UNIVERSITY OF OSLO
Department of Informatics

Secdroid: An Improved Alarm Distribution System

Master Thesis

Håvard Bauge

March 19th 2013

Secdroid: An Improved Alarm Distribution System

Håvard Bauge

March 19th 2013

Abstract

Alarm systems are installed in buildings to protect assets and for people's security. When alarms get triggered, they require a fast response. Security companies have security response units that respond to alarm events. This thesis looks at how the alarm distribution system affects the alarm responses, and how helpful features and increased overview may improve the overall efficiency in a group of response units.

This thesis looks at the system currently in use, and a system previously used by a security company. Based on current short comings, we have designed and implemented a new system called Secdroid. The system consists of a server and several clients, where the server distributes alarm assignments to the clients. Secdroid has more functionality compared to the systems used by the security company. The most important functions are maps showing each unit's assignments and other units in the area, and functions for calculating duration and distance from other units to assignments, if a unit is in the need of assistance. The new system has been tested with good results. The users of the systems perform their tasks more efficiently, resulting in significantly shorter response times. The feedback from the users is also very good compared to the previously systems used by the security company.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background & Motivation

Alarm systems are installed in most office buildings and industrial properties. They are also commonly installed in private houses. They are installed to help protect assets and to help keep people safe. In the case of an unwanted event, the right personnel should be notified in order to cope with the event and assist people who may be affected. Alarm systems passively monitor buildings or other areas and get triggered by adverse events. Alarm systems may have a various selection of sensors, each being triggered by a predefined event.

An example of sensors that may be installed in a private house is shown in Figure 1.1. The example contains door/window detectors and motion detectors to detect burglaries, water and freeze detectors to detect water leakages, smoke and heat detectors to detect fire, and an alarm which can be pressed if the user for some reason feels unsafe. In addition to detectors passively monitoring buildings, it is common for places like banks, convenience stores and pharmacies to have emergency buttons which can be pressed in case of a robbery or urgent need of assistance. Elevators have emergency buttons which can be pressed if someone gets stuck in an elevator.

A fast response for such events may be crucial to save assets and in some cases also people's lives and health. Burglars may get caught, small developing fires may be extinguished and water leaks may be stopped. In any case, someone gets to the alarm location and can contact police, fire departments, plumbers or others to secure the building and limit the damage as much as possible. In nearly all of the above described scenarios, time is critical. The system for receiving and processing alarm events needs to be as efficient as possible. People in distress, house owners and office managers should get help as soon as possible when something happens to them or their buildings. When only minutes can separate a good and bad outcome, it is critical to minimize the delay in the entire sequence of events, from the alarm being triggered until someone is on the way to assist.

Alarm systems are normally connected to an alarm company. When an alarm system gets triggered, it sends out a message to the alarm

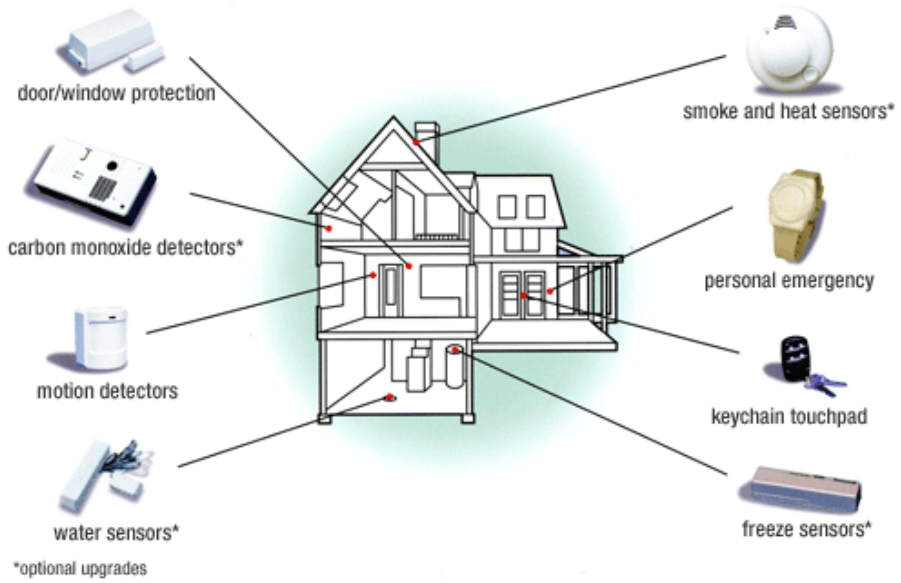

Figure 1.1: Example of sensors in a private house [24]

company. The message is typically sent out over the Internet or over wireless cellphone network, or both. When receiving an alarm notification, the alarm company follows the protocol for the location from where the alarm message is sent and the type of alarm (fire, burglary, etc.). If the initial check does not clarify the reason the alarm was triggered, a mobile response unit gets sent out to the location. The security guard receives an assignment that contains information about the customer, the location of the alarm and the alarm type. After receiving the assignment, the security guard drives to the location as soon as possible.

When several alarm assignments reach one response unit at around the same time, the order to which the unit should respond is up to the security guard driving the unit's discretion. It is based on, among other factors, the type of alarm event and the event's geographical location. Information included with the assignment may have an effect on the priority. The information may include confirmed security breaches or other information suggesting that a security breach is likely to have occurred. For example, a burglary alarm where several sensors have been triggered. Events around the same area should, if possible, be responded to sequentially, to limit the time consumed by driving.

In this thesis, we have looked into the alarm assignment distribution in one of the major security companies in Norway. A system is used to distribute alarm assignments. When an alarm has been triggered and a response unit is needed at the location, the system at the security company's dispatch looks up the designated unit exclusively based on the assignment's address. Each response unit holds a client device where it receives assignment. The software on the device displays the information about the assignment, messages and an event log where information about the alarm triggering times and the unit's progress in the response.

In areas with more than one unit available, a unit may ask for assistance from other units if it experiences a high workload. The system currently in use by the security company does not provide the user with information about the other units in the area. Information about other units' positions, availability and distances to alarm locations would be helpful when a unit is in the need of assistance. Now, the users of the system have to manually contact other response units and ask about their locations and availability. The usage of units from other departments which have mobile units, is very limited, simply because of their unknown locations.

1.2 Problem Statement

The system used today shows no information about other units in the area, not even other response units. Using the current system, guards have to manually contact either the security company dispatch or the other units to ask about their workload and positions. The device used in the system cannot be used to forward assignments. Instead, the users have to contact the alarm dispatch and have them send the particular assignment to the designated unit. In addition to the limitations in the current system's functionality, the system is also slow and unreliable and it has a history of frequently crashing. System crashes during periods with high workloads slows down the user using the system while the user waits for the problem to be resolved. The current system is installed on a *PDA (Personal Digital Assistant)*.

This thesis looks into the problems and limitations of the existing system and discusses a possible solution in a new system, named *Secdroid*, which we have developed. The underlying goal of the new system is to improve efficiency in the terms of decreasing the alarm assignment response times. The response times can be reduced *1*) by including units in the area which are performing other tasks than alarm responses, *2*) by giving the alarm response units a better overview over assignments and other units in the area, *3*) by making it easier to forward assignments between units and *4*) by developing a more reliable system. A goal is to also improve the user interface by making an interface which is more intuitive and easier to use and which is also easier to learn how to use.

By implementing the suggested improvements in a new system, the resource of units outside of the alarm response department can be used. Today, the usage of other units is limited because the alarm response units are unaware of the other units' locations. A better overview will likely lower the threshold of asking for assistance during high workload, since the workload and positions of the other units are known. An easier assignment forwarding functionality will limit the hassle of forwarding assignments and encourage guards to involve other units to a higher degree than today to achieve more efficient responses to alarms. A more user-friendly system will limit the training costs and limit the chance of user generated errors.

