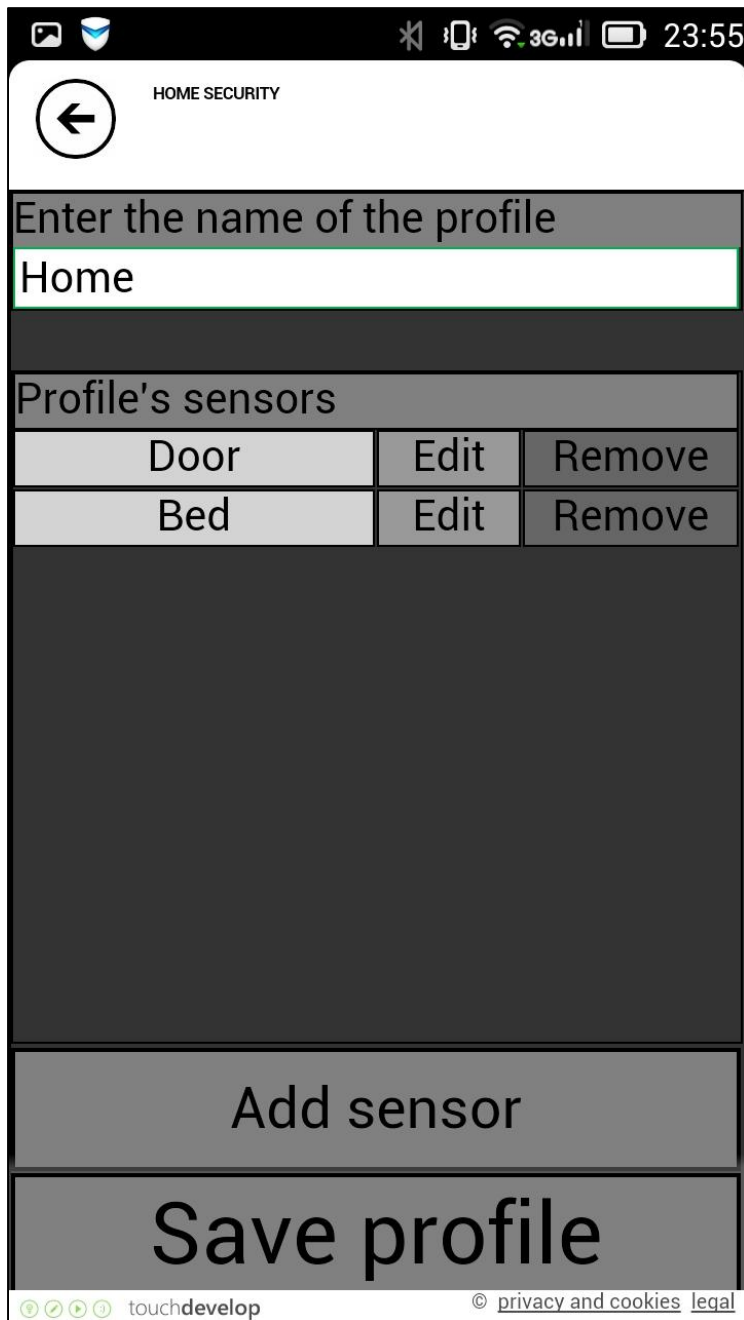
The pin selection interface uses the same pin labels that are printed on the Engduino itself next to the pins, so that the user can then match the configuration in the app with the configuration of the system.

The pin status column represents whether the pin is free for selection (light grey), currently selected (green), occupied by another sensor (red, not shown), occupied by the buzzer (yellow), or is unavailable to use for other reason (grey).

The user can select the pin by tapping on the boxes that indicate that the pin is free for selection.

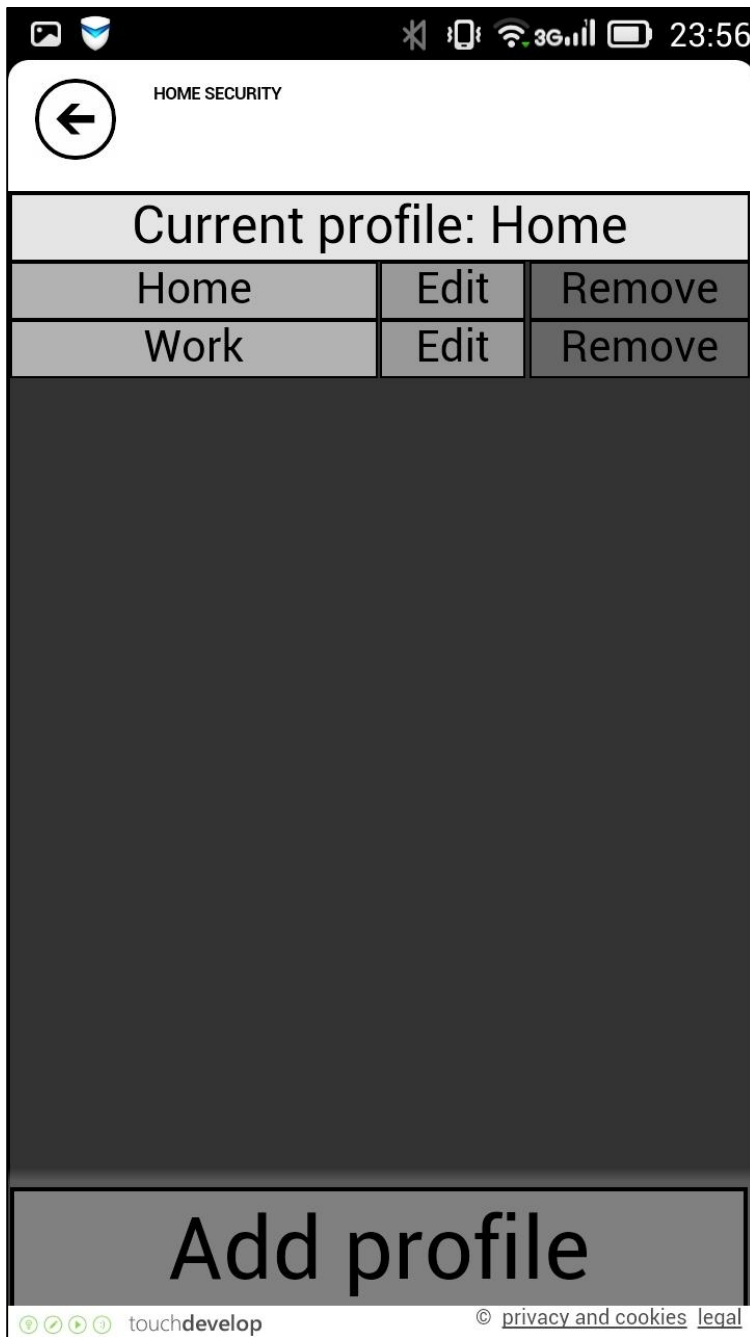
This page actually takes up more space than is on the screen, and requires scrolling down to reach the “Add sensor” button.

