Technische Universität München

Distributed Multimodal Information Processing Group

Prof. Dr. Matthias Kranz

Bachelor Thesis

Mobile Remote Control Application (Android) for Intelligent Environments

Mobile Fernsteuerungsanwendung (Android) für intelligente Umgebungen

Author: Markus Gsinn
Matriculation Number: [redacted]
Address: [redacted]
Advisor: Dipl.-Ing. Stefan Diewald
Prof. Dr. Matthias Kranz
Begin: 18 July 2011
End: 2 December 2011

Abstract

In this thesis about Home Automation, a survey about the use of mobile devices and future Home Automation systems as well as an example Home Automation application for Android devices had to be developed. The focus of the development of the Android application lay on designing a comfortable user interface and on finding an efficient way to import and display homes. For remote controlling devices, like light switches, equipment form the commercial manufacturer Home Easy has been used. Apart from that, an overview about contemporary techniques in the field Home Automation had to be given.

The resulting example Android application, called VMI Home Automation, is able to import building plans from manually created XML files and display them on the screen. The user interface enables the selection of room and floors on the map and provides several switching elements for the individual electric devices. Furthermore, the interface can be customized with little effort, as all graphical components are either created from XML or PNG files.
The online questionnaire developed in this thesis has been filled out by 35 persons. It revealed that the majority of the participants had a smartphone and that they are quite interested in Home Automation systems, though they have concerns about security and privacy. It has to be noted, that most of the people taking part in the survey had an age under 30 and were rather enthusiastic about technology.

Kurzfassung

Im Rahmen dieser Arbeit über Home Automation, sollte eine Umfrage zur Nutzung mobiler Geräte und zukünftiger Home Automation Systeme durchgeführt werden. Des Weiteren sollte eine Beispiel Home Automation Anwendung für Android Geräte erstellt werden. Der Schwerpunkt bei der Entwicklung der Android Anwendung lag im Design der Benutzeroberfläche und darin einen effizienten Weg zu finden um Gebäudepläne in die Anwendung zu importieren und darzustellen. Um Geräte, wie zum Beispiel Lichtschalter, fernzusteuern wurden Produkte des kommerziellen Anbieters Home Easy verwendet. Abgesehen davon sollte ein Überblick über die derzeit im Bereich Home Automation verwendeten Technologien gegeben werden.

Die entstandene Android Anwendung namens VMI Home Automation kann Gebäudepläne aus XML Dateien importieren und diese auf dem Bildschirm darstellen. Die Benutzeroberfläche ermöglicht die Auswahl von Räumen und Stockwerken und stellt mehrere Schaltelemente für die einzelnen Geräte bereit. Außerdem kann die Oberfläche mit wenig Aufwand umgestaltet werden, da alle grafischen Elemente aus XML oder PNG Dateien erzeugt werden.
Der in dieser Arbeit erstellte Onlinefragebogen wurde von 35 Personen ausgefüllt. Es kam dabei heraus, dass die Mehrheit der Teilnehmer ein Smartphone hatte und dass diese sehr interessiert an Home Automation Systemen sind, obwohl die Bedenken bei Sicherheit und Privatsphäre solcher Systeme haben. Es muss angemerkt werden, dass die Leute die an der Umfrage teilnahmen größtenteils unter 30 und eher technikbegeistert waren.

Contents

Chapter 1

Introduction

During the last years, the number of people using mobile devices like smartphones and tablet computers largely increased and will still grow further. Today it is no problem to access the internet on a mobile device via mobile communications or other wireless technologies from almost everywhere. Apart from communications, people use their smartphones for a large variety of other activities as there are applications for all possible and impossible things.

In addition, distributed wireless sensors and other electronic devices got cheaper and cheaper and are thus applied more often in new areas, for example in building technology for optimizing the energy consumption of electrical equipment. As these devices are usually battery-driven, a variety of new, low-energy wireless transmission techniques have been developed.

With the proliferation of mobile devices and other accompanying technologies, such as widespread, well-developed mobile communication networks, a lot of new ideas, possibilities and problems emerged. At the moment one very interesting and aspiring topic, made possible by all those developments, is the field of Home Automation. A lot of companies and institutes are working on automatizing and remote controlling all possible aspects of private homes and public buildings. Right now there are a lot of different technologies and approaches in the field of Home Automation. The recent developments have led to a lot of new approaches and will probably make Home Automation systems affordable for a larger group of the society.

In my bachelor thesis I will describe several existing and future technologies and present an example application as well as the results of a survey about the use of mobile devices and future Home Automation systems.

Chapter 2

State of the Art

2.1 What Home Automation is

Home Automation is a very broad field and there exist a lot of different definitions of what exactly Home Automation is in the literature. What most of those definitions have in common is, that a Home Automation system should at least be able to control heating, ventilation, air conditioning (HVAC) and lighting. In addition to that, it should collect sensor data, for example the temperature, for further processing or displaying. Some systems also support multimedia, health care and security applications. A Home Automation system enables the user to conveniently control his home via a user interface. This user interface can be a terminal, a personal computer, a remote control or, as discussed later, a mobile device. It is also possible that some processes of everyday life, for example the opening and closing of the window blinds, are automated. In most cases the system contains a central unit like a computer, a server or terminal to which all other units are connected. For communication between the individual parts of a Home Automation system a variety of different technologies is used, as there is not yet an established standard. Wireless as well as wired solutions are applied. Some of them are presented later in this chapter.

2.2 Different Communication Technologies

An important part of a Home Automation system is the communication technology that is used to exchange information between the individual components. Wired and wireless solutions exist. The wired systems are mainly bus systems. Though sometimes power line networking is mentioned, for example in [6], it is not described in this thesis as it is not without controversy in Europe. A possible problem could be the causing of disturbances at frequencies used for amateur radio.

2.2.1 Wired Technologies

The advantage of wired communication technologies is that they usually offer less delay and higher transmission rates than wireless solutions. Even though Home Automation systems do not require high data transmission rates, under some circumstances it could be useful, for example in multimedia applications. In addition to that, the signal can hardly be intercepted from outside the building.
According to [11] the most-used wired protocols for Home Automation in Europe are the European Installation Bus (EIB) and its successor KNX. Other important systems are Controller Area Network (CAN) and DomoBus, which was developed as an academic project and is described in [8].

2.2.2 Wireless Technologies

Apart from wired solutions, a large variety of wireless solutions for connection and communication between different devices, sensors and actuators exists. Wireless technologies offer a lot of advantages compared to wired technologies. According to [10] the installation costs are significant lower, as no cables have to be laid. Besides that, wireless components can be placed at locations where wiring would not be possible. Examples are moving elements or historical buildings, where no constructional changes are allowed. Furthermore, wireless networks can be reconfigured or expanded with less effort than wired networks. Another advantage is that mobile devices like PDAs, laptops or smartphones can connect to the network if the appropriate transceiver is integrated in the device.
One of the limiting factors of wireless devices is the power supply. As the majority of them is battery-driven, power consumption is an important criteria, whereas the transmission range and signal strength must nevertheless be sufficient. In the following, some common (Wi-Fi and Bluetooth) and some promising (ZigBee, Z-Wave and 6LoWPAN) wireless technologies will be presented and their employment in Home Automation systems will be discussed.

According to [10], Wi-Fi is not suitable for Home Automation systems as it was designed for applications that need high data rates, like video streaming, and therefore consumes too much energy. For the same reason, the classic Bluetooth technology is not an appropriate candidate. Even though there exists a low-energy version of Bluetooth (mentioned in [9]), it is not applicable as the basic wireless network technology in a home automation network. A transmission range of ten meters is not enough.
Another candidate for the use in Home Automation system is ZigBee. It is a networking technology, based on the IEEE 802.15.4 standard, developed by the ZigBee Alliance for low-data-rate and short-range applications [5]. ZigBee operates at the frequencies of 868 MHz (Europe) respectively 915 MHz (United States) and 2.4 GHz. The ZigBee specifica-

	ZigBee	Z-Wave	6LoWPAN
Radio frequency (in MHz)	868/915/2400	868/908/2400	868/915/2400
Range (in m)	10 - 100	30(indoor)/ 100(outdoor)	10 - 100
Bit rate (in kb/s)	20/40/250	9.6/40/200	20/40/250
Security	AES	AES-128	AES
Internet connection	gateway required	gateway required	no gateway required
Specification publicly available	yes	no	yes

Table 2.1: Important Characteristics

tions are publicly available.
Another possible alternative is Z-Wave. It is a proprietary, wireless protocol that was developed especially for home control applications [10]. It operates at 868 MHz in Europe and at 908 MHz in the United States. The latest generation of Z-Wave components can also use the 2.4 GHz channel.
The last possible candidate technology presented in this thesis is IPv6 over low power wireless personal area network (6LoWPAN). In [7] this technology is analyzed very detailed. 6LoWPAN is also based on the IEEE 802.15.4 standard and operates at the same frequencies as ZigBee. It has been designed for low power and low cost applications. In addition to that IPv6 is the core technology for the next generation of the Internet.
Table 2.1 shows an overview about some important specifications of ZigBee, Z-Wave and 6LoWPAN (the values for Z-Wave require 400 series chips or newer).

Given the large amount of papers that can be found about ZigBee and especially about ZigBee in context with Home Automation, this technology is perhaps the most promising candidate for being a standard wireless transmission technology for Home Automation system in the near future. 6LoWPAN has also be considered for the future as is supports IPv6, which enables the assignment of an IP address to any device in a home. The keyword here is “The Internet of Things”.

2.3 Example of a Home Automation System

So, how does a contemporary (more scientific) Home Automation system look like? [4] proposed a ZigBee based system with an integrated Wi-Fi network. The integration

is done via a home gateway, which consists of a ZigBee microcontroller and a Wi-Fi module, that connects to the home's Wi-Fi modem. Wi-Fi is needed for multimedia applications and for enabling the control of the Home Automation system via mobile devices, supporting Java. Those devices have to be connected to the local wireless network or have to access it via internet and the gateway. In addition, all devices can be controlled by a ZigBee remote control. All commands that are sent to the system from outside are checked by a "Virtual Home" for security issues before forwarding them to the system. The components of the Home Automation system communicate via ZigBee. Critical traffic can be encrypted with the AES algorithm. The devices that can be accessed are ZigBee operated lights, radiator valves and a safety sensor, which contains temperature, flame, carbon monoxide and smoke sensors.
Another sophisticated and more complex system, which even includes far-reaching security functions, is described in [11].

2.4 Mobile devices in Home Automation Systems

With the proliferation of personal digital assistants (PDA), smartphones and other handheld devices the idea of integrating them into Home Automation systems occurred. According to [3], it is the next logical step to combine "the mobile world with the technology of one's home to control most home appliances and systems en route wherever you are". Besides that, one of the proposals for improvement in [4] was to enable remote controlling by mobile devices without the need of a physical internet connection to the system. Another advantage of using smartphones or tablets in Home Automation systems is that those devices contain a lot of sensors and high quality touchscreen displays, which would be too expensive for permanently installed terminals. Like this, only one mobile device is necessary instead of several fixed ones at different locations.