Secdroid will target Android operated smartphones and run in the

background of the operating system on the smartphone. The system will be used without giving the users any devices in addition to a cellphone, which they already need to perform their work tasks. Secdroid will include a map where the user's alarm assignments, and the other units using the client software in the area will be marked on the map. The map will give information about the other units' workload, which will give the users of Secdroid a good overview. In addition to this, the driving time from other units to particular assignments will be available. Secdroid will include a function for easily forwarding assignments to other units. Reliability is an important factor when developing Secdroid, and to develop a reliable system is one of the goals.

1.3 Scope & Limitations

This thesis covers alarm responses within a single security company. The client used by the security guards responding to alarm assignments is developed for smartphones running the Android operating system. The system has been tested in Oslo, Norway during five shifts with a total duration of 43 hours. During those shifts, 151 alarm assignments were distributed. 14 unique users tested the system. Units performing other tasks than alarm responses, as mentioned in the problem statement, are not included in the performed tests due to lack of resources.

1.4 Contributions

Secdroid is a new system which has been developed to improve alarm responses. Secdroid uses client software developed for smartphones running the Android operating system, described in Section 3.3. Server software has also been developed to control and to keep track of the clients, described in Section 3.2. An interface, *Secdroid Assignment Management Interface*, for monitoring assignments has also been developed, see Section 3.5. The assignment management interface has been developed for testing the system.

The clients enable the users to receive and manage alarm assignments. The users can forward assignments via the server by using the client software. It also includes a map where the users can see where its assignments and other units in the area are located.

The server software distributes assignments to the clients. The server software is also responsible for checking that the clients have received the assignments they are supposed to have. The clients regularly report to the server, and include a list of assignments stored in the client's database. The server software utilizes services provided by Google to retrieve coordinates of addresses, to calculate distances from units to addresses and to send assignments and messages to the clients by using *push technology* developed by Google for Android.

The Secdroid Assignment Management Interface lets the administrative user create assignments, distribute assignments to clients and cancel

assignments. The interface is also showing the status of active assignments and users and a history of completed assignments. The interface has a map where active assignments and the units' positions are pinpointed.

The system has been tested and the average response time has been decreased with almost 19%, as described in Section 4.1.4, which fulfills the goal of decreasing the response times as mentioned in the problem statement. The goal of improving the user interface has also been fulfilled, according to the user survey described in Section 4.2.

1.5 Outline

This thesis consists of five chapters. An explanation of each chapter is listed below:

Chapter 1: Introduction The first chapter provides an introduction to the thesis. It explains the background and importance of creating the system described in the thesis.

Chapter 2: Background & related work Looks more in depth at the background and the scenario to which this thesis applies. It also explains how previous systems works and mentions some similar systems.

Chapter 3: Design & implementation Details about the design and implementation of the system.

Chapter 4: Experiments & discussion Information about the testing, statistics, test results and user feedbacks.

Chapter 5: Conclusion A summary, contributions and further work.

Chapter 2

Background & Related Work

2.1 Scenario

The company manages a fleet of multiple mobile units responding to alarm events. The security company responds to about 2000 alarm events every month in the city of Oslo, where the experiments were conducted. That amounts to about 65 events every day. The city is divided into four geographical areas, with one response unit responsible for each area. The most common way to commute is by cars and that is also the basis of this thesis.

There are several different alarm events handled by the security company described in this thesis. The most common type is a burglary alarms at private houses or companies. Other types include fire, robbery, assistance, elevator or technical events. Those have different priorities, and it is up to the security company and the unit receiving the assignments to make good prioritizations. A general priority scheme is shown in Table 2.1.

After an alarm has gone off, a response unit is sent to the location. The assignments are obtained from a alarm company who detect alarm events from their customers, mostly corporations and private houses. Both the alarm company and its subscribed customers expect quick response times from the response unit, regardless of the event type or the severeness of the event. However, the response units are trained to use their judgement and the security company's protocols in order to achieve the best responses as possible.

Priority	Example alarm types
Very high	Robbery, customer assistance
High	Elevator, fire
Normal	Burglary
Lower	Technical
Low	Service

Table 2.1: Alarm event priority guide

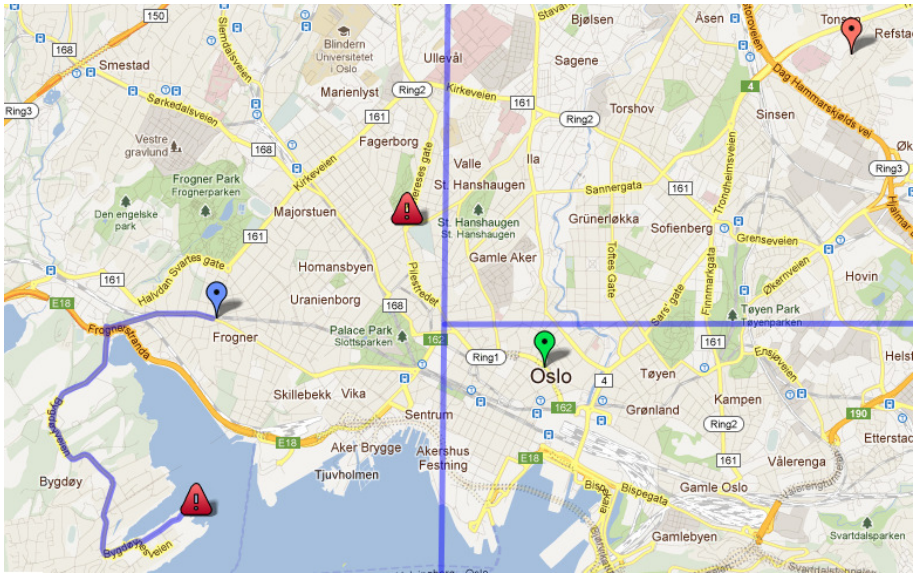

Figure 2.1: Example assignment and unit distribution

2.1.1 Response to assignments

When a response unit receives an alarm assignment, it has to get to the address where the event occurred in the shortest possible time. The challenge arises when a unit receives one or more new assignments while it is not yet finished with its current assignment or assignments. Keep in mind, assignments may have different priorities. The geographical locations of assignments may also make it more efficient to rearrange the order of the responses, to prevent using driving time by driving between different areas.

Figure 2.1 shows a map over the central Oslo area. There are three active units. The current position of the units are marked with pinpoints, each having responsibility over the entire areas divided by the lines. The unit located at the farthest left on the map has two assignments in queue, marked with triangles with an exclamation mark, while the others have none. The unit at the farthest left on the map received a second assignment while on route to the first assignment it received.

The unit with assignments in queue will now have to manually contact other units to get help with this newly received assignment, which is in the opposite direction of the first alarm, but still within his or her designated territory of Oslo. The system currently in use by the security company gives no indication of who or where the closest unit is, except for the predefined areas known to the units which break up the city into three areas. According to those areas, the unit in the top right corner of the map should be closest to the address where the second assignment is located. However, the map in Figure 2.1 clearly shows that the unit close to the center of the map is available and significantly closer. To get assistance from this unit would also maintain a better distribution among the units in case additional assignments are received.

In an example scenario, several units may be covering the same area causing the workload at that point in time to be unequally distributed. One unit may have several alarm assignments in queue while other units do not have any. It is implied that other available units assist the unit with the longest queue of assignments. However, because the assignments are distributed to units based on the address of the client only, each unit is unaware of the workloads of the others. This setback forces the unit which needs assistance to contact the others to ask for help. At that point, if there are two or more available units, the initial response unit has to choose which assignments to send to which units, all this while performing his job duties efficiently and correctly.