Figure 11: Add/edit profile with sensors



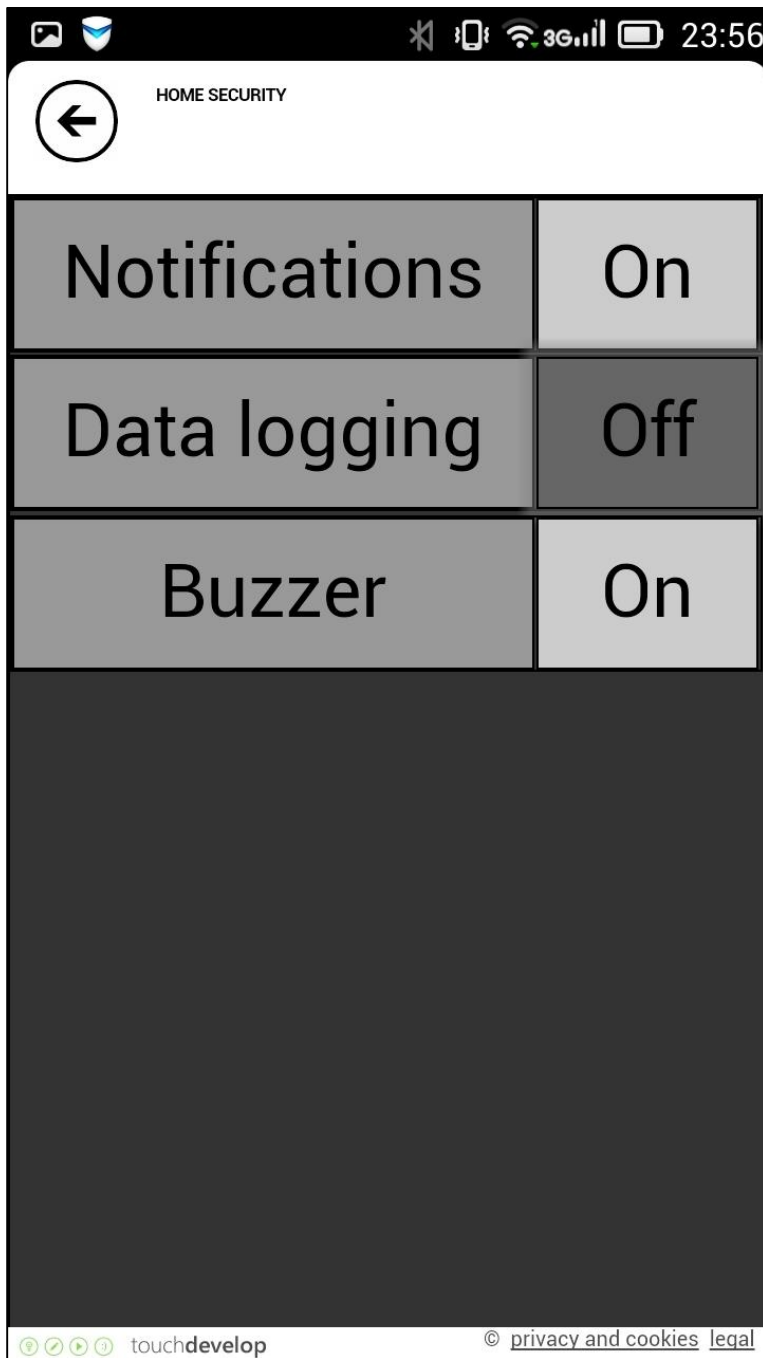
This is how the add/edit profile page looks like after a few sensors have been added to the profile.