In [1] a field study about the use of mobile computing devices in the context of their location in the home is described. As the "mobile computing devices" in this study were laptops, it would be interesting to study the use of smartphones and tablets, which are even more mobile, in order to determine how to integrate them into future Home Automation systems. [1] were tracking the position of the laptops and logging the actions the device was used for in several households for a couple of weeks. Precedent to that, they asked all the participants about where and how they use their devices and where in the house they spend most of their time. The results of the field study, a subsequent conversation with the participants and the results of the precedent questioning revealed, that in some cases there were discrepancies. This was mainly because people did not think about or remember certain habits or situations in which they did certain things.
Given the findings of this field study, an online questionnaire about the use of mobile

devices and future Home Automation systems has been developed in this thesis. This is the reason why the later introduced example application contains a logging function for a possible deployment in future user studies.

Chapter 3

Survey about Mobile Devices and Future Home Automation Systems

3.1 Composing the Questionnaire

One part of this bachelor thesis is a survey about the use of mobile devices and about how future Home Automation systems could be designed, respectively which functions the users want to have. The survey should reveal how many people are using smartphones or other mobile devices, how they are using them, if they could imagine using Home Automation applications, which functions they want to have implemented in Home Automation systems and what they think about security and privacy in such systems.
In order to retrieve this information an online questionnaire with 26 questions has been designed at www.soscisurvey.de. This website was used because it provided all necessary functions and tools free of charge for scientific purposes.

The questionnaire can be divided into three sections. The first section contained four questions to get some information of the social background of the people answering. This was necessary to be able to analyze the results accordingly.

The second part was about mobile devices. It contained the questions if the person answering already had a smartphone or tablet computer. If the answer was yes, it was asked which display resolution and screen size their device had. The goal was to determine how many people already have mobile devices and if yes, what they look like. Furthermore it was asked how many applications the people installed on their devices, for which activities they use it, where they use it in general and at which places they use it in their home, which is interesting for the design of Home Automation systems.

The third section was the actual Home Automation part. At the beginning it was asked whether the people taking part in the survey could imagine using Home Automation in the future and if they already possess equipment. Afterwards they should answer questions about which items in their home they wanted to control. It was also asked when, where and in which situations they wanted to do this. In addition to that, some gaps for free text input have been provided in order to let the participants communicate their own ideas. Another important issue was questions about security, privacy and problems with the management of multiple users. Besides some yes and no decisions, some spaces for free text input were provided again. Here, the people should describe how those problems could be solved and how they picture a future home automation system.

3.2 Results of the Survey

The online questionnaire has been completed by 35 persons in the period from 4 November 2011 until 20 November 2011. Figure 3.1 to 3.4 show the age, sex, highest completed education and the field in which the participants are active.

It can be seen that most of the people were between 21 and 24 and that two and half times more men than women took the survey. The highest completed education of the majority was the German "Abitur" or a completed study. A huge part of the participants stated that they are working in the field of engineering. Regarding the age and education, it can be assumed that a large amount of the participants were engineering students.

The mobile devices part of the survey revealed that about 60 percent of the participants own a smartphone (see Figure 3.5) but only four of them, which is a little bit more than ten percent, own a tablet (see Figure 3.6). As the number of people not answering the questions about screen sizes and display resolutions was equal to the number of people not having a smartphone or tablet, this number does not show up in the corresponding diagrams.
Regarding the screen size of smartphones (Figure 3.7), it can be seen that more than 80 percent of the participants answering the question and knowing their screen size, have devices with an at least 3.5 inch screen diagonal. The diagram about the screen resolution (Figure 3.9) does not show any trends towards a certain display resolution, but as a third of the people taking part at the survey voted for "I don't know" and in consideration of the results from the screen size question, it can be assumed that the majority of the participants have smartphones with medium and high display resolutions.
As the number of participants owning a tablet was comparatively small with four persons, there cannot be drawn any significant conclusions about the use of tablets yet.

The question about the number of applications installed on the mobile device (Figure 3.11) revealed that more than 85 percent of the people answering the question have five or more applications installed. One third of the participants have even more than 20

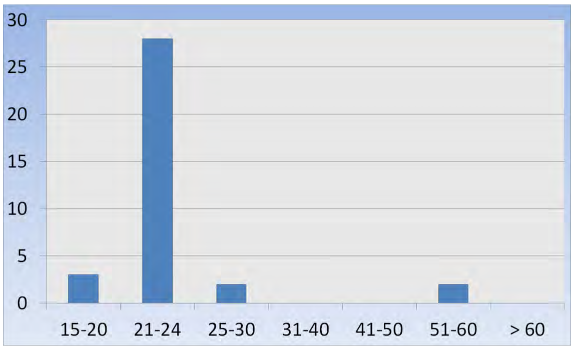

Figure 3.1: Age of the participants

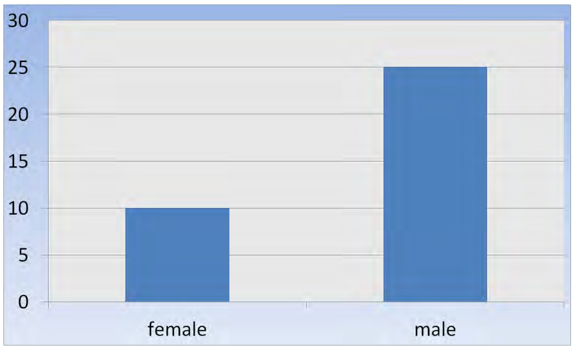

Figure 3.2: Sex of the participants

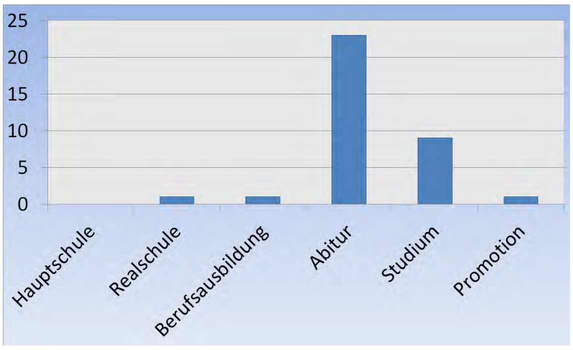

Figure 3.3: Education of the participants

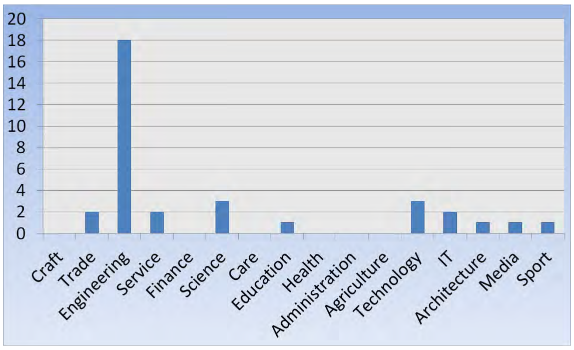

Figure 3.4: Fields of activity

programs on their smartphones or tablets. Apart from communication, like email, SMS and telephone, the devices are also very often used for image capture, navigation and administration of calendars. Other activities that should not be neglected are games and social networks. The most common answer in the fields left for other activities than the proposed ones, was the use of the devices as an alarm clock. The detailed diagram of the distribution can be seen in Figure 3.12.

Figure 3.13 shows that the people taking part in the survey use their smartphones or tablets at certain places, like at home or at work, as well as on the way. Almost half of them stated that they use their device abroad. The question about the location mobile devices are used at home (see Figure 3.14) revealed, that they are used almost everywhere except from the garage and the toilet. The favorite rooms are living room, bedroom and office. In the questionnaire the toilet was not one of the answers suggested, but some participants mentioned it in the field for own text input.

The Home Automation part of the questionnaire showed that only a very small part of the participants already has Home Automation equipment (Figure 3.15), but that a lot of them

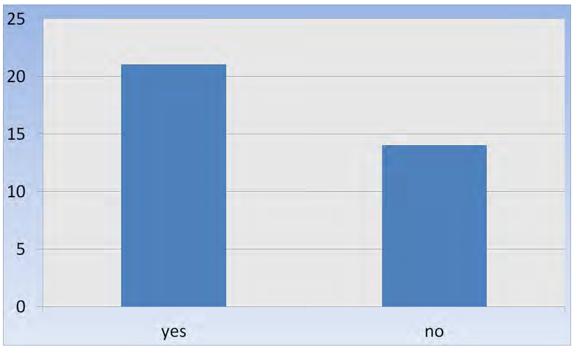

Figure 3.5: People possessing a smartphone

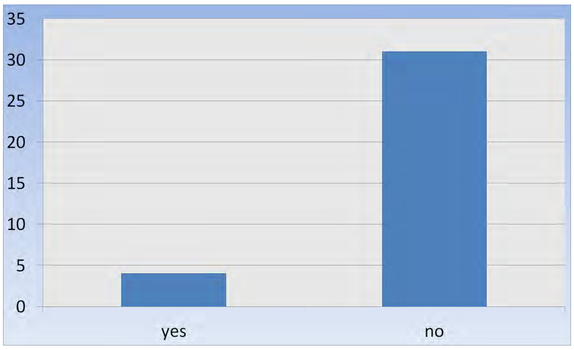

Figure 3.6: People possessing a tablet

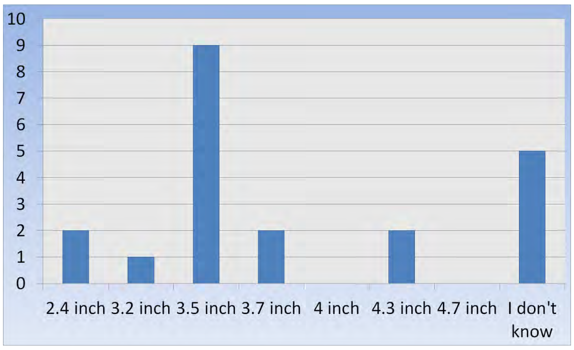

Figure 3.7: Smartphone screen diagonals

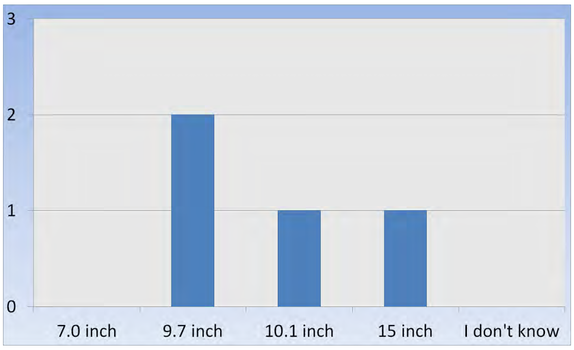

Figure 3.8: Tablet screen diagonals

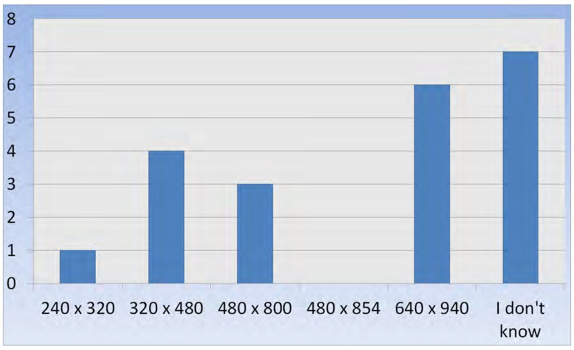

Figure 3.9: Smartphone display resolutions

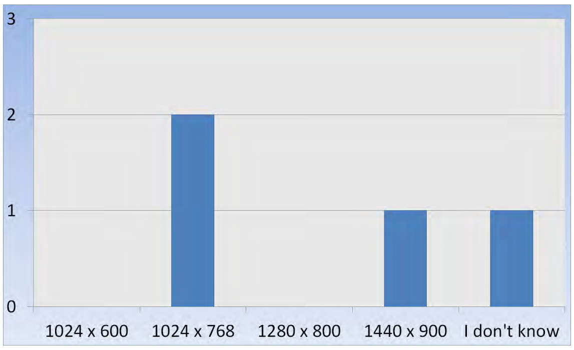

Figure 3.10: Tablet display resolutions

can imagine using mobile devices for future Home Automation applications like remote controlling household appliances (Figure 3.16). In Figure 3.17 it can be seen which devices the participants of the survey would like to control remotely, using their smartphones.