2.1.2 Assignment lifetime

Dispatches are control rooms where operators communicate with customers and security guards. The operators and the system at the dispatches distribute assignments and receive reports from the security guards responding to the assignments. There are differences between an alarm company dispatch and the security company's dispatch:

Alarm company dispatch Each alarm company dispatch receives alarms from their customers. Their customers are private houses, office buildings, etc. The communication to the end users goes via the alarm company dispatch. The alarm company dispatch uses services provided by the security company.

Security company dispatch The security company dispatch has alarm companies as their customers. The security company dispatch receives assignments from one or more alarm companies and can send security guards to the locations provided by the alarm company.

Figure 2.2 illustrates the process from when an alarm has been triggered until the location has been checked by a security guard. The process contains of the following main steps:

1. Alarm is triggered, location ID is sent to an alarm central dispatch.
2. The location ID is received at the alarm central dispatch, the dispatch looks up the customer information and decides whether or not to send someone to the location.
3. The assignment is received by the security company's dispatch and they assign a unit to the alarm based on the location's address.
4. A unit receives the alarm assignment and responds.
5. When the unit is done at the location, it reports back to the security company's dispatch.
6. The security company's dispatch processes the report to make sure everything looks right before they send the report back to the alarm central.

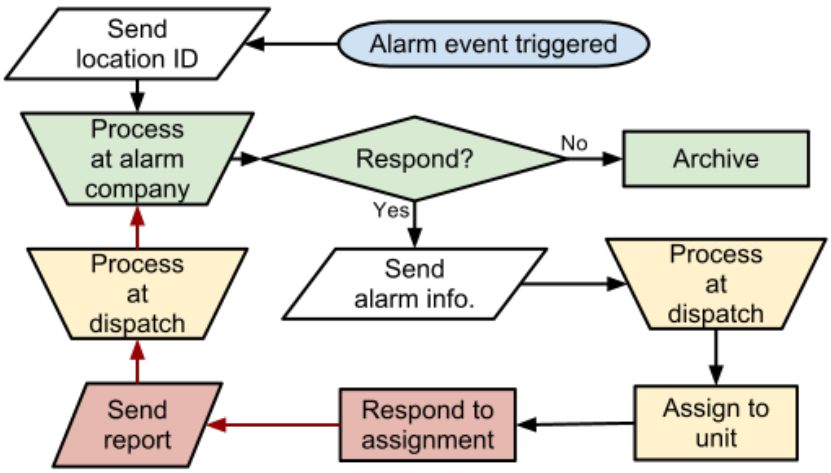

Figure 2.2: The assignment lifetime

The first in Figure 2.2 begins at the location of an alarm system. Alarm systems have one or more communication connections to the alarm company. The connection usually includes phone lines, mobile phone lines and/or an Internet connection. When an alarm triggers, the alarm system notifies the alarm company. The information may vary, but at the very minimum, something is sent to the alarm company to identify the location. Identification information such as the caller ID or a customer number is sent out and usually includes information about the alarm type as well. In some cases, information is also sent out that tells which detectors have been triggered and the alarm activation and deactivation specifications.

When the alarm company receives this information from the alarm system, it looks up the customer in their customer database. The database holds information regarding the customers' preferences. Some customers may want to be contacted to decide for themselves whether or not they want a response, whereas some may only want responses outside of normal office hours and so forth. If the criteria for said customer includes responding to the alarm in person, the information about the customer and alarm information is sent to the security company dispatch.

The security company dispatch then creates an assignment based on the information received from the alarm company dispatch and assigns a unit to be responsible for the assignment. The unit selected is usually based on the address of the assignment. The dispatch then notifies the assigned unit.

The unit responds to the assignment according to the instructions given by the alarm type. When the unit is finished at the location, it reports back to the security company dispatch. After evaluating the report, the security company dispatch sends the report to the alarm company. The alarm company saves the information for future references and may also contact their customer with details about the cause of the alarm and measures taken.

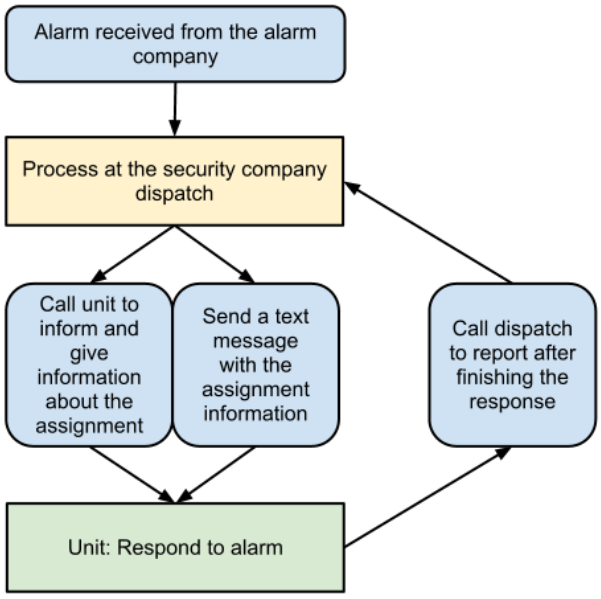

Figure 2.3: Assignment lifetime using a manual system

2.2 Previous System Used by the Security Company

The previous system used by the security company was replaced by the system currently in use, described in Section 2.3. The previous system is based on distributing alarm assignments by using *text messages*. In the previous system, calling and sending text messages was used to communicate with the clients. The use of phones was an upgrade from the radio communication system, which was in use prior to that system.

Figure 2.3 shows the lifetime of an assignment from it is received from the alarm company. The unit-lookup was an automated process, but the unit had to be manually notified. An operator at the dispatch had to call the unit to inform him or her about the new assignment. The unit confirmed over the phone, and the assignment was sent as a text message. The text message included information about the assignment type, the customer's name and dress and other relevant information about the event. Real-time alarm information, such as new zones being triggered or deactivation of the alarm system was not forwarded to the unit.

When arriving at the assignment location, the unit had to notify the dispatch about its arrival. That was done either by calling or by sending a special code as a text message.

After finishing at the assignment address, the unit had to call the dispatch to report, before it could leave the location. In Figure 2.3, the arrows going back from the unit illustrates the reporting process.

2.2.1 Advantages & disadvantages

The previous system relied to a large extent on manual operations. Reports from the security guards had to be manually written down by the operator at the dispatch after receiving them over a phone line. The risk of errors was large, as messages could be misinterpreted over the phone.

When using a system based on phone calls, more time was used by delivering and receiving messages and to manually input data. That increased the workload, especially at the security company dispatch, but also for the user. The user might have other phone calls to make or other things preventing him or her to make a phone call to report back. It was also frequently high waiting times before the dispatch answered the phone call which led to longer response times for assignments in queue. The system also lacked real-time information about status changes at the alarm system and information about other units.

The greatest advantage with the previous system was its reliability. By using already implemented, well tested and maintained services (mobile phone network), the system had a great uptime. In the case of a network failure with the mobile network operator, backup phones from another mobile network operator were available. That made the system very robust, with a downtime close to zero percent. The previous system is still frequently in use as a backup system for when the system which is currently in use by the security company fails.

The system was relatively easy to learn and to use. The survey conducted in connection to this thesis work reveals that this system is preferred over the system currently in use among many of the users, see Section 4.2.

2.3 System Currently in Use by the Security Company

The system currently in use is mostly an automation of the previous system, where the dispatch operator and the unit had to communicate using phone calls. The current system allows alarm assignment distribution and reporting to be done using a hand-held *PDA (Personal Digital Assistant)*. The system is developed by *Evry* [32]. This automated system decreases the workload of both the unit and the dispatch operators.