Figure 12: Manage profiles page with profiles



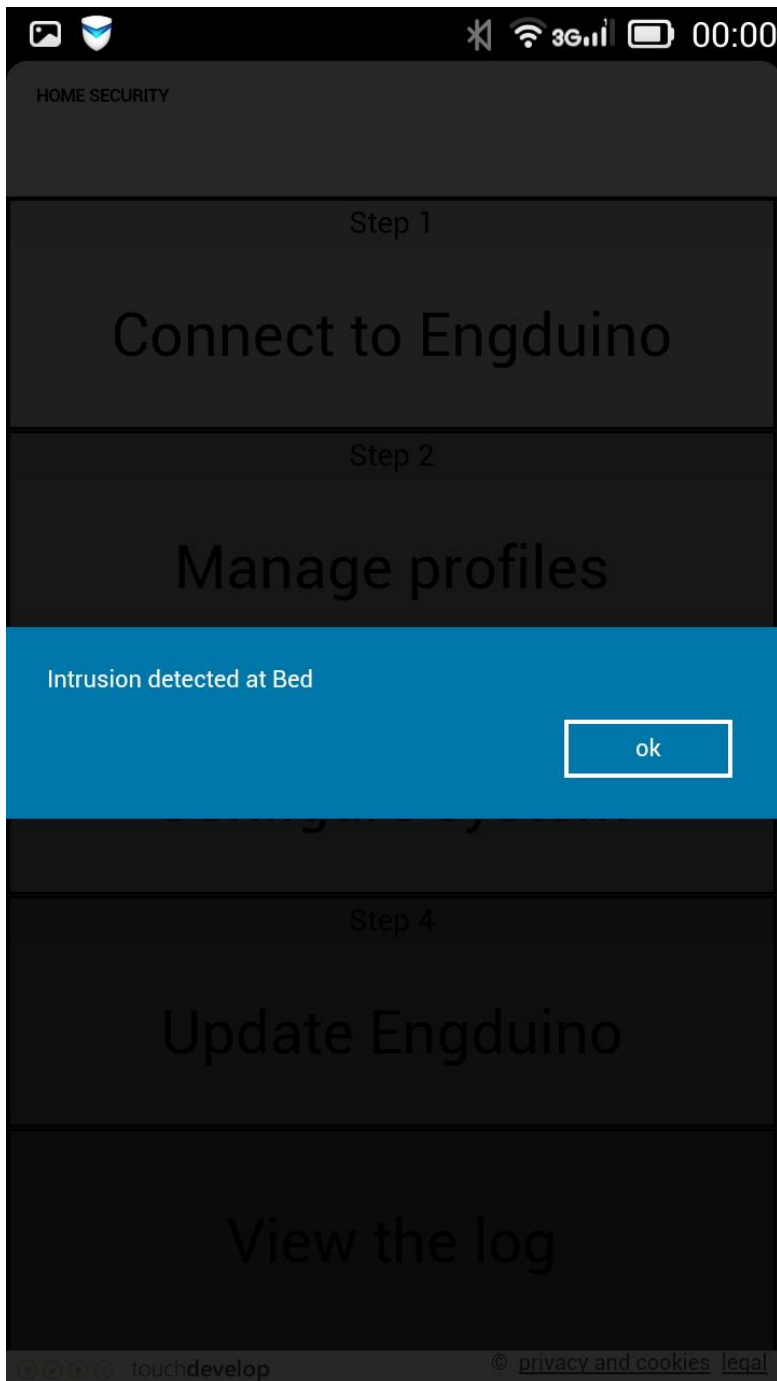
This is how the manage profiles page looks like when a few profiles have been added, and the "Home" profile selected.

Figure 13: Configure systems page



This is the configure systems page. Here the user can turn on and off notifications, data logging and the buzzer by tapping on the box labelled on/off.

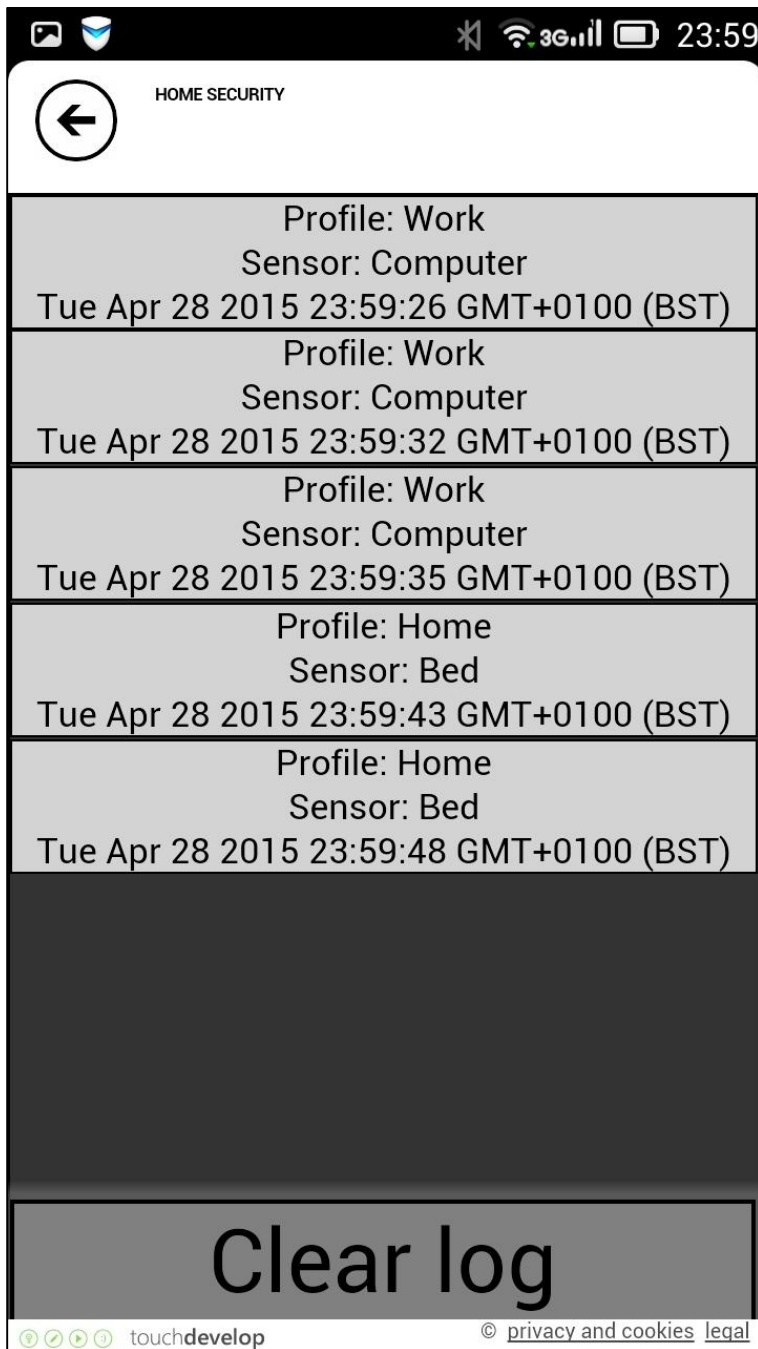
Figure 14: Notification message



This is how a pop-up notification that tells the user about an intrusion looks like.



Figure 15: Logs page



This is the logs page. Here the user can see the activity that the sensors connected to the Engduino have shown while the app was running.

Tapping “Clear log” will remove all log entries from the database.