The next two questions dealt with the retrieving of real-time information about equipment

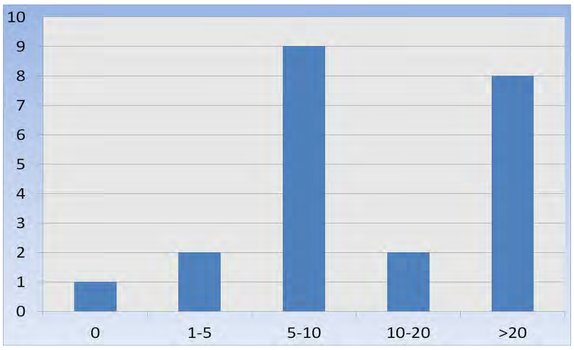

Figure 3.11: Number of applications installed on the device

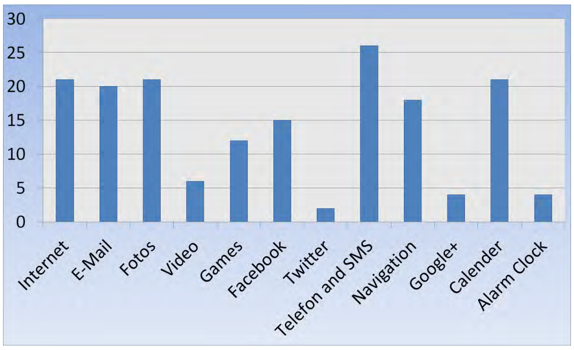

Figure 3.12: Activities and Services the device is used for

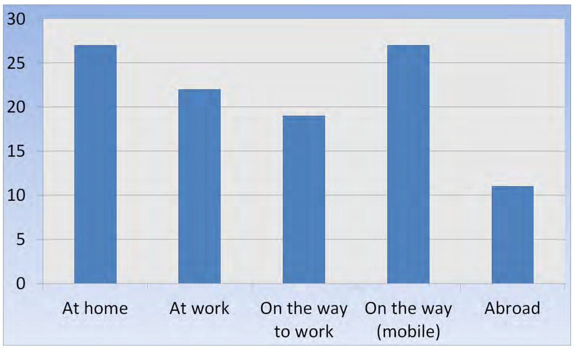

Figure 3.13: Places where the device is used in general

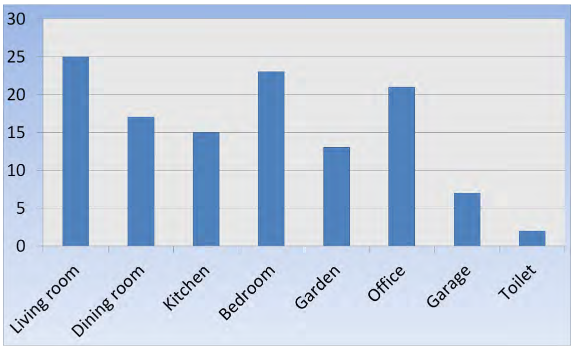

Figure 3.14: Places where the device is used at home

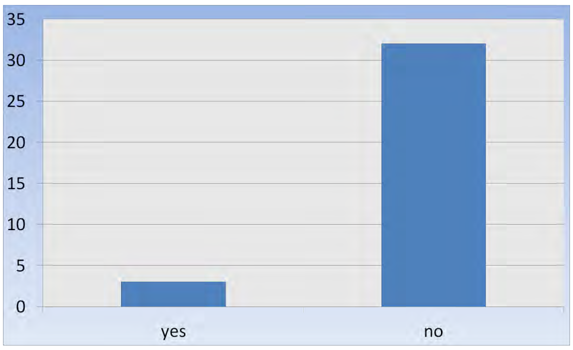

Figure 3.15: Number of people already possessing Home Automation equipment

in the home. Here again, the majority of the people answering the question would like to be able to get real-time information about their home in the future (Figure 3.18). Figure 3.19 shows what kind of real-time information is desired. One participant stated at the field for own text input, that he or she would like to be able to retrieve information about which persons are currently present in their residential community.

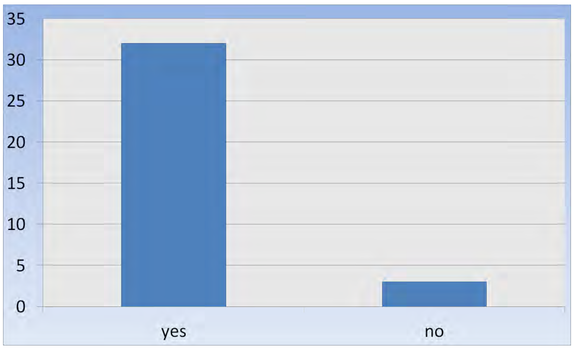

Figure 3.16: Number of people that could imagine using Home Automation in the future

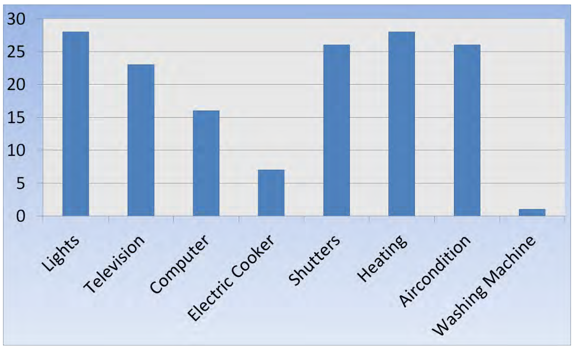

Figure 3.17: Devices people would like to remote control with mobile devices

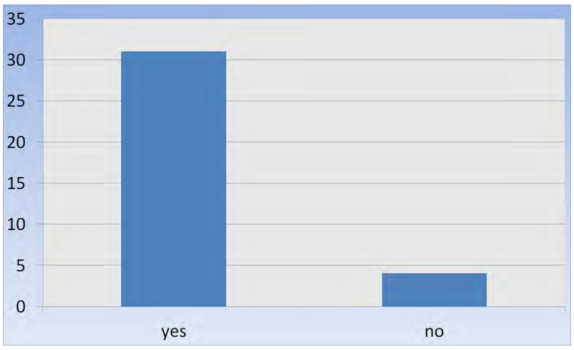

Figure 3.18: Number of people that would like to retrieve real-time data

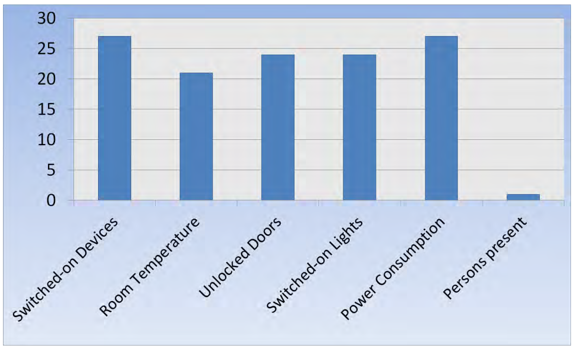

Figure 3.19: Real-time information desired to be retrieved

Figure 3.20 shows where real-time data is desired to be retrieved. This diagram is very similar to the one showing where people use their mobile devices. This question, where checkboxes could be set for the different locations, was followed by one where the participants had to describe with their own words in which situations and how often they wanted to access real-time data of their house. The answers about the retrieval frequency varied a lot, from “seldom” to “when necessary” to “three times a day” to “all the time”. The participants indicated that they would like to be warned if they leave doors unlocked or the cooker switched on respectively they wanted to check this information by their selves in situations when not being sure whether something is switched on or not.

The next two questions were about possible conflicts occurring when more people are using the system and how they could be resolved. Figure 3.21 shows the number of people considering conflicts with multiple users possible. The following question contained a field for own text input in order to let the participants describe how they would like the conflicts to be solved. The proposals were mainly a user administration with defined rights, location-based access rights and combinations of both. Location-based access rights

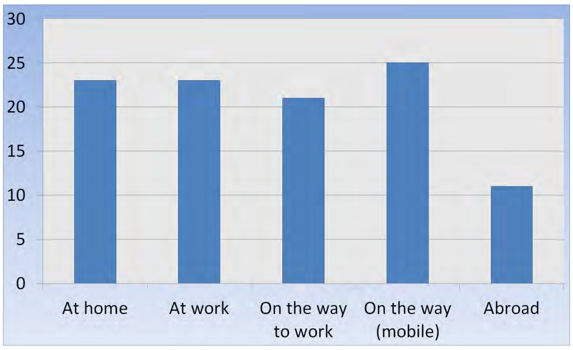

Figure 3.20: Locations where real-time informations are desired to be available

mean, in this case, that the person currently present in a certain room should have the rights to control the elements of this room instead of someone not present at the moment.

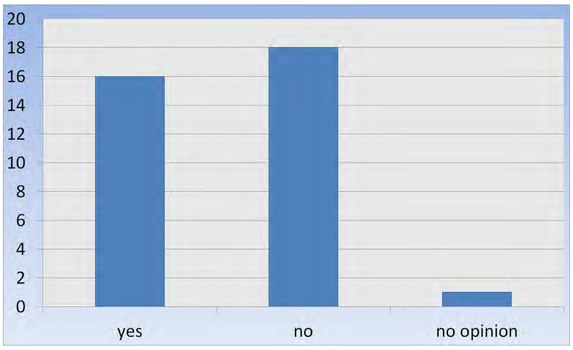

Figure 3.21: Number of participants seeing conflicts in multi user access

In Figure 3.22, it can be seen that 80 percent of the people taking part in the survey have concerns about the security or data integrity of Home Automation systems. The answers to the question about privacy threats in those systems show a similar result (Figure 3.23).