This system works well in a scenario where there is only one unit handling all the assignments. In such a scenario, the dispatch sends out the assignment to the only available unit in that area. The unit responses upon retrieval. If the unit is busy, the assignments are put in queue until the unit is available. With several assignments in queue, a prioritization system, developed by the security company, is used to help determine which assignment to do next. The PDA does not give information about other units in the area. That information has to be obtained by calling the dispatch or every other unit and asking.

The process of receiving assignments and reporting using the PDA is shown in Figure 2.4. Each unit holds one PDA device. The PDA

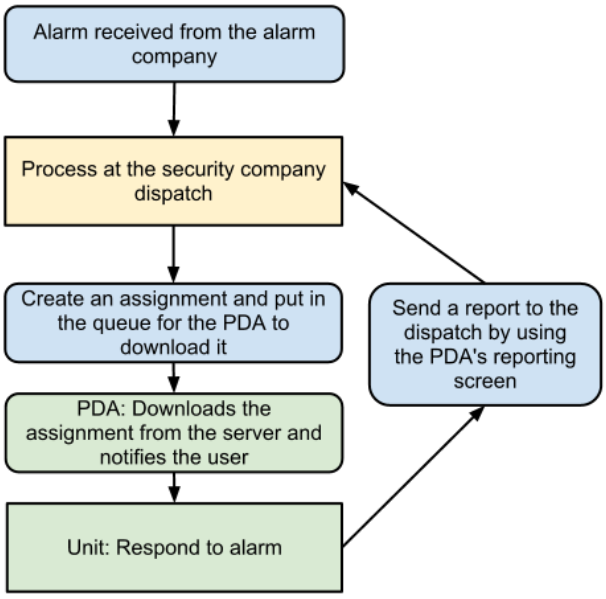

Figure 2.4: Flowchart using an automated system

automatically receives new assignments from a centralized server as they come up in queue.

1. Assignments are sent to the dispatch from each alarm company.
2. A system at the security company's dispatch attach a unit to the received assignment using a predefined localization system based on the assignment location's address.
3. Then, the assignment is downloaded by the unit's PDA over a mobile Internet connection.
4. When the assignment has been downloaded, it creates a notification to the user.
5. The user then has to click on the message in its message box. The mailbox icon is visible in the top right corner of Figure 2.4.
6. When the user selects the assignment message in its message box, the basic information about the assignment is displayed to the user, as shown in Figure 2.7. Now, the user has to chose to accept or decline the assignment.
7. If the user accepts the assignment, the assignment is put in the PDA's assignment list.
8. The user uses the PDA reporting screen to create and send a report back to the dispatch. The reporting screen is shown in Figure 2.7.

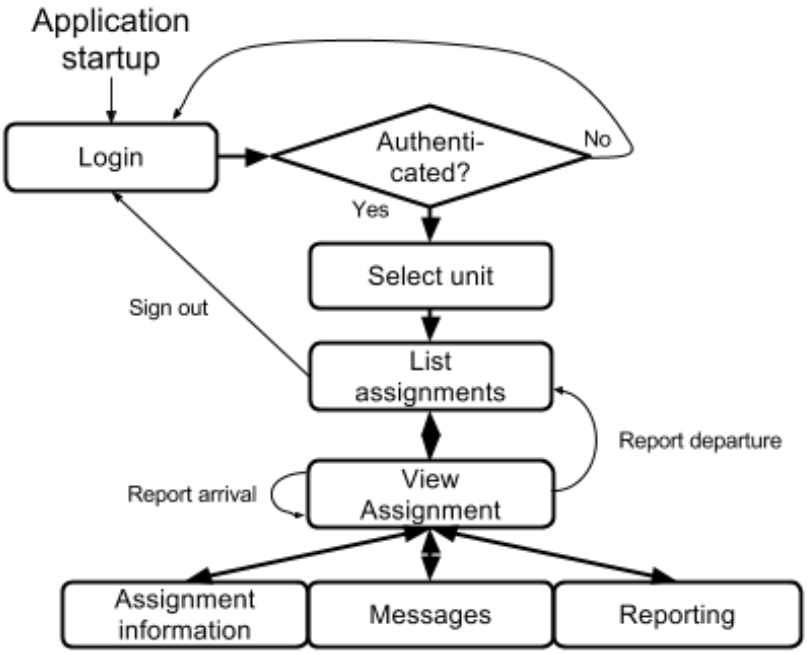

Figure 2.5: The PDA application life cycle

2.3.1 Application life cycle

Figure 2.5 shows the PDA application life cycle. Every screen used in the application is explained in the list below.

Login When the application starts up, the user is asked to provide its employee number and password to sign in.

Select unit Once the user has been authenticated, it is asked to select which unit to sign on to.

List of assignments This is the main screen of the application. It shows a list containing any active assignment.

The menu on this screen includes an option for signing out.

View assignment This screen shows the basic information about the customer. It also contains a log showing the alarm activity.

The menu on this screen includes options for reporting arrival and departure, cancel the response and an option for launching a screen containing the full assignment information.

Messages A screen showing additional messages sent from the security company dispatch.

Assignment information This screen contains the full information about the assignment, including the ID number, customer number, name, address, the alarm company and the alarm triggering time.

Reporting The reporting screen. This is where the user selects the probable alarm triggering cause, measures taken at the location and detectors which were triggered.

2.3.2 Assignment distribution

The assignments are distributed using a technology called *Alystra* [31], which is developed by Evry. The Alystra technology uses an open TCP connection to push assignments to the PDA. When the PDA is connected to a mobile network, the assignments are usually received within some seconds. The security company dispatch does not receive information about whether or not the assignment has been received by the PDA.

The solution is that the system at the dispatch monitors the assignments. If the user does not accept the assignment within 1.5 minutes, an operator at the dispatch gets notified, and should contact the unit and inform it about the assignment. With that solution, a delay of minimum 1.5 minutes occurs if the PDA is off-line.

According to user reports, the PDA is frequently disconnecting, even in areas where it is known mobile Internet coverage. The disconnections may be caused when the open TCP connection used for pushing assignments to the PDA device may be closed by the network, and that the mechanism for detecting disconnections is not good enough.

2.3.3 Activity log

The activity is visible on the bottom of the assignment view screen. The log shows alarm events as well as every attached units' events, such as arrival and departure. The log does not, however, show historical log events. Events which occurred before a unit received the assignment are not included in the log list. This is considered to be an error in the software, as it should have been there.

2.3.4 Reporting

Arrival and departure reporting are options which can be selected from the view assignment screen. They are two separate buttons, where the valid action (reporting or departure) is click able depending on the response state. The PDA system has simplified the reporting system for sending reports after the user is finished at the assignment location. Now, departure reporting back to the dispatch can be done by filling out a reporting form on the PDA and sending it to the dispatch, as shown in Figure 2.6.

The reporting process is, however, a little bit tedious because of a slow response from the PDA and a non-intuitive interface. For example, only one measure can be added at a time. To add several measures, the steps of selecting the measures button, confirming that the user wants to add another measure and then select the new measure has to be repeated for every new measure. When the user has filled out the reporting screen the

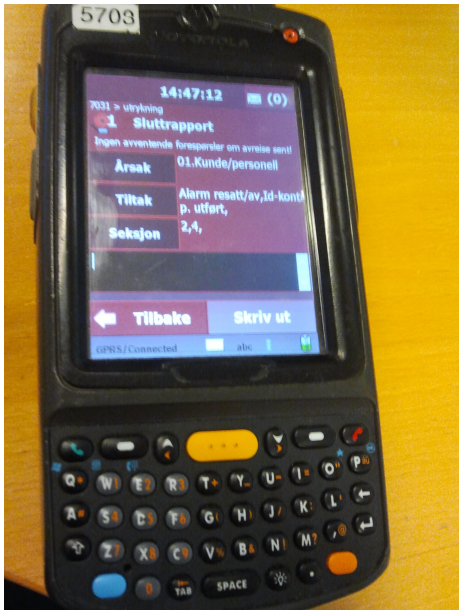

Figure 2.6: The PDA reporting screen

"back" button has to be selected and "departure" has to be chosen from the assignment view screen's menu.