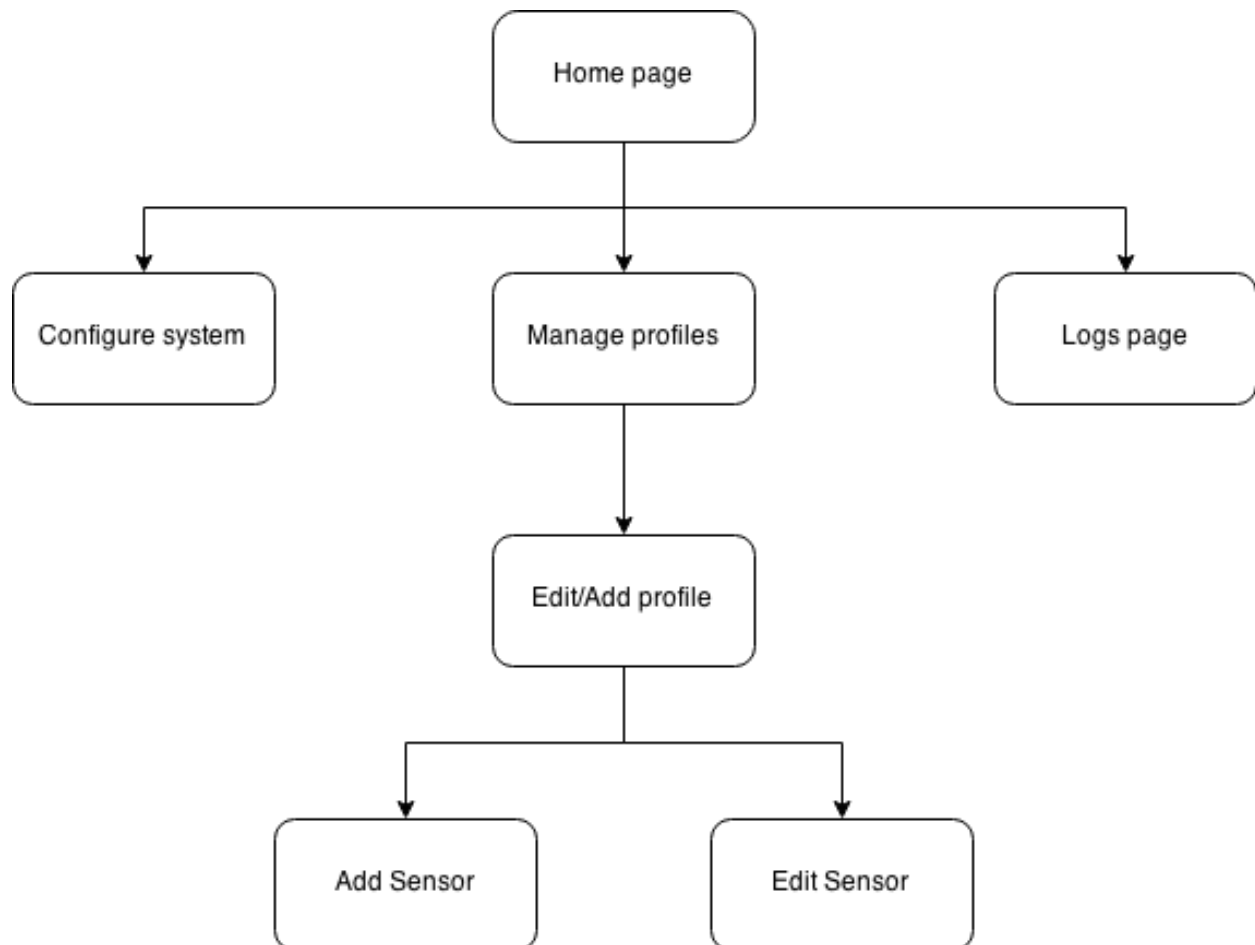
## Technical Design

### The app structure

Below is a flow diagram that represents the structure of the pages in the app (the back button will act in the direction opposite to the arrows in the diagram).

It is worth clarifying that the “Edit/Add profile” is indeed one page, not two.

The diagram was drawn using a webapp available at [www.draw.io](http://www.draw.io).



## Hardware used for the project

The app uses the UCL Engduino 3.0 microcontroller board coupled with with Engduino Bluetooth module (for the mobile version) to power the sensors and communicate with the app.

The system is designed to work any sensors that produce a digital output (HIGH or LOW), and in our case we used IR trip sensors which outputs HIGH when the IR beam between the receiver and transceiver is broken and a motion detector which senses the changes in IR light. We also used a piezo buzzer to produce an alarm sound.

IR tripsensors were built according to the schematic by James Petersen [4], using IR transceivers and receivers. For the motion sensor we used the Parallax Wide Angle PIR Sensor.

## Software used for the project

The Engduino was flashed with a program that implements the Engduino protocol for serial communication. Both the protocol and the implementation were written by Jan Medvesek of UCL, which is part of the Engduino extension for the Arduino IDE written by UCL's Engduino team [1]. We collaborated with Jan in order to create a couple of new methods for the protocol by first implementing the methods ourselves and then sending them to Jan, who then modified our solutions and made the commits to the repository.

The app was written using the TouchDevelop language and programming environment developed by Microsoft. To build the mobile app we exported our TouchDevelop program to a cordova program, which was then used to build an android application. For the PC version we used the TouchDevelop local shell environment to run our app in.

To achieve bluetooth communication on the mobile platform we used the cordova bluetoothserial library created by Peli de Halleux [2], which is a TouchDevelop wrapper for the cordova bluetoothserial plugin [5]. For the PC version the communication with the Engduino is done via USB and it uses the shell library [3] to communicate with the Engduino via serial port.

## App data

The app uses a database comprising of three tables:

- Sensors table
  - sensorName (type: String)
  - sensorID (type: Number)
  - pinUsed (type: Number)
  - profileID (type: Number)
- Profiles table
  - profileName (type: String)
  - profileID (type: Number)
- Logs table
  - profileName (type: String)
  - sensorName (type: String)
  - timeActivated (type: DateTime)

## Desing patterns in the project

The observer pattern has been used for logging the data – whenever it is detected that a new message from the Engduino has arrived, the app reacts by updating the log table of the database by inserting a new entry into it. A few other things also react to the event of receiving a message – notification pops up and the buzzer is sounded.

State pattern is used in the pin selection interface (figure 10). Each cell of the pin state column has a state associated with it (represented by a number), and the state determines the colour and text of that cell. The information of cell states is also used when the app checks whether the user has selected the correct amount of pins (i.e. exactly 1).

The adapter pattern is used in the cordova bluetoothserial library which we use for bluetooth communication. This is a library that is actually a wrapper of an existing cordova plugin, adapted to be used for TouchDevelop programming.