At the end of the online survey the participants should describe how they imagined the usage of a Home Automation system in a household with several people. The answers were very diverse, but most of them included that the system should be controlled by mobile devices. Some people preferred a central terminal or a terminal on each floor in order to avoid conflicts. Others favoured the combination of a central terminal and several mobile devices. It was proposed that certain parts of the systems should be protected with a password and that it should be made easy to add new devices to the system, for example via scanning a barcode.

Summarizing the results of the whole questionnaire, it can be seen that mainly people with an age under 30 years and mostly with an engineering background, took part in the

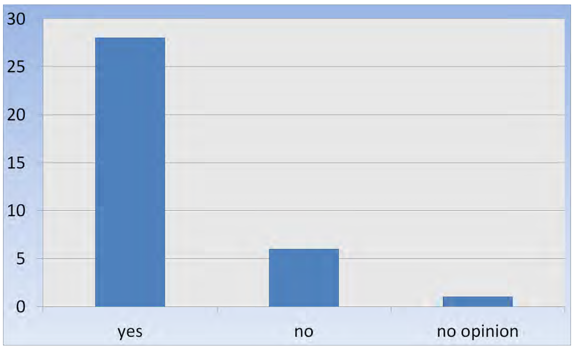

Figure 3.22: Number of people having concerns about the security of Home Automation Systems

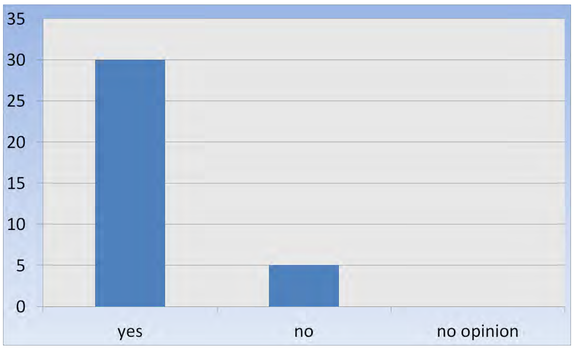

Figure 3.23: Number of people having concerns about the privacy in Home Automation Systems

survey. In general the participants have positive thoughts about future Home Automation technics, though the majority has concerns about security and privacy issues. The fact that 60 percent of the people taking part in the survey had a smartphone, but only four of them, which is about ten percent, had a tablet, should be considered for future Home Automation applications. The research about where people are using their mobile devices revealed that they are using them almost everywhere and for many different things.

Chapter 4

Home Easy

Home Easy is a commercial provider of Home Automation products. A large variety of devices, like remote controlled electrical sockets, lights and light switches and the corresponding remote controls are offered. Some examples of Home Easy products can be seen in Figure 4.1 to 4.6. According to the website of the German electronic mail order business Conrad Electronic, Home Easy devices transmit data at a frequency of 433 MHz or 433,92 MHZ and reach a range between 30m to 50m[1].

Home Easy equipment is used for the example Home Automation system in this thesis because it is shipped ready to use and no additional equipment like base stations or servers are required. In addition to that, the components can be integrated into existing environments without great effort. For example, a switch can be glued to any flat surface in a room. The corresponding light can be controlled via a remote control bulb holder that fits into standard sockets for lights. This is especially an advantage when setting up prototype rooms at the university, as far-reaching changes to the original installations are problematic. Beyond that Home Easy equipment is low priced and easy to obtain. By end of November 2011, the remote control unit used for this thesis (see Figure 4.5), and a remote controlled electrical socket with dimming function (see Figure 4.6) were 29.95 € respectively 19.95 € at Conrad Electronic's online shop. The remote controlling of Home Easy equipment with Android devices is explained in Chapter 5.

[1] http://www.conrad.de/ce/de/overview/2805041/HOMEeasy-Funk-Fernbedienungen

Figure 4.1: Remote control unit

Figure 4.2: Remote controlled light switch

Figure 4.3: Remote control bulb socket

Figure 4.4: Remote control door release kit

all pictures were taken from www.homeeasy.eu

Figure 4.5: HE853 USB remote control

Figure 4.6: Remote control dimmable socket

all pictures were taken from www.conrad.de

Chapter 5

VMI Home Automation

5.1 Overview of the Application

In the practical part of the bachelor thesis, an example Android Home Automation application, called “VMI Home Automation”, was developed. The hardware that was used to control was, as mentioned above, from Home Easy, as it can be attached to existing arrangements very quickly. It can be accessed via an USB remote control unit plugged into the Android device (see Figure 5.1). In addition to that, there has been research on this components at the Distributed Multimodal Information Processing Group. It is only necessary to call the “he85” binary executable for sending commands to the receiver units. The “he853” executable has to be in the “PATH” in an executable state. It responsible for the control of the USB transmitter.

Figure 5.1: Android device with Home Easy USB remote control

The main focus of the application development lay on finding a convenient and efficient way to describe a building, import this information into the application and to design an intuitive and comfortable to operate user interface. An additional requirement was to implement a logging function which can track the user's input and position during a potential future field study.
Figure 5.2 shows a simplified overview of how the application works. The central element is the "HomeActivity" class. It extends the Android Activity class and it is the only Activity existing in the VMI Home Automation application. All other classes are used or controlled by the "HomeActivity". It orders the XML Parser, which is called "FloorHandler", to load the necessary files from the SD card and to save the retrieved data in the variables of the local data representation (described in more detail in Section 5.3) for later employment. Another task is to react on user inputs and to interact with the user interface (described in Section 5.4). In addition to that, the "HomeActivity" calls the "UserInputLog" and supplies it with all necessary parameters, like for example location and time, in order to store this data in a SQLite database. This database can be exported to the SD card.

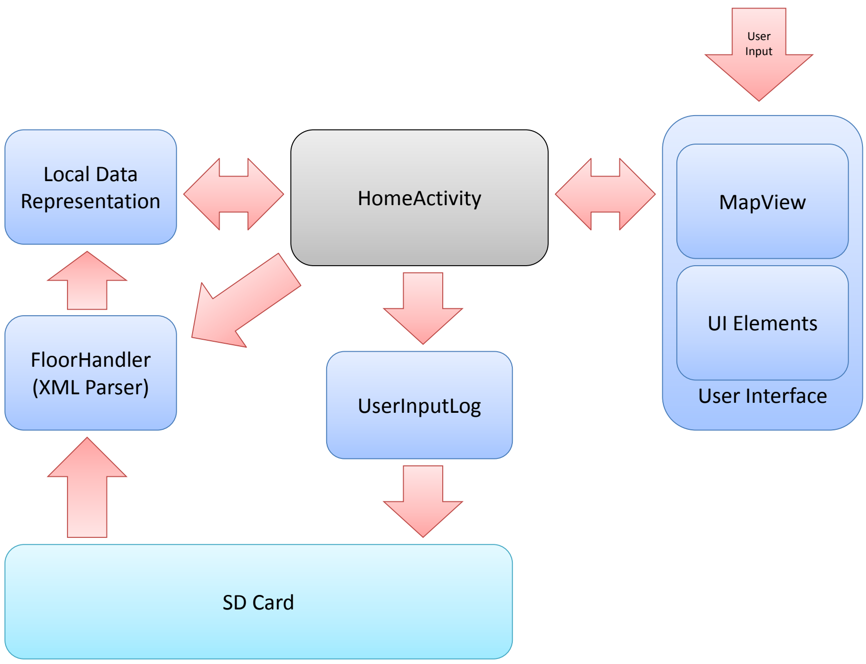

Figure 5.2: Simplified overview of the components of the application. The "HomeActivity" is the main entry point

In the following will be described how building plans can be imported into the application, how maps are displayed, how the user interface works and how the logging function is realized.

5.2 Describing Floors in XML

For importing the floor plan into the application the Extensible Markup Language (XML) format is used. It is applied because it is a very common file format that can be utilized very quickly and efficiently in Java code as there are different parsers and classes available. In addition, the XML format can be read comparatively easy by humans.
At the moment the XML file is created manually as there is not yet compatible software available for doing this directly from existing data like building plans.

Each floor of a building is described in a single XML file. The first layer in the XML is the floor element. Every floor consists of a certain number of rooms, hence the floor element in the XML file may contain several room elements. As each room may have several controllable items, such as lights, and non-controllable items, such as doors and windows, each room element in the XML includes tags for each of those elements. In the following an example XML of a floor is shown.

```xml
<?xml version="1.0" encoding="utf-8"?>

<floor name="Groundfloor" level="0">
  <room name="Office 1">
    <corners>0 0; 400 0; 400 300; 0 300;</corners>
    <window>100 0; 300 0;</window>
    <window>0 100; 0 200;</window>
    <light name="Office 1" id="001"/>
    <door name="Office 1" direction="in" hinge="left">
     200 300; 100 300;
    </door>
  </room>
  <room name="Meeting Room">
    <corners>400 0; 800 0; 800 300; 400 300;</corners>
    <window>500 0; 700 0;</window>
    <window>800 100; 800 200;</window>
    <light name="Light Meeting Room" id="002" dimmable="true"/>
    <light name="Light Board, Meeting Room" id="003" dimmable="false"/>
    <door name="Meeting Room" direction="in" hinge="left">
     400 200; 400 100;
    </door>
    <beamer name="Beamer Meeting Room" id="764">
     410 200;
    </beamer>
    <aircondition name="Aircondition Meeting Room"/>
    <heating name="Heating Meeting Room"/>
  </room>
  <room name="Office 2">
```

```
    <corners>550 300; 800 300; 800 600; 550 600;</corners>
    <window>800 400; 800 500;</window>
    <light name="Office 2" id="576"/>
  </room>
  <room name="Lobby">
    <corners>0 300; 550 300; 550 600; 0 600;</corners>
    <window>0 400; 0 500;</window>
    <light name="Light Lobby" id="235"/>
    <door name="Office 2" direction="out" hinge="left">
     550 500; 550 400;
    </door>
    <door name="Main Entrance" direction="in" hinge="right">
     200 600; 300 600;
    </door>
  </room>
</floor>
```

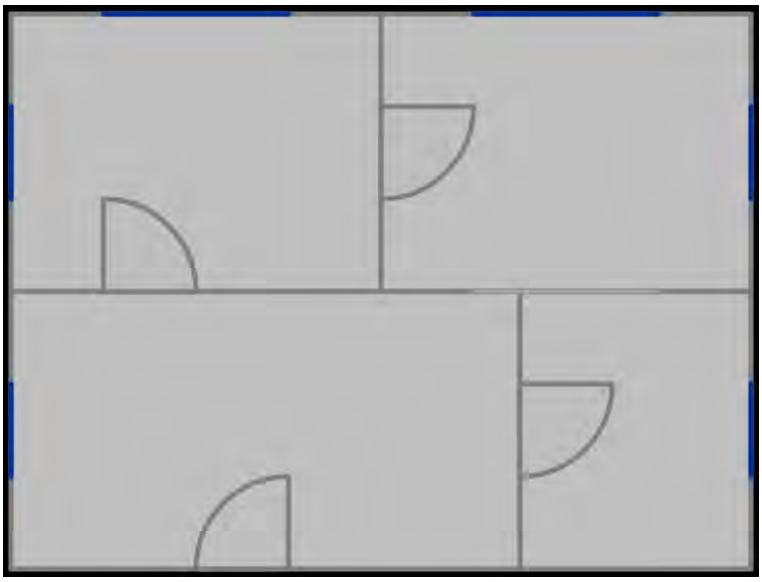

Figure 5.3: Floor described via the XML above

The floor element has two mandatory attributes for assingning a name and a level to each floor (see Table 5.1). A floor element may only contain one type of children, a room, whereas the amount of children is arbitrary (see Table 5.2).