2.3.5 Drawbacks

Positioning and distance The units are unaware of other units' positions and distances to addresses where they need assistance.

Communication between units Units are unable to directly communicate with each other using the PDA system. They rely on cell phones for that, which is another item to be accountable for.

Prioritization Each assignment is equally prioritized. No difference is made in the system according to the severeness of the assignment.

Other units' workloads Each unit is unable to know the workload of other units.

Other applications The PDA application always runs in the foreground, which disables the user from using any other applications installed on the PDA.

There may be several mobile units covering different sub-areas within a city. Each unit needs to know the others' positions for it to ensure it makes the best prioritization decisions. There are common scenarios where the best assistant choice is not the most logical choice as illustrated in Figure 2.1. The PDA system does not give information about other units' positions, which would have been helpful when it is critical to use as little time as possible to make the correct important decisions. The alternative is

Figure 2.7: PDA hangup

to manually call other units to ask about their positions. This wastes a lot of time.

It can be useful to have knowledge of other users' workloads. At times, there might be nothing in queue for some and very much in queue for others. Although a unit may potentially handle their heavy workload, it would be beneficial to have the option to divide the workload, especially if emergencies arise.

Every assignment received by the PDA is formatted the same. The user has to read the text in order to figure out what type of assignment it is, and how critical the assignment type is.

System crashes

The PDA is frequently crashing, according to user reports. It is known for random crashes with no obvious reason. To recover from a crash, the device has to be rebooted, which takes several minutes. Reports suggests that crashes are more frequent with high usage and when it holds two or more assignments at the same time.

There are some known scenarios where the PDA crashes. One example is shown in Figure 2.7. The user has just chosen to accept an assignment, but the accept is not approved by the system at the security company dispatch. This happens when the assignment is canceled after the user chose to accept the assignment and before the system at the dispatch registers the accept. If this occurs, the user has to go through a long recovery process which includes rebooting and sign-out/sign-in.

2.4 Related Work

We have not been able to find any published research in the area of alarm assignment distribution during the work on this thesis. There are however research in the Fleet Management field, with similar concepts. Fleet Management systems are commonly used in modern cities. It is used by emergency vehicles, taxis, transportation companies and buses, trams and other public transportation, just to name a few.

Real-time bus coordination It has become very common for bus companies to operate with real-time information about bus departures. Buses are equipped with GPS devices and are connected to the Internet. This lets them submit their location to a centralized control center for real-time analysis. Some issues that may occur are: delayed buses, full buses, accidents or bus breakdowns. Methods for predicting schedules and passenger information have been created [27] [42].

Taxicab control system Modern day taxi companies use GPS and mobile networks, normally, to submit the taxicabs' positions to a centralized control center. That makes it easy to dispatch the closest available unit to customers ordering transportation.

Oslo Taxi Traffic Control System (OTT) is a system developed by the Oslo-based taxicab company Oslo Taxi. It assigns units into a zone based on their position, and a queue number for that particular zone [43]. The report referred to is from 1998, but the system is till in use by Oslo Taxi and other taxi companies [23].

There are also several existing fleet management applications designed for Google's Android operating system. Most of these applications are available for download at Google's application store, Google play [34]. Some of the applications are listed below:

Device fleet management This application lets you track one the movement of multiple android devices. The application has a web interface for tracking historical movements [55].

Scania fleet management An application for keeping track of vehicles' current positions. The application also can also provide information about current fuel level and speed [35].

Google fleet management service Google is developing a service for businesses to keep track of units and assignments. It allows one view units and assignments on a map and assign custom made jobs to the workers. This is the service currently known that is most similar to the service described in this paper [45] [38]. A screenshot of the map is shown in Figure 2.8.

The solutions mentioned in this section are not designed for alarm management. The system described in this thesis also differs from existing

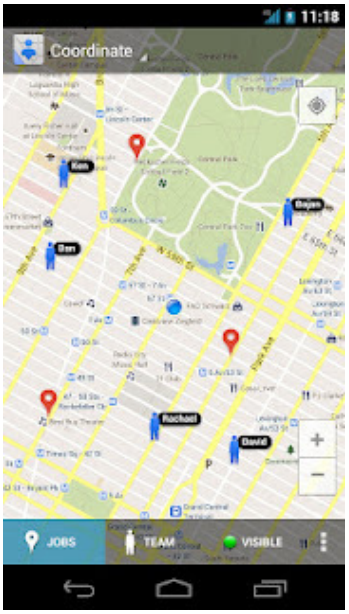

Figure 2.8: Screenshot of Google's fleet management service [45]

solutions with the ability to show other units and their distance from units to a location.

2.5 Summary

Alarm responses are time-critical. The customer expects short responses and security companies need to be efficient to perform best possible responses by using their available resources. The challenge of being efficient increases with more units available in the same area. A good system, which gives a good overview over the unit's own assignments and other units' assignments, can be very helpful in improving alarm responses.

In this chapter, two systems used by the security company are described. One system which is currently in use and one which was used prior to the system currently in use. These systems do not give the users an overview over other units' positions or workloads. They do not utilizes other units' positions to calculate driving time to assignments, which is very useful if a user is in the need of assistance. These points are some of the goals for a new system, as described in Section 1.2.

The system previously used relied on manual operations. Alarm assignments were sent out to the units as text messages and an operator had to call the unit to confirm that the assignment was received. When a unit was finished at an assignment address, the user had to call the security company dispatch to report. When the system was relying on manual operations, there was a high degree of human-generated errors. Messages were sent over the phone and could easily be misinterpreted. The system was, however, a very reliable system, since it using well-established

cellphone networks.

The previous text messages system was replaced by a new system. The newer system, which is currently in use by the security company, automates the assignment distribution and reporting. An automated system limits manual operations and decreases the assignment distribution and reporting time. This system's software is installed on PDAs, which forces the users to carry on one extra device in addition to the cellphone needed, since the PDAs do not support making phone calls. In addition to its functionality limitations, the client software is also very unreliable which slows down the users when performing their tasks.

The next chapter describes a new system is designed to eliminate the drawbacks described in this chapter. The new system has added features for calculating distance to assignments, show units and assignments on a map and for forwarding assignments, which will help the users get a better overview over their current assignments and other users in the area. A better overview will likely improve the efficiency and decrease response times.

Chapter 3

Design & Implementation of Secdroid

3.1 Introduction

This chapter explains a solution for resolving problems and limitations in current alarm distributions systems. To improve the efficiency of units in an area, which is resulting in shorter response times, the units need a good overview. When a unit experience a high workload, it should receive assistance from other units. If a unit in the need of assistance knows the position and workload of the other units in the area, it will simplify the process of selecting the best possible unit to receive assistance from.

There exists systems for distributing alarm assignments. However, these systems isolate each unit in an area without giving it an overview over other units. The units are mobile and move around in a large area. Systems currently in use or previously used by the security company does not utilize the positions of the units on duty. The user will have to manually contact the other units to receive information about their availability and positions. The unit should be able to automatically receive this information, but that is not not possible with the current solutions.