The interpreter pattern is in communicating with the Engduino through the Engduino protocol. Every time a message is received from the Engduino, the app parses it and then reacts accordingly.

The facade pattern is used to simplify referencing the Engduino's pins. The pin numbers that the Engduino actually uses are not intuitive and follow very little order, for example one pin might be number 12, whereas the neighbouring pin could be number 3. To simplify the issue of referencing a particular pin, the app provides functionality to map from the actual Engduino pins to the app representation of the pins and vice versa.

## User testing procedure guidelines

In order to have some structure to the user testing process, these guidelines were written to more or less standardise the procedure. Unfortunately we could not conduct very many tests, but the few times that we did we gained some valuable information and good ideas on improving our products (such as showing a pin to be occupied by another sensor when creating a new sensor).

### Use case

1. Connect to Engduino
2. Add a sensor
  - a. Enter the name of the sensor
  - b. Configure the pins
3. Update Engduino
4. Check the logs of the sensors
5. Clear log

### Question topics about usability

1. Learning to operate the system
2. Exploring new features by trial and error
3. Remembering names and use of commands
4. Performing tasks is straightforward
5. Help messages on the screen
6. Supplemental reference materials

### Question topics about UI

1. Font sizes
2. Color scheme
3. Labels
4. Text format
5. Text boxes
6. Captions
7. Buttons
8. Icons
9. Menu styles

## References

- [1] UCL Engduino Team. *Engduino extension for Arduino IDE*. Github repository:  
[https://github.com/engduino/Engduino\\_v3.1.0\\_arduino\\_extension](https://github.com/engduino/Engduino_v3.1.0_arduino_extension)
- [2] Peli de Halleux. *Cordova bluetoothserial library*. Source code and documentation:  
<https://www.touchdevelop.com/gumze>
- [3] Peli de Halleux. *Shell library*. Source code and documentation:  
<https://www.touchdevelop.com/qmzn>
- [4] James Petersen. *Infrared (IR) tripwire module*. Retrieved from Instructables:  
<http://www.instructables.com/id/Infrared-IR-tripwire-module/>
- [5] Don Coleman. *Bluetooth Serial Plugin for PhoneGap*. Github repository:  
<https://github.com/don/BluetoothSerial#bluetooth-serial-plugin-for-phonegap>

## Appendix – Biweekly reports

### Bi-weekly report #1

App project Team 10 – Sultanaev, Whitehall, Yu

**21.01.2015, 15:00**

**Topics discussed:**

Meeting up with the team members, team members' background and preferences, assigning roles to team members, methods for communication and file sharing.

**Problems overcome:**

Communication will be done through a WhatsApp group, since all of us regularly check it, all messages will be received quickly.

Sharing source code will be done through a private repository on GitHub, other files can be shared through Dropbox or Facebook.

**Next meeting:**

Getting information on our clients

**28.01.2015, 11:00**

**Topics discussed:**

Going through what our app is about and what we will require for the project, potential questions for the client interview.

**Problems overcome:**

We were told that we will be using software that requires windows 8, this is solved by each of the team members getting a copy for themselves from Dreamspark and then installing it into either a virtual machine or setting up a dual-boot.

We decided that each team member comes up with as many interview questions as possible and post them on the WhatsApp group. The team leader will then compile these questions into one list which will be used for the interview.

**Next meeting:**

Skype call with our Microsoft contact

**03.02.2015, 9:00 & 17:00**

**Topics discussed:**

The new information we got from the Skype call with our contact at Microsoft arranged by Dean Mohamedally, meeting our client.

**Problems overcome:**

The time for the client meeting will be 05.02.2015 at 15:00.

**Next meeting:**

Interviewing our client

**05.02.2015, 15:30**

**Topics discussed:**

Results of our client interview

**Problems overcome:**

The team leader will re-write the transcript from the interview into a more readable format and share it with the other team members.

**Next meeting:**

Drawing our first sketch for the app's user interface

**09.02.2015, 13:00**

**Topics discussed:**

Drawing the app's user interface.

**Problems overcome:**

The UI lead will re-draw our sketches more cleanly and share picture of the result with the team.

**Next meeting:**

Discussing the client's feedback from sketch



**10.02.2015, 9:00 & 12:30**

**Topics discussed:**

The client's feedback from the sketch, additional details about the functionality of our app.

**Problems overcome:**

The team leader will order 3 Endguinos so that the team can get started with learning about Touchdevelop programming on the Engduino.

**Next meeting:**

Re-drawing the app's UI sketch

**11.02.2015, 11:00**

**Topics discussed:**

Re-drawing the app's UI sketch, general plans for reading week.

**Problems overcome:**

For convenience, the meetings during reading week will be held through Skype.

**Next meeting:**

Details about what to do during reading week and the design report

## Bi-weekly report #2

App project team 10 – Sultanaev, Whitehall, Yu

**17.02.2015, 17:00**

**Topics discussed:**

Modifying and polishing the user interface concept for our app, more discussion about team members' roles and what each team member should concentrate on studying.

**Problems overcome:**

The UI was polished and many small tweaks were made, mostly to the "Logs" page. Team was split up so that one of the team members concentrates on the UI design and programming, and another on the Engduino programming. The team leader will participate in both aspects of the project.

**Next meeting:**

Polishing the design report

**18.02.2015, 18:00**

**Topics discussed:**

The meeting was dedicated to working on each aspect of the design report for our first milestone. The team leader had a draft prepared for the meeting, and during the meeting the team was making suggestions on what to add and how to polish each section. Also finalising the categorisation of our requirements into the MoSCoW style.

**Problems overcome:**

All sections of the report were improved due to the team effort, the sections that received the most changes were "Context" and "Requirements".

**Next meeting:**

Assigning work packages for the next week to the team members.

**21.02.2015, 18:00**

**Topics discussed:**

Who is going to do what next week.

**Problems overcome:**

Whitehall was assigned to create a prototype UI in TouchDevelop, such that at least some of the buttons are interactive and it is possible to get to every page of the app. By 24.02 she was assigned to complete four out the six pages for the app.