The room element contains only one attribute for assigning a name to the room (see Table 5.3). The coordinates of the room corner points are stored between the tags of

Name	**Value**	**Mandatory**	**Default**
name	String	yes	none
level	Integer	yes	none

Table 5.1: <floor>'s attributes

Name	**Mandatory**	**Quantity**
room	no	arbitrary

Table 5.2: <floor>'s children

the corners element inside the room element. Each single point coordinate is a pair of two integer values that are separated by whitespace. Individual points are separated by semicolons. This representation is used for all coordinates in the XML format. The origin of the coordinate system which is used to map the floor plan is in the top left corner. The x coordinate describes the horizontal position, the y coordinate describes the vertical position. As the coordinates are stored as integer values, the unit centimetre is used in the example XML file. If this is not accurate enough, any other unit can be used as long as it is linear and anything but integer values are used. For example, millimetres instead of centimetres could be used to describe the above mentioned floor with higher accuracy.

Aside from the corners element, each room element may contain several additional predefined elements for each object a certain room could contain (see Table 5.4). The current version of the application supports the following "room elements" in the XML file:

- light
- door
- window
- beamer
- screen
- air condition
- heating

As the Application features the "RoomElement" class as common base class for all room elements, it has to be assured, that each subclass is able to assign a name. If this name

Name	**Value**	**Mandatory**	**Default**
name	String	yes	none

Table 5.3: <room>'s attributes

Name	**Mandatory**	**Quantity**
corners	yes	one
light	no	arbitrary
door	no	arbitrary
beamer	no	arbitrary
screen	no	arbitrary
aircondition	no	arbitrary
heating	no	arbitrary

Table 5.4: <room>'s children

has to be assigned manually, it has to be accounted for that case in the XML.

The light element possesses, except from the name, two more attributes (see Table 5.5). One for setting a Home Easy id and one for defining whether the light is dimmable or not. A light element does not have any children.

For drawing doors onto the map several information like the opening direction of the door and the position of the hinges are necessary. These data is stored as attributes in the door element of the XML (see Table 5.6). The opening direction is defined as the direction in which the door will open, observed by someone entering the specified room. The value "in" means that the door will move into the room while opening, the value "out" means that it will move towards the observer and "out of the room" while opening. In addition

Name	**Value**	**Mandatory**	**Default**
name	String	yes	none
id	int	yes	none
dimmable	boolean	no	false

Table 5.5: <lights>'s attributes

Name	**Value**	**Mandatory**	**Default**
name	String	yes	none
direction	String	yes	none
hinge	String	yes	none

Table 5.6: <door>'s attributes

Name	**Value**	**Mandatory**	**Default**
name	String	yes	none
id	int	yes	none

Table 5.7: <screen>'s attributes

to that, the position of the door has to be known. Therefore a start point and an endpoint are assigned to each door. The start point is located at the opposite side of the hinge, the end point is on the same side as the hinge. The coordinates of those points are stored between the XML tags of the door element. The values are separated in the same format as the corner coordinates of the floor. A door element does not have any children.

Example of a door element:

```
<door name="Office 2" direction="out" hinge="left">550 500; 550 400;</door>
```

Plotting windows in the map is a very basic task. Only the two limiting points of the window are required. This is again realized via a pair of coordinates between the window element tags in the XML file as described above. As the window class inherits from the room element class, it has to set a name in the constructor. But since a window does not need a name right now, the value is set to null in the constructor used by the XML Parser. A window element does not have any attributes or children.

Example of a window element:

```
<window>100 0; 300 0;</window>
```

The screen element, for representing projection screens, is nearly similar to the window element. But it needs an attribute for assigning a Home Easy id in the XML (see Table 5.7). A screen element does not have any children.

Example of a screen element:

```
<screen name="Screen Meeting Room" id="354">700 300; 500 300;</screen>
```

Name	Value	Mandatory	Default
name	String	yes	none
id	int	yes	none

Table 5.8: <beamer>'s attributes

The beamer element consists, aside from the name attribute required by the base class and the attribute for the home easy id, of a single coordinate between the XML tags to indicate its position on the map (see Table 5.8). A beamer element does not have any children.

In the current version of the app, the air condition and heating element support only the features of the room element base class, as there is not yet any controllable hardware available for testing.

5.3 Representing the Physical World in Code

5.3.1 Hierarchy

In order to use the information about a building and all its items efficiently and in order to safe states or similar of certain objects in the room or on a floor, the necessary data has to be described useful by the corresponding classes. As a building is described by XML files, each depicting a single floor, the entity of a floor is chosen as the model for the "Floor" class, used for storing all the data of a level. As a structure usually consists of several floors, the application stores them in an "ArrayList", which can be sorted by the level of the floors with an own implementation of the Java Comparator interface, called "FloorComparator".
The Java "ArrayList" class is also the technique used for assigning rooms to a floor and the individual items to a room. Figure 5.4 shows the hierarchy of the different classes used for the representation of floors in the application. In the following, the individual classes will be described in more detail.

5.3.2 The Floor Class

The "Floor" class is, as mentioned above, the basic element for storing the information extracted from the corresponding XML file. It posses attributes for assigning a name and a level to it. The level value is needed for sorting several "Floor" elements, if more than one is imported. Additionally, each instance contains an ArrayList of all the rooms on a floor. The UML diagram of the "Floor" class can be seen in Figure 5.5.

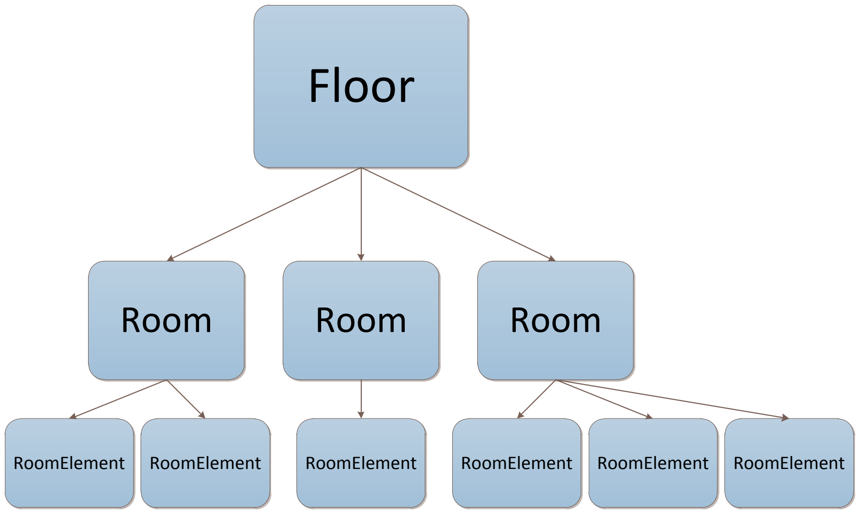

Figure 5.4: Hierarchy of the classes representating a floor

5.3.3 The Room Class

One step underneath the "Floor" class in the data hierarchy is the "Room" class. It contains attributes for assigning a name and for marking whether a room is selected by the user or not. In addition to that, the "Room" class includes two "ArrayLists", one for storing the coordinates of all the corner points and one that contains all the elements that are available in a certain room. An UML diagram of the "Room" class can be seen in Figure 5.6.

5.3.4 The RoomElement Class

The bottom layer of the data hierarchy consists of elements of the "RoomElement" class. This is the basic class for all elements a room may contain. It contains only one attribute for assigning a name. As this attribute has to be set in the Constructor, all classes that inherit from "RoomElement" have to implement the method for setting a name. Any item of a room has its own corresponding Java class that is derived from "RoomElement". This is necessary for making it possible to store all elements of a room in one "ArrayList" of a "Room" object on the one hand, and for enabling an accurate enough adjustment of a Java class to all aspects of the item in the real world on the other hand. Figure 5.7 shows a UML diagram of the "RoomElement" class with all its children.

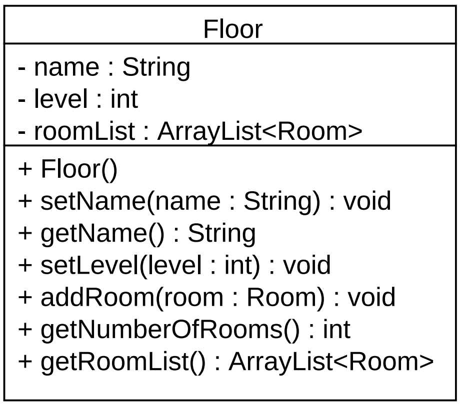

Figure 5.5: UML diagram of the “Floor” class

5.4 Composition of the User Interface

5.4.1 Overview of the User Interface

Since the user interface is a very important part of the VMI Home Automation application, it will be explained in more detail in this chapter. Before the functionality is described, there will be given a quick overview of the general layout.
The aim was to design a user interface that is intuitive, comfortable to use and that visualizes the surroundings with its elements in an easy understandable and convenient way.
The interface can technically be separated into three columns (see Figure 5.8), each realized by an Android LinearLayout element. The left column contains buttons for selecting individual elements of one type on the whole floor, for example, all lights on the first floor. The middle part can be separated into two pieces, which are again realized by LinearLayouts. The top layout, which is called “FloorSelector” in the following, contains buttons for selecting one of the floors in the building and an Android TextView for displaying the name of the currently selected floor. The rest of the space in the middle column is occupied by the so called “MapView”. It is responsible for displaying the building plan, selecting rooms and visualizing selections. The “MapView” is explained in more detail in Section 5.5. The control elements of the selected objects, such as light switches, are located in right column. Due to layout and design reasons, the application supports only landscape orientation.