To address this, I propose Secdroid (**Sec**urity Fleet Management for An**droid**) as a replacement for the current PDA solution. This new system offers better functionality and is designed to be more user friendly and more operationally reliable. The system gives the user a good overview over every unit in the area, which is needed for efficient alarm assignment responses. Secdroid has functionality for showing every unit in the area on a map where also their availability is indicated. With Secdroid, user can also calculate distance from other units to an assignment, to determine which unit is closest.

The client application is developed for Android smartphones and it is custom made for the purpose of being used by a company and its selected employees/units. Therefore, the application is not expected to be released in the Android's application store, Google play [34]. However, the application is easily installed from any computer using a USB cord. The installation process takes only one click and is finished in a couple

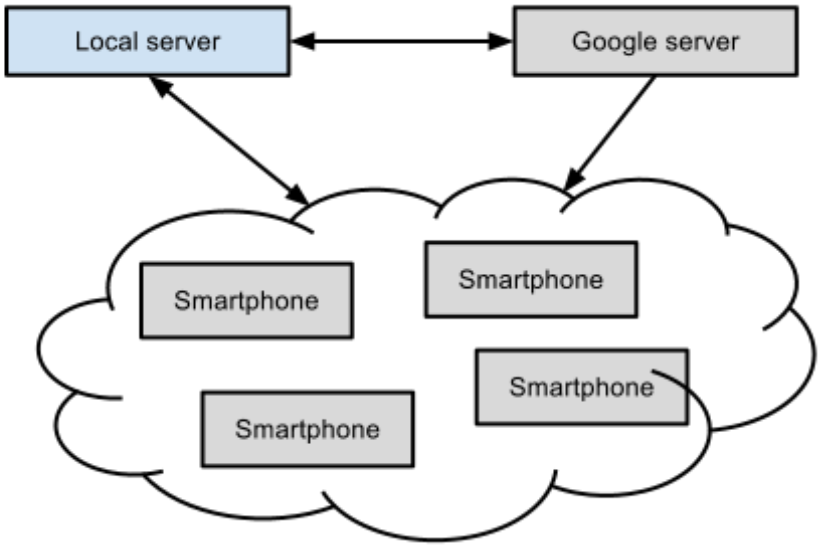

Figure 3.1: System overview model

of seconds. The application can be installed on any device within the company's discretion. The client solution is explained in Section 3.3.

The application runs in the background on the smartphone, and the phone can be used normally while the application is running. This enables the company to install the application on devices held by units outside of the alarm response departments with only making small investments. Other units still need a mobile phone, and the procurement of smartphones would have been a possible investment, independent of this software.

Security companies usually have mobile units doing other work than responding to alarms. If those units are also running the program in the background, those units can be utilized by the response team to decrease response times to a far greater extend than today.

The security company dispatch has a system for communicating with the PDA. This system is outside the scope of the thesis. However, a similar system is needed to test the Secdroid system. A simple interface for distributing and managing assignments, called *Secdroid Assignment Management Interface* has been developed.

3.1.1 System overview

The system consists of a centralized server communicating with all the clients, which also takes use of some of Google's API services. A model of the system is shown in Figure 3.1.

The main communication lines shown in Figure 3.1 involves:

- The server handles the client authentication and logout processes.
- The server receives assignments from the alarm companies, and distributes them to the clients. The clients report assignment status

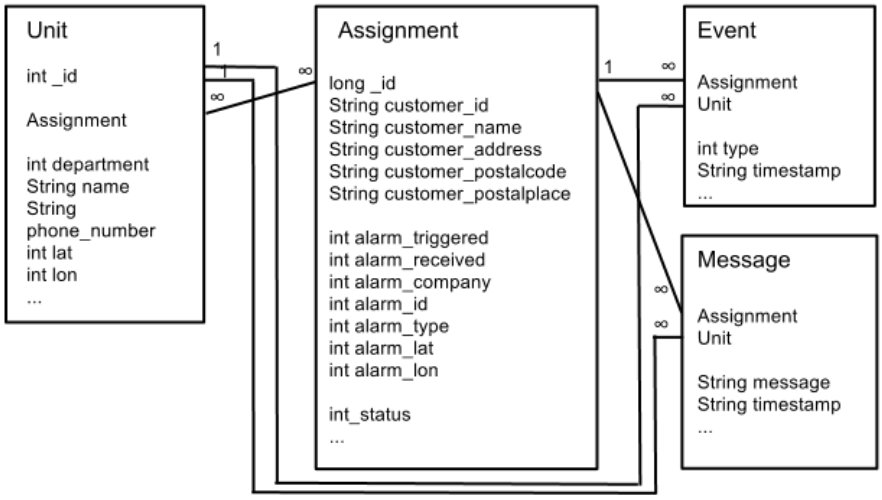

Figure 3.2: Class and database diagram

updates to the server and the final report when an assignment respond has been finished.

- The clients periodically report their position to the server to allow the server to calculate distances from the clients to addresses.

- The Google servers are used to get address coordinates which are used to put assignments on the map, and to calculate distances.

- Assignments distributed from the server are sent via the Google server, using a push-to-Android device service provided by Google.

The communication between the clients and the local server is over Mobile Internet. The *Hypertext Transfer Protocol (HTTP)* is used to transfer information between the local server and the Google server, as well as between the clients and the servers.

3.1.2 Databases & classes

Both the server and the client store information in databases. Figure 3.2 shows the most important tables and relations at the server. An assignment can have many messages and many events. A user can have many assignments and an assignment can have many users. Table 3.1 lists the database tables in use at the server, with a short description of each table.

The clients need to store less information and have a simpler database, with tables for storing own *assignments*, other *units* in the same area and tables for *messages* and *events* belonging to the unit's own assignments.

Both the server and the client software are programmed *object oriented* and use the same class structure. There are classes for assignments, units, messages and events.

Table name	Description
Alarm companies	Information about alarm companies which may requisition responses.
Assignments	The details about an assignment.
Postal numbers	Every existing postal number and the ID of the unit responsible for each of them.
Units	Information about units.
Assignments - units	A junction table to link assignments and units.
Messages	Messages including assignment ID and unit ID.
Events	Events including assignment ID and unit ID.
Devices	List of devices allowed to be used for signing on.
Users	Information about users including usernames and passwords.

Table 3.1: The tables used in the server database

3.2 The Server

This section describes the most important features of the server, the software, and solutions developed for the server. It also describes interactions with Google's servers.

3.2.1 Software & solutions

The server setup is fairly easy. The server software can be installed on basically any server. The software and solutions used at the server are listed below.

Apache The web daemon used can be any web daemon supporting *PHP*, and the most natural choice is the Apache web daemon, which comes with a plugin PHP support [18].

PHP: Hypertext Preprocessor PHP: Hypertext Preprocessor (PHP) is a server-side scripting language used to create documents accessible over the HTTP protocol [50].

MySQL The database system implemented is MySQL. MySQL is the "World's most popular open source database" [44] and is well-supported by web scripting languages [51].

The system utilizes *PHP Data Objects (PDO)* (see below) and is the database type can easily be changed.

PHP Data Objects PHP Data Objecs (PDO) is an interface used to connect to databases and supports several database types [49]. PDO uses the same syntax to run queries on every database supported, which makes switching between databases easy. To switch to another database type, the only code modification needed is in the PDO construction call. The example below shows a new PDO construction call connection to a MySQL database:

```
$dbh = new PDO('mysql:host=mysql.example.com;
dbname=database', user, pass);
```

The PDO construction call is only located in one place in the source code, in the database handler file. PDO was released with PHP version 5 [48].

jQuery jQuery is a JavaScript library used to create dynamic websites without reloading the website [39]. Among other usages, jQuery is able to fetch content from the server while the user types in a HTML form.

libcurl Libcurl is a library used for file transfers, supporting several protocols, among them HTTP and HTTPS, and programming languages such as PHP [26].