Yu was assigned to create a simple app that monitors the temperature value from the Engduino and posts to the TouchDevelop wall when the value goes over a set threshold. By 24.02 he was assigned to only have the program post to the command line.

Soultanaev (team leader) was assigned to continue the research into building the sensor circuitry and which components to order for the project.

**Next meeting:**

Showing the progress on the work packages to the team leader.

**24.02.2015, 9:00**

**Topics discussed:**

Whitehall and Yu showed their progress on their work packages.

**Problems overcome:**

Whitehall had some issues with linking the pages in the app (e.g. having a button press bring up another page), but this was a technical issue that was solved during the meeting.

Yu had issue with having the Engduino program posting the TouchDevelop wall, he was pointed to the relevant information (there was no time during the meeting to actually solve the problem).

We also got our contacts for ordering the parts for our project.

**Next meeting:**

Assigning work packages for the rest of the week.

**25.02.2015, 12:00**

**Topics discussed:**

Work packages for the rest of the week.

**Problems overcome:**

Whitehall was assigned to clean up her code for the UI by making it more modular and naming variables more appropriately. The aim was to make future modifications to the UI code easier.

Yu was assigned to continue working on having the Engduino post to the TouchDevelop wall and researching the link between TouchDevelop and Engduino.

Soultanaev (team leader) will order the relevant hardware and continue researching the hardware aspects of the project.

**Next meeting:**

Discussing the progress and assigning work packages.

**28.02.2015, 14:30**

**Topics discussed:**

Current progress, goals for the next week, work packages, altering the requirements.

**Problems overcome:**

The goals for the next week are to have a mostly functional UI (most buttons should do something) and complete the “must have” part of our requirements.

Whitehall was assigned to continue her work on the UI. By 03.03 she is to add the ability to add profiles to the app.

Yu was assigned to create a TouchDevelop app that allows to toggle the monitoring of the Engduino sensor values on and off and have the app react when the sensor values go over a set threshold. He was to complete this by 03.03.

Soultanaev (team leader) was assigned to work on a prototype light barrier circuit to be used with the app.

During the course of the development we noticed that using more than one type of sensor in our project will most likely result in having too much complexity for us to handle in the time given to us. We decided that we should focus on only using light barriers for our system, since it is the most appropriate type of sensor and it is the least complex one. We e-mailed our client to ask whether we can do this. No matter what the client’s reply will be, we still decided to focus on first having our app work with light barriers.

**Next meeting:**

Showing the progress on the work packages to the rest of the team and discussing who will do what for the remainder of the week.

## Bi-weekly report #3

App project team 10 – Sultanaev, Whitehall, Yu

**03.03.2015, 18:00**

### Topics discussed:

The client replied to us about our suggestion to only use light barriers for our project by saying that we should focus on the light barrier first, but to try and expand into using other sensors once that is done.

Yu and Whitehall were still working on their assignments and would show their progress tomorrow.

A small prototype light barrier circuit that works with the Engduino was successfully built.

### Next meeting:

Looking at the progress with the UI and the Engduino-TouchDevelop interaction.

**04.03.2015, 12:00**

### Topics discussed:

Yu could not manage to make the Engduino communicate with the TouchDevelop app, because it turned out that there actually is no library that would allow a TouchDevelop app to communicate with a device connected via USB with the functionality that we need.

Whitehall had progress with the UI in the sense that most buttons were already functional, however she had trouble coming up with a data structure that would allow the addition and removal of profiles by the user.

### Problems overcome:

We looked more closely at what libraries are available in TouchDevelop and found a wrapper for a cordova plugin that would allow us communication via Bluetooth, as well as the built-in Bluetooth functionality that is restricted to windows phones. Our initial approach was to start with USB and then move onto Bluetooth, but now we decided to switch exclusively to Bluetooth, since it would not be possible for us to write a library and then a TouchDevelop wrapper for that library which would allow us to communicate with the Engduino over USB. Our strategy now is to acquire the Bluetooth module for the Engduino as quickly as possible and start working with it. The team leader also scheduled a meeting with one of Engduino's designers in order to better familiarise himself with the Bluetooth module and how it works with the Engduino.

For the data structure that would store profiles in our app we decided to use the following structure: Collection of profiles that stores profile objects -> Profile object that stores the profile name and a collection of sensor objects associated with it -> Sensor object that stores the sensor name and the pins which it occupies on the Engduino in a string format.

**Next meeting:**

Discussing our findings with regards to the Bluetooth communication. And plans for the weekend.

**06.03.2015, 19:40**

**Topics discussed:**

Soultanaev met with one of the Engduino designers and learned about how the protocol that they created for communicating with the Engduino over Bluetooth works and shared what he learned with the rest of the team. We managed to get the Bluetooth communication work using various apps for Android downloaded from the Google Play Store, so we knew that there weren't any problems on the Engduino side.

We decided that over the weekend we will concentrate on getting the connection between a phone and the Engduino with a Bluetooth module sorted using the TouchDevelop wrapper for the cordova BluetoothSerial plugin and work on the profile management data structure.

**Next meeting:**

Showing the progress and discussing plans for the next week.

**10.03.2015, 09:00**

**Topics discussed:**

Despite our and other teams' efforts, we couldn't manage to get the TouchDevelop wrapper for the cordova BluetoothSerial plugin to work – the app always crashed when it attempted to form a connection with the Bluetooth device.

The development of the data structure required to store the user's profiles showed progress, but still had many bugs, and a lot of the discussion was about how to fix these bugs.

**Problems overcome:**

We contacted the author of the TouchDevelop wrapper for the cordova BluetoothSerial plugin to get some help. He suggested that we first attempt to build a minimal cordova app that uses the BluetoothSerial plugin and see if it works, in order to determine whether the issue is in the plugin or in the wrapper. We decided to follow this suggestion.

We also borrowed a smartphone with the Windows Phone operating system in order to try using the built-in Bluetooth functionality available exclusively for Windows Phone devices.

**Next meeting:**

Discussing the progress with the Bluetooth issue.

**11.03.2015, 12:00**

**Topics discussed:**

Our attempts at building a cordova app that uses the BluetoothSerial plugin as well as using the built-in Bluetooth functionality of TouchDevelop for windows phones turned out unsuccessful so far.