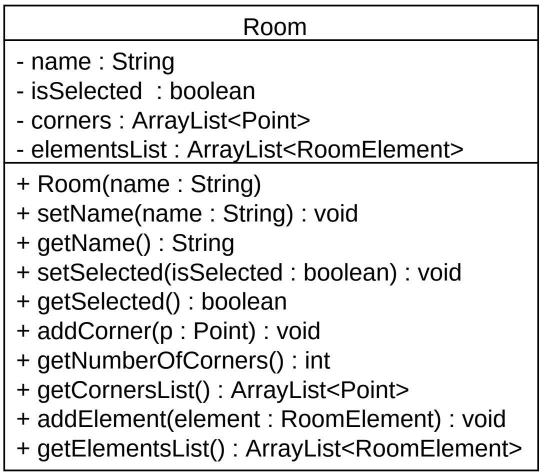

Figure 5.6: UML diagram of the “Room” class

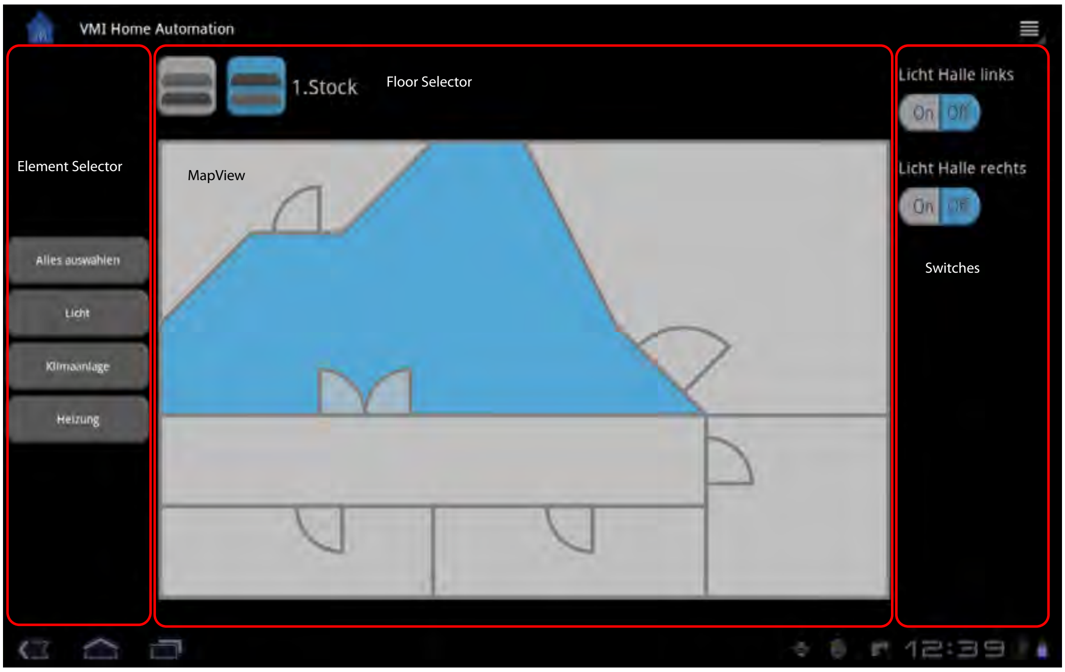

Figure 5.8: The user interface can be divided into 3 columns (marked red)

5.4.2 The Behaviour of the User Interface

If the application is able to load the correct XML files from a folder called “Home Automation” on the SD card in the “onCreate” method, the program can be used immediately. If there are no files available or the files have changed, the new ones have to

be stored on the SD card in the specified folder. When this is finished, the XMLs can be loaded again by the user via an entry in the options menu. The options menu is located in the so called “Action Bar” in the top right corner of the screen, like it is the standard in Android 3 applications[1].
When this is done, the user can start using the application. On the left side (see Figure 5.8), all controllable elements of a certain type on the whole floor can be selected, for example all lights. The “MapView” will now highlight rooms on the map where those items are located. On the right side of the screen, the elements for controlling the selected “RoomElements” are situated. This can be, for example, on-off switches or dimmer switches. Another way to use the Home Automation equipment is to click on the desired room. The “MapView” will then highlight the selected room and all controls for the elements located inside the room can be selected in the column on the right.
For selecting another floor, the “FloorSelector” above the “MapView” is responsible. It can supply buttons for switching between two or three floors. In the case of one floor, it will only display a name as no choice between different levels is necessary. If there are more than three floors available, a dropdown menu will be provided.

5.5 Drawing Maps

5.5.1 Calculating available Space and Scaling

A very essential part of the VMI Home Automation application is the “MapView”. It is responsible for displaying floors and allows the user to select rooms. This class extends the Android “View” class and overrides its “onDraw” and “onMeasure” method.
In “onMeasure” the size of the canvas, on which the map will be drawn, has to be determined. It shall use as much space as possible. The amount of this space is defined in the “onCreate” method of “HomeActivity”. In horizontal direction, the “MapView” may use all the space that is not needed by the “LinearLayouts” right and left of it. In vertical direction, some space for the “FloorSelector” plus some offset is reserved (see Figure 5.9). For assigning a certain size to each of the elements mentioned above, the unit density-independent pixel (dp) is used, as the application may run on different devices with a different pixel density, measured in dots per inch(dpi)[2]. A density-independent pixel is defined as a physical pixel on a 160 dpi screen. To determine the amount of “real” pixels Formula 5.1 is used and implied in “HomeAcitivity” (see Listing 5.1).

$$\text{px} = \text{dp} * (\text{dpi}/160) \tag{5.1}$$

[1]http://developer.android.com/guide/topics/ui/actionbar.html
[2]http://developer.android.com/guide/practices/screens_support.html

Listing 5.1: Calculating the space available for the map

```
FREE_DISPLAY_WIDTH = DISPLAY_WIDTH
        - ((int) ((res.getDimension(R.dimen.linearLayoutLeftWidth)
        + res.getDimension(R.dimen.linearLayoutRightWidth) +20.0f)
        * (WIDTH_DPI / 160.0f)));
FREE_DISPLAY_HEIGHT = DISPLAY_HEIGHT
        - ((int) (200.0f * (HEIGHT_DPI / 160.0f)));
```

The first thing that is done in “onDraw” is to subtract some pixels from the maximum canvas size to get some distance to the border, to avoid the map joining the control elements for providing a better appearance of the whole application. Afterwards the ratio between the map size, defined in the unit used in the XML file of the current floor, and the size of the canvas, in pixel, is calculated in horizontal as well as in vertical direction (see Formulas 5.2 and 5.3).

$$\text{hratio} = \frac{\text{width of the map}}{\text{width of the canvas}} \tag{5.2}$$

$$\text{vratio} = \frac{\text{height of the map}}{\text{width of the map}} \tag{5.3}$$

These two numbers are needed for determining whether the map size in comparison to the canvas size is larger in vertical or in horizontal direction. Based on this result the same scaling factor can be selected in the horizontal (see Listing A.1 in the appendix)and the vertical scaling function (see Listing A.2 in the appendix), for not distorting the proportions of the map and for not placing any elements wrong. The scaling functions are needed for calculating the ”canvas coordinates” of a ”real world” point.

The mathematical formulas that are used for the scaling functions can be seen in Formulas 5.4 to 5.7. In Table 5.9 is given a short overview of the meaning of all the variables used. For a better understanding Formula 5.4 will be explained in more detail. It is used for scaling in horizontal direction when the value of the ratio between the width of the map and the width of canvas is higher than (or equal to) the value of the ratio between the height of the map and height of the canvas.
The first step is to move the floor plan towards the left side. This is done by subtracting the minimal occurring coordinate value in horizontal direction from every horizontal coordinate value. The next step is to perform the actual scaling. In this case, the result is multiplied with the ratio between the width of the canvas and the width of the map, for fitting the map into the canvas and using the complete canvas size in horizontal direction (minus the offset towards the borders on both sides). The final step is to add an offset to each coordinate for moving the scaled map away from the left border. As this offset

was subtracted two times from the maximum canvas size, the floor plan will now have the same distance to the border on each side.

The other equations work similar. The only exception is Formula 5.6, as there are made some adjustments to center the floor plan in vertical direction instead of moving it to the border.

In some cases, for example for detecting click events on the map, it is important to know the coordinates from the XML file. For this reason public functions to recalculate these coordinates out of the ones used on the canvas are also implied in “MapView”. As this only requires to resolve the above mentioned equations by another variable, those functions are not explained in more detail.

$$\text{if wratio } \geq \text{vratio:} \quad x_{canvas} = (x_{map} - x_{min}) \cdot \frac{\text{width}_{canvas}}{\text{width}_{map}} + \text{offset} \tag{5.4}$$

$$\text{if wratio } < \text{vratio:} \quad x_{canvas} = (x_{map} - x_{min}) \cdot \frac{\text{height}_{canvas}}{\text{height}_{map}} + \text{offset} \tag{5.5}$$

$$\text{if wratio } \geq \text{vratio:} \quad y_{canvas} = (y_{map} - y_{min}) \cdot \frac{\text{width}_{canvas}}{\text{width}_{map}} + \text{offset} + \text{centering} \tag{5.6}$$

$$\text{if wratio } < \text{vratio:} \quad y_{canvas} = (y_{map} - y_{min}) \cdot \frac{\text{height}_{canvas}}{\text{height}_{map}} + \text{offset} \tag{5.7}$$

$$\text{centering} = \text{height}_{canvas} - \frac{\text{width}_{canvas}}{\text{width}_{map}} \cdot \text{height}_{map} \tag{5.8}$$

5.5.2 Drawing the Map

After the size of the canvas is determined the drawing of the actual map can be started. All coordinates that are used in the “onDraw” method have to be scaled by the above mentioned functions. The first step is to draw the individual rooms on the map. This is done by iterating over the list of rooms of the current floor. Each room has, as mentioned earlier, a list of corners over which is also iterated. Starting from the first point in this list, a path is created. This is done by using the Android “Path” class. After the path is closed, it is drawn with two different “Paints”. Objects of the type “Paint” are needed for drawing onto a canvas and can be adjusted in many ways (see Listing 5.2). In this case, the color of the “Paint” object is defined in the colors XML of the res/ folder of the Android application. This simplifies subsequent modifications. The “Paint” of the path with the style set to “stroke” is used for drawing the walls, the other one, with the style

Variables		
Name in formula	Name in Code	Meaning
x_{canvas}	result	horizontal coordinate on the canvas
y_{canvas}	result	vertical coordinate on the canvas
x_{map}	x	horizontal coordinate on the map
y_{map}	y	vertical coordinate on the map
x_{min}	min_x	the smallest coordinate value on the map in horizontal direction
y_{min}	min_y	the smallest coordinate value on the map in vertical direction
width_{map}	map_width	the map size in horizontal direction
height_{map}	map_height	the map size in vertical direction
width_{canvas}	canvas_width	the maximum canvas size in horizontal direction minus two times a certain offset
height_{canvas}	canvas_width	the maximum canvas size in vertical direction minus two times a certain offset
offset	DISTANCE_TO_BORDER	the offset that is added on each side of the towards its borders
centering	FREE_CANVAS_HEIGHT	the value needed for centering the floor plan in vertical direction

Table 5.9: Variables used in formula 5.4 to 5.7

set to “fill”, is employed for marking the inner part of the room. Depending on whether the room is selected by the user or not, another color for the interior of the room is chosen. The whole code for drawing rooms can be seen in Listing A.4 in the appendix.