Google Geocoding API Geocoding is used to translate addresses into coordinates. Google Geocoding API provides a method for retrieving coordinates by sending a HTTPS GET request to Google's server which responds with the address information either in *JSON* or *XML* formats, where JSON is the recommended format [28].

JavaScript Object Notation JavaScript Object Notation (JSON) is a standard format for exchanging text data and is in this system used for exchanging data between the local server, Google's server and the clients. JSON can be compared to an array written as text with key and value pairs [40].

Extensible Markup Language Extensible Markup Language (XML) is another standard for exchanging text data. XML is developed by the World Wide Web Consortium (W3C) [25].

XML Path Language XML Path Language (XPath) is a language used to run queries on a XML document. A query returns one or more *nodes* from the XML file. A node can contain one or more elements or sub-elements. XPath is the XML query language recommended by the World Wide Web Consortium (W3C) [20].

To use a web daemon and HTTP is a natural choice for sending and receiving text data in this scenario, when one of the parts in the transfer, the client, connects to the server and expects a response. By using a server-side scripting language, documents can be created for the clients to receive and process. The clients can send information to the server by using queries in the *Uniform Resource Locator (URL)*. The syntax of a HTTPS URL is shown below [21].

```
https://<host>:<port>/<path>?<searchpart>
```

Where the *host* part is the address of the server, the *port* is the port of the server and the *path* is the requested document. The *searchpart* is sent to the requested document which process the information.

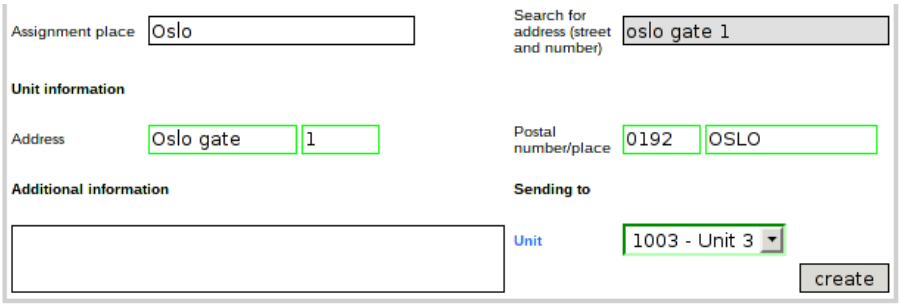

Figure 3.3: Address lookup in the assignment management interface

PHP and Perl are two server-side scripting languages which can be both be used with the Apache web daemon on any platform [50] [46]. They are both very functional web scripting languages and are great choices for web scripting. PHP was selected because that is the language which I have most experience with.

3.2.2 Receive & create assignments

The server is theoretically supposed to receive assignments from the affiliated alarm companies. To integrate the system with the current well-established system would have been a laborious and unnecessary process. The security company dispatch distributes assignments all over Norway and utilizes advanced systems for it. Therefore, an interface for manually copying the assignments has been developed, called *Secdroid Assignment Management Interface*, as explained in Section 3.5. That will not have an impact on the test results, as the timer for calculating response times starts when the assignment has been created using the assignment management interface.

Figure 3.3 shows an example of the lookup function, in the assignment management interface. When an address is typed into the input field at the top right of the figure, the corresponding address and unit is put into the input fields below. The address and coordinates are looked up using the *Google Geocoding API* and the unit is looked up in the server's database. The coordinates are needed to place a marker on the map and to calculate distances. Figure 3.4 illustrates the process of creating a new assignment and the process is explained below:

1. *jQuery* is used to look for changes in the address search input field.

2. When the input field has been changed, jQuery queries another PHP file with the place and address as parameters.

3. The queried PHP script uses *libcurl* to send a HTTPS GET request to the *Google Geocoding API*. The request has the place and address as parameters.

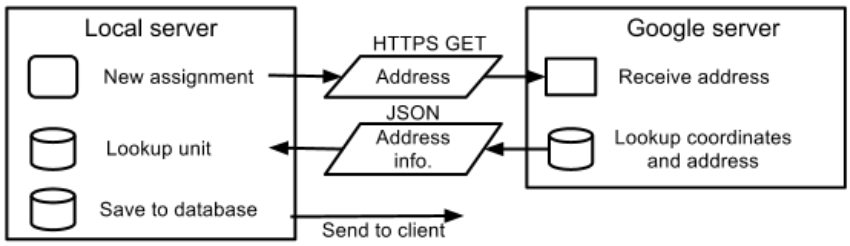

Figure 3.4: Creation of a new assignment

4. The API returns a JSON formatted file containing detailed information about the address. The correctly spelled address,[1] the postal code and the coordinates are retrieved from the JSON file.

5. The PHP script uses the postal code retrieved from the JSON file to lookup the unit responsible for that area.

6. The unit ID and the address information are formatted into a new JSON string and it is returned to the jQuery script.

7. jQuery receives the information, and put it into the HTML form in the assignment management interface, as shown in Figure 3.3.

3.2.3 Distribute assignments and messages

The server notifies the clients about new assignments and messages using push technology. Push technology is a method of "pushing" messages from a server to the client. It is the opposite of pull technology, where the client periodically contacts the server to check for updates. By using push technology, messages get delivered to the clients as soon as they are created, which is essential for this system, since its goal is to limit the assignment response times.

Since the clients are powered by batteries, the clients' energy consumption is important. To start a connection to the server to check for updates is energy consuming. In this system, relatively few messages need to be delivered to the clients as fast as possible. That makes push technology the best solution in regards to battery consumption and efficient responses. Android comes with an integrated system for pushing messages to Android smartphones, which is called *Cloud to Device Messaging Framework (C2DM)* [30].

C2DM is a service integrated in the Android operating system, which lets a server send push notifications to a particular device.[2] The notification is not meant to contain any data, but to tell the phone to connect to the server in order to fetch updates [30].

There are a few alternatives to using C2DM in addition to polling. Text messages can be sent out and retrieved by the application, or a custom

[1] Google Geocoding API supports small spelling errors.

[2] C2DM needs Android version 2.2 or higher and the Google Play application installed.

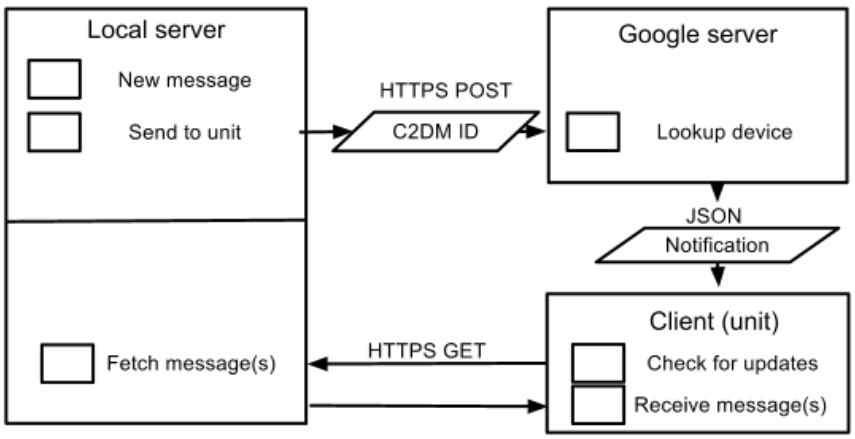

Figure 3.5: Server to client assignment push using C2DM

made push service can be developed. With the usage of text messages, additional costs for sending text messages are included. An own push service can be created, but it will increase the battery consumption to have another active connection which is needed for a push service. C2DM can be used by every application and system service on an Android phone, which limits the active connection to one.