The profile management data structure is still being implemented, since the team members have had a lot to do with other modules and could not dedicate as much time to this project as expected.

**Problems overcome:**

We decided that it is still too early to decide that the cordova BluetoothSerial plugin is not working, and got in contact with the author of the plugin. We will be working with him in order to fix the problems we are having.

**Next meeting:**

Discussing the progress with the Bluetooth issue.

**13.03.2015, 17:30**

**Topics discussed:**

Through our communication with the author of the cordova BluetoothSerial plugin, we managed to successfully build a cordova app that uses the plugin, thus determining that the plugin is functional and that it is the TouchDevelop wrapper for the plugin which doesn't work.

**Problems overcome:**

We contacted the author of the TouchDevelop wrapper for the cordova BluetoothSerial plugin with this information and also sent him the debug logs of the TouchDevelop app running and crashing on an Android phone as evidence and to aid debugging. He admitted that it was his wrapper that didn't work and promised that he would fix it by Monday. We also asked him about the built-in Bluetooth functionality in TouchDevelop for Windows Phones, and he said that this functionality broke with the Windows Phone 8.1 update and suggested that we do not worry about Windows Phone at all for the time being.

Since it is still uncertain whether the Bluetooth communication will work by the time we have to show our milestone 2 demo, we decided to put our focus into making the UI as finished as possible in order to have something to show as our demo of the first build.

**Next meeting:**

Discussing our plans for getting ready for the milestone 2 demo.

## Bi-weekly report #4

App project team 10 – Sultanaev, Whitehall, Yu

18.03.2015, 19:00

### Topics discussed:

Through our communication with the author of the TouchDevelop wrapper for the cordova bluetoothserial plugin, we managed to test and get the wrapper working as it needs in order for us to be able to advance our project.

Although our app was looking visually adequate, little of the actual functionality was working and we had to decide what to do about it so that we have a proper demo working by the time the milestone 2 presentation is due. We also need an icon for our app.

Looking through the functionality supported by the Engduino protocol, we found out that the protocol doesn't actually support all of the functionality we need (although it had a basis for it, which is what got us to believe that it was available).

### Problems overcome:

We decided that it will be quicker to rewrite the app from scratch with the functionality in mind from the start, rather than try to modify the current build to implement the functionality. We will focus on just getting the functionality right, since there is not enough time to do any polish. Sultanaev and Yu will be working on this.

Whitehall was assigned to work on the icon for our app.

For the Engduino protocol, Sultanaev will write an initial version of the methods required for the app and send them to a member of the Engduino design team for evaluation. The methods should allow setting a pin type to input or output, write a high or low value to a pin, and send a package when the state of a pin changes.

### Next meeting:

Showing our progress moving towards milestone 2



**19.03.2015, 19:00**

**Topics discussed:**

The Engduino protocol functionality was implemented and sent to our contact at the Engduino design team for evaluation, who then made modifications to the methods in order to make them fit into the overall design of the protocol. We will be using his version of the methods.

Most of the functionality for the first build was implemented, and Whitehall was assigned to do some polishing to the visual aspect of the app.

Whitehall created an initial sketch for our app icon:



**Next meeting:**

Moving forwards after the milestone 2 demonstration

**20.03.2015, 17:30**

**Topics discussed:**

Our objectives now are to get the icon finalised, implement the rest of the functionality and do visual polishing on the app.

**Problems overcome:**

We decided that our workflow moving forward will be such that Sultanaev will implement a functionality in the app and Whitehall will polish that functionality while the next piece is being implemented. Yu will work on the conceptual parts of the functionality (research into algorithms and libraries to use) before Sultanaev starts implementing it.

Whitehall will also be working on the app icon.

Sultanaev's next task is to improve the pin selection interface.

Yu will be figuring out how to determine which of the user defined sensors was activated based on the package received from the Engduino.

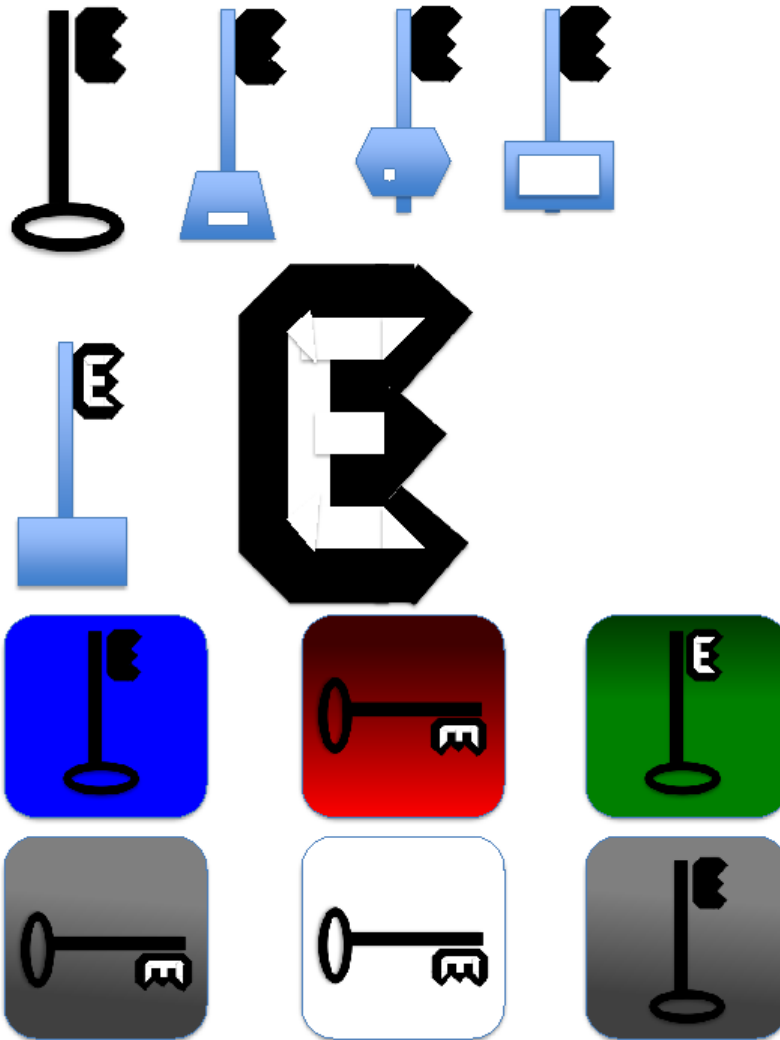
**Next meeting:**

Discussing the icon sketches

21.03.2015, 18:00

Topics discussed:

Whitehall submitted a collection of icon sketches to use:



We decided that the red and gray gradient backgrounds are the most promising and it is better to have the key vertically rather than horizontally.