Listing 5.2: How different “Paints” are modified in code

```
roomPaintSelected = new Paint(Paint.ANTI_ALIAS_FLAG);
roomPaintSelected.setColor(r.getColor(R.color.room_color_selected));
roomPaintSelected.setStyle(Paint.Style.FILL);

wallPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
wallPaint.setColor(r.getColor(R.color.wall_color));
wallPaint.setStrokeWidth(5.0f);
wallPaint.setStrokeJoin(Paint.Join.ROUND);
wallPaint.setStyle(Paint.Style.STROKE);
```

The next step, after the rooms have been finished, is to draw other “RoomElements” on the canvas. In the current version of the VMI Home Automation application following elements can be visualized:

- Screens
- Windows
- Doors

Drawing windows and screens works the same way as drawing rooms, a path between two points is drawn. Depending on whether a screen is down or not, or whether a window is closed or not, another color is used. As there is not yet sensor data about windows available, this feature is not used for windows right now.

Displaying doors on the map is a more complex task, as there have to be considered some parameters and cases. As mentioned in Section 5.2, a door is assigned a start point, an end point, an opening direction and a hinge position. From these parameters has to be determined the orientation and the correct position of the door on the map. In order to do that, four cases are distinguished, depending on the two values for the opening direction as well as the hinge position. After this is done, the orientation of the door is determined. In order to avoid unnecessary calculations or a division by zero, the cases where a door has to be placed exactly horizontal or vertical on the map are excluded via if statements. In any other case the angle has to be calculated via the inverse tangent function. The necessary parameters for using this function are determined via the coordinates of the start point and the end point (see Figure 5.10). How this is realized in Java code can be seen in Listing A.3 in the appendix. Case 1 in the source code is for example applied to rotate a door with its hinge on the left side and an opening direction into the room.

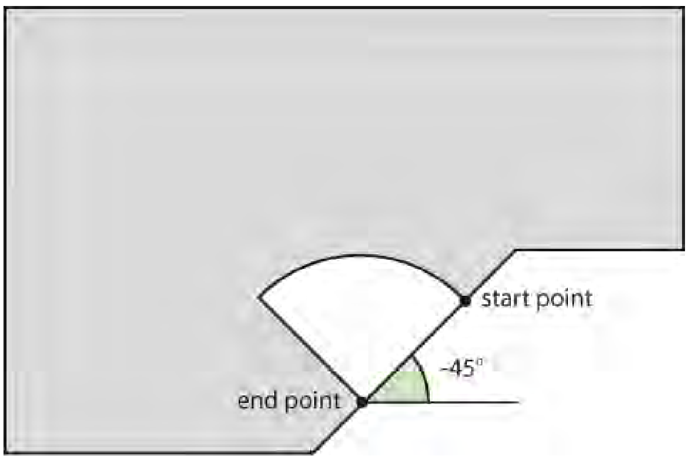

Figure 5.10: Example where the door has to be rotated

The actual drawing is performed by the "drawArc" function of the Android "Canvas" class, which can be used to draw ovals, or in this case a quarter circle, like doors a visualized on floor plans. Therefore the bounds of the full circle have to be limited via an Android "RectF" object, which basically is a rectangle. The rectangle is defined by setting values for its top, bottom, left and right border. These values are calculated by adding the size of the door, scaled on the canvas, to the center point of the circle, which is the same point as the hinge position or the end point (see Figure 5.11). It has to be paid attention to the fact that the origin of the canvas coordinate system is in the top left corner. The whole code for drawing doors can be seen in Listing A.5.

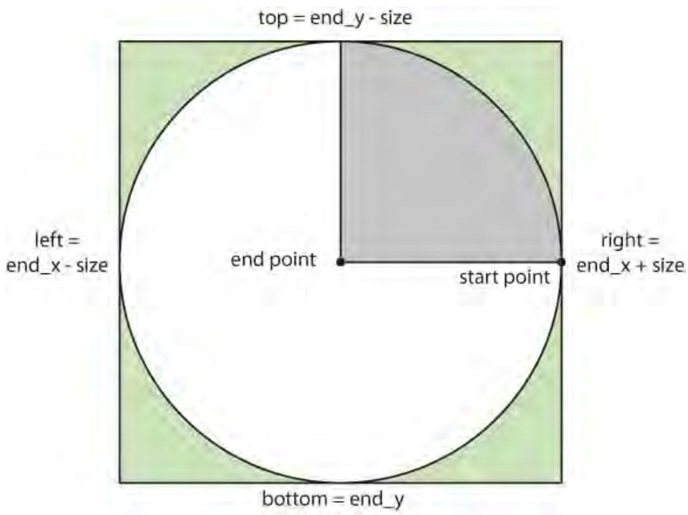

Figure 5.11: How a "RectF" bounding box is set

5.6 Logging User Inputs

For enabling the logging of user inputs in a potential field study, a SQLite database is used. It consists of seven columns for storing the required data. There are columns for the name of the used control element in the application (for example a light switch button), the name of the executed action (for example switching the light off), the time and the date. Additionally, there are three columns for saving the location. These columns are longitude, latitude and altitude. For determining the position Skyhook[3] is integrated into the application. The Skyhook software development kit (SDK) can be imported conveniently into Android applications. Skyhook tries to determine the position via nearby Wi-Fi access points or via GPS. As the application will usually be used indoors, only Wi-Fi can be used.