Next meeting:

Showing off the progress and discussing what to do next

**25.03.2015, 18:30**

**Topics discussed:**

Our contact at Microsoft implemented serial communication on the PC platform, which meant that now we will be able to create a PC version of our app, so we will have to consider the modifications to the interface that will need to be made for the PC version.

We were also discussing the progress on the tasks we assigned to ourselves.

**Next meeting:**

Showing off the progress and discussing what to do next

**26.03.2015, 18:00**

**Topics discussed:**

There were many deadlines for the other projects, so there was not as much progress for this meeting and not that many things to discuss. We simply showed the small amount of progress that we had and made a few comments on how to move forward.

**Next meeting:**

On 27.03 we will get some more information from our client, and the next meeting will be dedicated to what our client tells us.

**27.03.2015, 19:00**

**Topics discussed:**

Our client told us that we will have to produce a 2 minute video by the 15<sup>th</sup> of April instead of the initial 29<sup>th</sup> of April. This means that some of the team roles had to be reassigned.

We decided that Whitehall will now be the lead for the video production and that will be her sole task.

Soultanaev will now do both the functionality and polish of the app and making final adjustments to the hardware.

Yu will still be working on researching the conceptual aspects of the functionality. He will also be working on user testing. Currently he is assigned to develop the system of associating sensors with profiles and the ability to add and delete sensors and their effect on the profiles with which the sensors is associated. He is also researching facebook notifications through TouchDevelop and writing up a step-by-step user testing procedure.

**Next meeting:**

Showing off the progress and discussing what to do next

## Bi-weekly report #5

App project team 10 – Soultanaev, Whitehall, Yu

30.03.2015, 18:00

Topics discussed:

Yu had some progress on the system that would allow associating sensors with profiles that would support the addition and deletion of sensors and profiles, but not a full implementation. We decided to let Soultanaev finish the work on this, since he is handling most of the app programming and has a better understanding of how all the components of the code work. Yu will continue his research into getting facebook notifications with our app and writing up a user testing procedure for our app.

Whitehall drew a storyboard for the scenes of our video and we discussed what to change:



Figure 16:Storyboard

We discussed the different icons that Whitehall showed previously and settled on using the gray theme. Whitehall will re-draw a few more icons in the gray theme and we will pick the final icon on the next meeting.

Whitehall will also write an initial draft for the script for the narration of our video.

We also managed to order a bluetooth module compatible with iOS devices, so that if we have the time we will be able to test our app on that platform as well.

**Next meeting:**

Picking the logo, discussing the video narration script

**31.03.2015, 17:00**

**Topics discussed:**

While we are waiting for the BLE module to arrive, we decided that Whitehall will install an iOS development kit and we will ask UCL for a subscription for her for the iOS developer programme, since she has an Apple machine. This way when we will have the time to start testing the iOS platform we will have all the required tools ready.

An final icon was picked for our app – the grey theme fits with the colour scheme of our app (which is also grey), the key represents the security theme and a part of the key is shaped like an Engduino:



Yu discovered that the library that simplifies the interaction between TouchDevelop and Facebook through oauth v2 is slightly out of date, and in order to create a Facebook that would have the permission to post on a user's wall we would need to submit a request for the permission which takes a long time to process. As a result we decided to drop this feature from our app project.

We decided to start filming the video on the 8<sup>th</sup> of April and Sultanaev will continue his work on all of the app functionality to get it ready for the filming.

**Next meeting:**

Discussing progress and plans for the future

### 03.04.2015, 17:00

#### Topics discussed:

Yu wrote a draft for the user testing procedure for our app. We did not discuss it very much, since the actual app is not fully finalised which means we cannot make more precise edits to the procedure, but having a draft ready will make the process easier in the future.

Whitehall was assigned to already get started on the intro and outro for our video, since these parts do not require any filming.

Soultanaev continues working on the functionality and fixing bugs.j

#### Next meeting:

Initial filming of the app

### 08.04.2015, 15:00

This meeting was dedicated to getting footage of our app and hardware in action for the video demonstration. We were experimenting with different set ups and seeing what works well on video and what does not. In the end we obtained a large amount of footage, most of which was not usable, but in the process we learned a lot about how to film our hardware in action and are more prepared for the second filming session which we decided to do on the 10<sup>th</sup> of April.

### 09.04.2015, 14:00

#### Topics discussed:

We met with our client to show him the progress that we had with our app and hardware.

The client was pleased with our demonstration and said that we need not worry about adding any new functionality and that we have met most of the requirements, even though we did not manage to get some of our stretch goals done such as facebook notifications.

The client commented that we should make it more obvious to the user how to progress through the app by adding labels such “step 1” to certain buttons and giving the user hints on what to do.

The client also gave us some suggestions regarding the video for our app, namely to first start with a “story” of our use case and then show how the app solves the problem.

#### Problems overcome:

In order to make the app more user friendly as the client requested, we decided to mark the elements of our home page with appropriate step numbers to make it easier for the user to

know how to get started. We also added prompts that for example tell the user to “Create a new a profile for their location by tapping add profile” if they haven’t created any profiles yet.

**Next meeting:**

Second filming session.

**10.04.2015, 15:00**

As with the 08.04 meeting, this meeting was dedicated to the filming process. We used what we learned in the first filming session and the suggestions from our client to film two use cases: one in a “work” context and another in a “home” context. In this session we only filmed the hardware in action, and in the next meeting we will film the app in action.

**11.04.2015, 13:00**

In this meeting we finished the filming of footage for our video demonstration. This time we filmed the app in action.

Whitehall was assigned to pick the best footage and edit together a rough draft of the video by 12.04 and to make edits to the script so that it fits the footage that we have.