[3]http://www.skyhookwireless.com

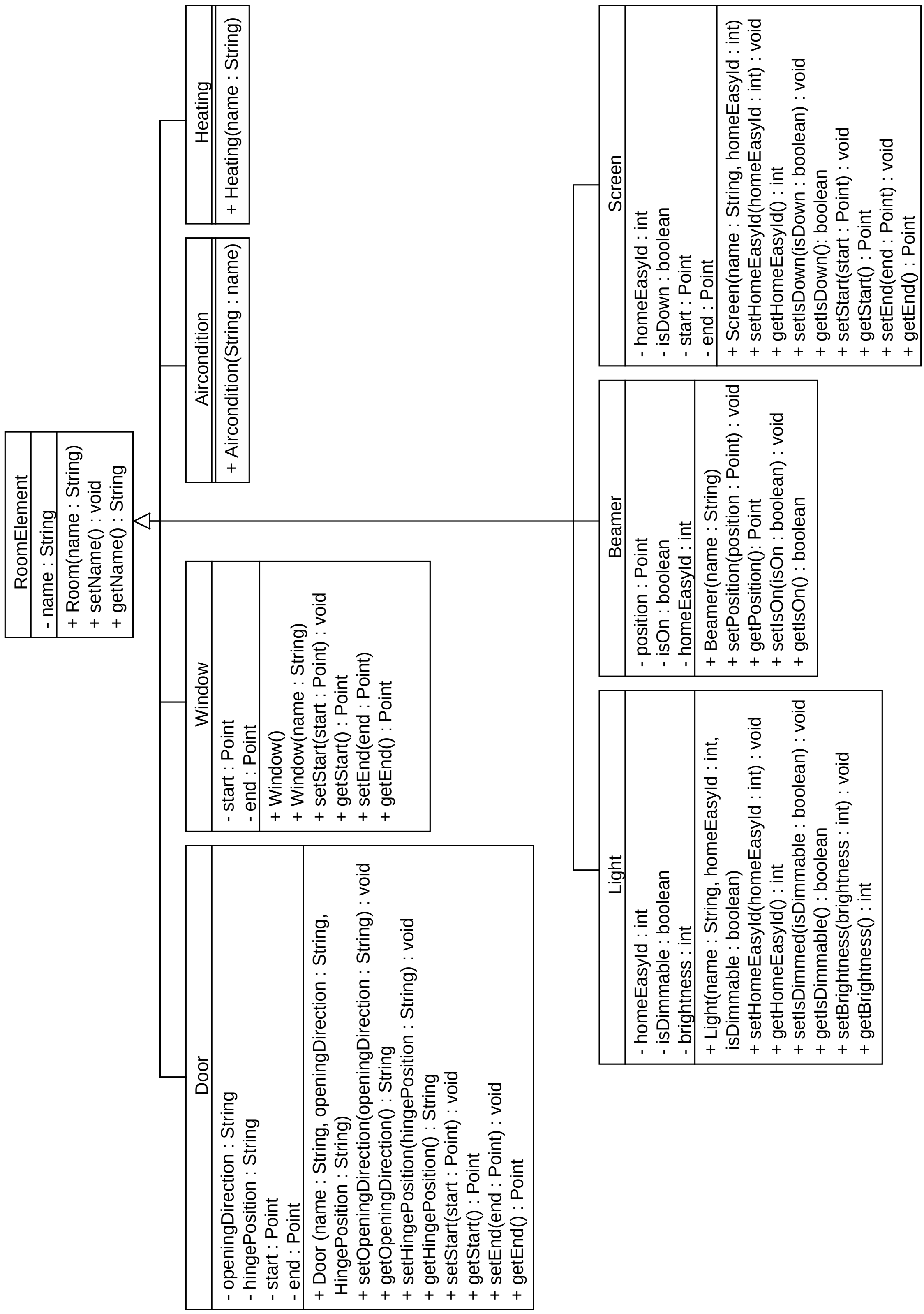

Figure 5.7: UML diagram of the "RoomElement" class and its children

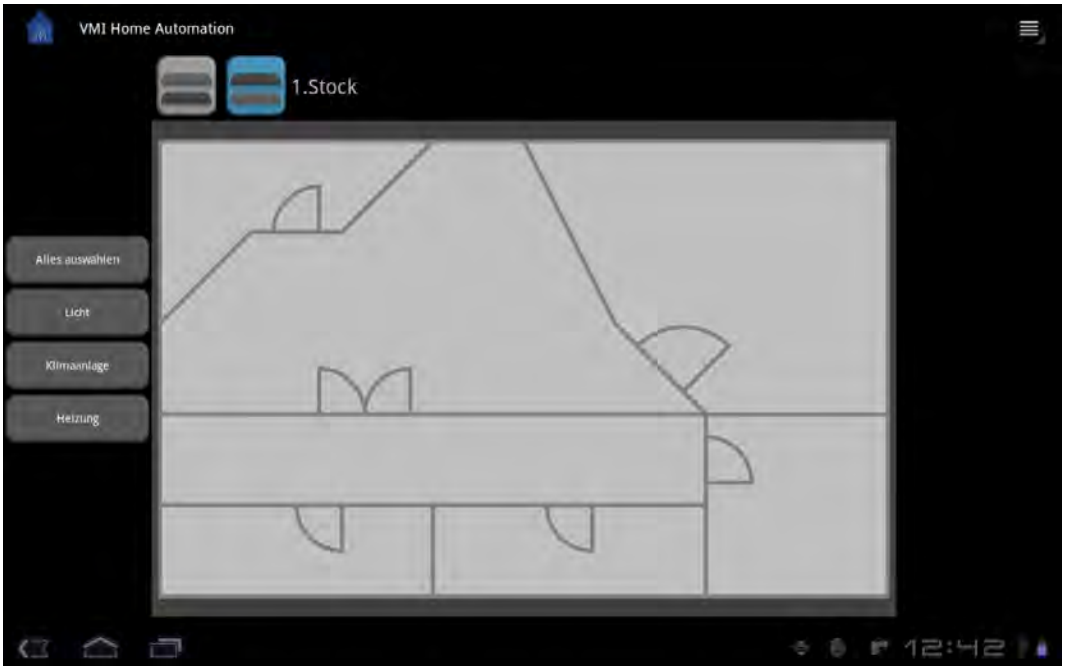

Figure 5.9: The space reserved for the “MapView” is colored gray

Chapter 6

Conclusion and Outlook

6.1 Conclusion

With the VMI Home Automation application, developed in this thesis, it is possible to visualize floor plans of buildings on Android devices and remote control a variety of Home Easy devices. As the colors and strings used for the user interface are defined in the corresponding XML files in the res/ folder of the application, no great effort is required to make modifications or translations into other languages. The same can be done for the buttons, as they are created from Portable Network Graphics (PNG), that can simply be replaced.
Unfortunately the logging function is not accurate enough for indoor applications, as there is an impreciseness of about ten to fifteen meters. The submitting of positions of several Wi-Fi access points at the university to Skyhook did not improve accuracy. Maybe this problem can be solved in the future by integrating an indoor positioning system, currently being developed at the Distributed Multimodal Information Processing Group.

The survey about the use of mobile devices revealed that mobile devices, especially smartphones, should be considered for future Home Automation systems as they are already widespread and used almost everywhere in a lot of different situations. Besides, most of the participants were interested in using Home Automation solutions in the future. As a lot of them have concerns about security and privacy, these issues should be taken seriously for the development of future systems and applications. Regarding the survey it can be criticized, that the number of participants was rather low and that most of them were rather young. As a large potential customer group for Home Automation system are elder people, because they are more likely to require assistance or medical surveillance, their requirements to this technology should also be examined and considered.

6.2 Future Work and Outlook

One thing that could be improved in the future is the way how floors are imported into the application. For example it could be looked for a way to import floors or houses created in Sweet Home 3D[1]. This would be more comfortable than creating XML files of floor plans manually.
In addition to that it should be easily possible for users to add new devices to existing environments. This could be realized similar to the idea presented in [2]. Barcodes could be printed onto the Home Easy elements and on a photocopy of the floor plan, indicating positions where Home Easy Elements can be placed. The application could provide an interface for scanning the barcode on the element that shall be placed and the barcode at the floor plan. By combining this information, a device can be assigned to a certain place. Apart from that the id defined in the XML could not only be used for Home Easy devices. It is imaginable that it could for example be used for IPv6 addresses of a new generation of Home Automation equipment.

Home Automation is a growing, dynamic market and a very popular research field where are lot of changes and new ideas are possible. The integration of mobile devices is a very likely development. One of the next steps could be the implementation of location-based services, like turning on the heating when the device, transmitting its owner's position to the Home Automation system, is less than fifteen minutes away from home.

[1] http://www.sweethome3d.com/de/index.jsp

Appendix A

Code Snippets

A.1 Scaling Funtions

Listing A.1: Scaling funtion for the horizontal direction

```
private float scaleX(int x) {
        float result;
                if (wRatio >= hRatio) {
                        result = ((((float) x) - min_x) / map_width)
                                * canvas_width + DISTANCE_TO_BORDER;
                }
                else {
                        result = ((((float) x)- min_x) / map_height)
                                * canvas_height + DISTANCE_TO_BORDER;
                }
        return result;
}
```

Listing A.2: Scaling function for the vertical direction

```
private float scaleY(int y) {
        float result, freeCanvasHeight;
                if (wRatio >= hRatio) {
                        result = ((((float) y) - min_y) / map_width)
                                * canvasWidth + DISTANCE_TO_BORDER;
                        freeCanvasHeight = canvasHeight
                                - (map_height/map_width)*canvasWidth;
                        result = result + (freeCanvasHeight/2);
                }
                else  {
                        result = ((((float) y)- min_y) / map_height)
                                * canvasHeight + DISTANCE_TO_BORDER;
                }
        return result;
}
```

A.2 Rotating Doors

Listing A.3: Calculating the rotation angle of doors

```
private float getDoorRotationAngle(Door door) {

        int start_x = door.getStart().x;
        int end_x = door.getEnd().x;
        int start_y = door.getStart().y;
        int end_y = door.getEnd().y;
        int delta_x = end_x - start_x;
        int delta_y = end_y - start_y;
        String hinge = door.getHingePosition();
        String direction = door.getOpeningDirection();

        if (hinge.equalsIgnoreCase("left")) {
                // case 1: left && in
                if (direction.equalsIgnoreCase("in")){
                        if (start_y == end_y) {
                                if (start_x > end_x)
                                        return 0f;
                                else
                                        return 180f;
                        }
                        else if (start_x == end_x) {
                                if (start_y > end_y)
                                        return 90f;
                                else
                                        return -90f;
                        }
                        else {
                                return (
                                        (float) Math.atan(delta_y/delta_x)
                                        * 180 / ((float) Math.PI) + 90.0f);
                        }
                }
                // case 2: left && out
                else {
                        if (start_y == end_y) {
                                if (start_x > end_x)
                                        return 0f;
                                else
                                        return 180f;
                        }
                        else if (start_x == end_x) {
                                if (start_y > end_y)
                                        return  90f;
                                else
                                        return 270f;
                        }
                        else {
```

```
                    return (
                        (float) Math.atan(delta_y/delta_x)
                        * 180 / ((float) Math.PI));
                }
            }
        }
        else {
            // case 3: right && in
            if (direction.equalsIgnoreCase("in")) {
                if (start_y == end_y) {
                    if (start_x < end_x)
                        return 0f;
                    else
                        return 180f;
                }
                else if (start_x == end_x) {
                    if (start_y < end_y)
                        return 90f;
                    else
                        return 270f;
                }
                else {
                    return (
                        (float) Math.atan(delta_y/delta_x)
                        * 180 / ((float) Math.PI));
                }
            }
            // case 4: right && out
            else {
                if (start_y == end_y) {
                    if (start_x > end_x)
                        return 180f;
                    else
                        return 0f;
                }
                else if (start_x == end_x) {
                    if (start_y < end_y)
                        return 90f;
                    else
                        return 270f;
                }
                else {
                    return (
                        (float) Math.atan(delta_y/delta_x)
                        * 180 / ((float) Math.PI));
                }
            }
        }
}
```

A.3 Drawing on the Map

A.3.1 Drawing Rooms

Listing A.4: Code for drawing rooms on the canvas

```
for (Room room : roomList) {
                currentCornersList =
                                new ArrayList<Point>(room.getCornersList());
                path = new Path();
                // flag for checking if the start point was set
                boolean movedToStartPoint = false;

                for (Point point : currentCornersList) {
                        if(!movedToStartPoint) {
                                path.moveTo(scaleX(point.x),
                                        scaleY(point.y));
                                movedToStartPoint = true;
                        }
                        path.lineTo(scaleX(point.x), scaleY(point.y));
                }

                // the path has to be closed
                path.lineTo(scaleX(currentCornersList.get(0).x),
                                scaleY(currentCornersList.get(0).y));
                path.close();
                movedToStartPoint = false;

                // the room is drawn
                if (room.getSelected())
                        canvas.drawPath(path, roomPaintSelected);
                else
                        canvas.drawPath(path, roomPaint);
                canvas.drawPath(path, wallPaint);
                }
```

A.3.2 Drawing Doors

Listing A.5: Code for drawing doors on the map

```
else if(e instanceof Door) {
        Door door = (Door) e;
        hinge = new String(door.getHingePosition());
        rotationAngle = getDoorRotationAngle(door);
        float size = getScaledDoorWidth(door);

        // Scaling the coordinates
        end_x = scaleX(door.getEnd().x);
        end_y = scaleY(door.getEnd().y);

        RectF bounds = new RectF(end_x - size  , end_y - size ,
                        end_x + size , end_y + size);

        if (hinge.equalsIgnoreCase("left")) {
                canvas.drawArc(bounds, rotationAngle, -90, true,
                        doorPaintStandard);
                canvas.drawArc(bounds, rotationAngle, -90, true,
                        doorBoundaryPaint);
        }
        else if(hinge.equalsIgnoreCase("right")) {
                canvas.drawArc(bounds, rotationAngle-180, 90, true,
                        doorPaintStandard);
                canvas.drawArc(bounds, rotationAngle-180, 90, true,
                        doorBoundaryPaint);
        }
```

List of Figures

List of Tables

Bibliography

[1] Ryan Aipperspach, Allison Woodruff, Ken Anderson, and Ben Hooker. Maps of Our Lives: Sensing People and Objects Together in the Home. Technical Report UCB/EECS-2005-22, EECS Department, University of California, Berkeley, November 2005.

[2] Chris Beckmann, Sunny Consolvo, and Anthony LaMarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In *Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp 2004)*, pages 107–124, September 2004.

[3] Björn Bittins, Jürgen Sieck, and Michael Herzog. Supervision and Regulation of Home Automation Systems with Smartphones. In *Proceedings of the 4th UKSim European Symposium on Computer Modeling and Simulation (EMS), 2010*, pages 444–448, November 2010.

[4] Khusvinder Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu. A ZigBee-Based Home Automation System. *IEEE Transactions on Consumer Electronics*, 55(2):422–430, May 2009.

[5] Carles Gomez and Josep Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6):92–101, June 2010.

[6] Li Jiang, Da-You Liu, and Bo Yang. Smart Home Research. In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, volume 2, pages 659–663, August 2004.

[7] Xin Ma and Wei Luo. The Analysis of 6LowPAN Technology. In *Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. PACIIA '08.*, volume 1, pages 963–966, December 2008.

[8] Renato Nunes. DomoBus - A New Approach to Home Automation. In *Proceedings of the 8th International Congress on Electrical Engineering (8CLEEE)*, pages 2.19–2.24, July 2003.

[9] A. J. Dinusha Rathnayaka, Vidyasagar M. Potdar, and Samitha J. Kuruppu. Evaluation of Wireless Home Automation Technologies. In *Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST), 2011*, pages 76–81, May–June 2011.

[10] Christian Reinisch, Wolfgang Kastner, Georg Neugschwandtner, and Wolfgang Granzer. Wireless Technologies in Home and Building Automation. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 1, pages 93–98, June 2007.

[11] Miguel A. Zamora-Izquierdo, Jose Santa, and Antonio F. Gomez-Skarmeta. An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. *IEEE Pervasive Computing*, 9:66–77, October 2010.