The process from an assignment has been created at the server until it is delivered to the client is illustrated in Figure 3.5 and explained below:

1. First, the server queries the database to fetch the *C2DM registration ID* of any clients signed on to the unit the assignment is intended for. The registration ID is an ID number which is unique to the Android device. The Android device registers with Google the first time the application is started in order to receive the registration ID, see Section 3.3.8.

2. Then, cURL is used to send a HTTPS POST request to Google's server. The requests contains the registration ID of the receiver and the applications authentication key, see Table 3.2 for the full cURL request.

3. After receiving the request, Google locates the device. If the device is online and reachable, a push notification is sent to the device.

4. When the application on the client receives the push notification, it starts an HTTPS connection to the local server to fetch the message.

Forward assignments

When a unit wants to forward an assignment to another unit, it sends a special request to the server. The request includes the *unit ID*, the *assignment ID* of the assignment it wants to forward and the *unit ID* of

cURL option	Value
URL	https://android.apis.google.com/c2dm/send
HTTPHEADER	Authorization: GoogleLogin auth=*authentication key*
SSL_VERIFYPEER	false
POST	true
RETURNTRANSFER	true
POSTFIELDS	registration_id = '*C2DM registration ID*', collapse_key = '0'

Table 3.2: cURL options for sending a c2dm message

the receiving unit. When the server receives this request, it sends out an assignment to the receiving unit as explained in this section. The sender unit ID is included, to let the receiver know who the assignment was forwarded from.

Android's new push to device system

Since the development of the application described in this thesis was completed, Google has launched a new and improved system to replace C2DM. The new system is called *Google Cloud Messaging for Android (GCM)*. C2DM is now referred to as the GCM beta version and is no longer maintained, but it will continue to work [30].

The GCM service has the same principles as C2DM with a few improvements and differences [4]:

- While the C2DM had a quota limitation on the number of messages each server could send, quotas are removed in GCM.
- GCM requests can include the message (up to 4KB) and the message will be forwarded to the device.
- GCM consumes even less battery than C2DM. The Android C2DM library is improved and the device does not need to establish an extra connection to the server, both which decreases battery consumption.
- The process is faster than using C2DM since the message is included in the GCM request and forwarded to the client.

The process of sending a message from the server to a client using GCM is illustrated in Figure 3.6. It is similar to the process of using C2DM. The difference is that the message that is included in the request is sent to the Google server. Once the Google server has received the request and authenticated the sender, the message is forwarded to the client. This omits the last step when using C2DM where the client has to connect to the local server to fetch the updates, as shown in Figure 3.5.

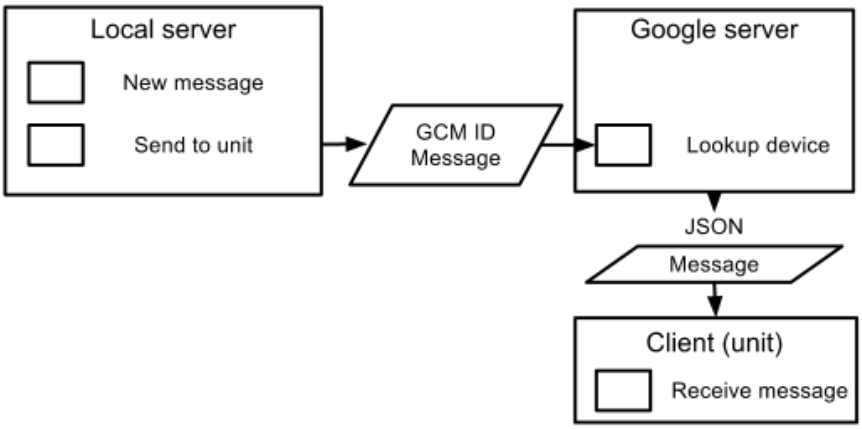

Figure 3.6: Server to client assignment push using GCM

cURL option	Value
URL	https://www.google.com/accounts/ClientLogin
HEADER	true
POST	true
POSTFIELDS	accountType = 'GOOGLE' Email = *account email* Passwd = *account password* source = *application ID* service = *ac2dm*
RETURNTRANSFER	true
FRESH_CONNECT	true
HTTPAUTH	CURLAUTH_ANY
SSL_VERIFYPEER	false

Table 3.3: cURL options for requesting a authentication key

Registering with Google

To be able to use Google's push-to-device services, the application needs to be registered and the C2DM service has to be activated [29]. This is done at the Google API management website. It requires a Google account and the unique developer defined application ID. When the application has been registered, an *authentication key* has to be obtained. The authentication key has to be included in push message requests.

The server obtains the authentication key by sending a HTTPS POST request to Google. Once the key has been obtained, the key is stored in the server's database. If the key is missing, wrong or expired, the server automatically requests a new key. The Google server will return HTTP Status Code 401 Unauthorized if the key is rejected.

The authentication key is obtained by sending a HTTPS POST request to Google containing a Google username and password and the application

ID. The server uses cURL to send the request. The full request is listed in Table 3.3.

3.2.4 Server to client message format

When one of the clients sends a request to the server, it expects a response from the server. The server generates a text document with key-value-pairs which the client downloads and processes. The document has to be in a format the client software can understand and process. There are two popular formats which fulfill the requirements and which are discussed in this section: XML and JSON.

XML

XML is a markup language where values are encapsulated in open and closed tags. XML is a more complex and strict language than JSON. XML provides the feature of adding attributes inside the tags, to provide additional information about the value. The processing of XML documents becomes more complex when all the features of an XML document have to be taken into account.

There are several libraries for processing XML documents, such as XPath. XPath is good for searching and filtering a XML document. The wide range of features that come with XPath are mostly superfluous in this scenario, where the entire document has to be processed and there is no need to subtract only parts of the document.

An example of a XML document is written below. The example shows a part of the document the client receives after signing on.

```
<?xml version="1.0" encoding="UTF-8"?>
<userid>1337</userid>
<hash>093ad56c01 ...</hash>
<unit>
        <_id>1001</_id>
        <name>Unit 0</name>
        <dep>1</dep>
</unit>
<unit>
        <_id>1002</_id>
        ...
</unit>
...
```

JSON

JSON is a simpler format than XML. The structure of JSON resembles the structure of a data array written in plain text. JSON is easy to generate and process. PHP provides a built-in function, *json_encode* which encodes an array or a string into JSON. JSON can easily be translated into objects and the values are easy accessible.

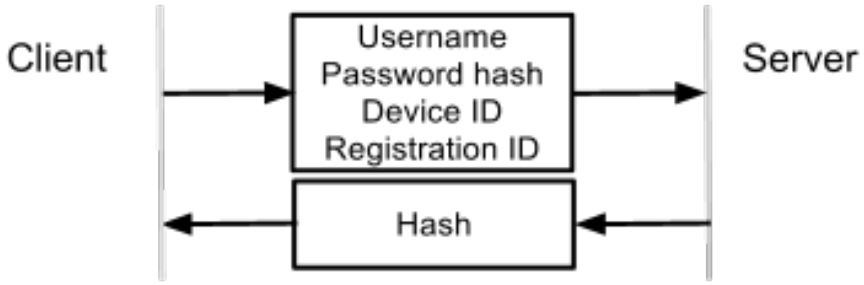

Figure 3.7: The authentication process

The same output as in the XML example is written in JSON format below.

```
{
        "userid": 1337,
                "hash": "093ad56c01 ...",
        "units": [
                {
                        "_id": 1001,
                        "name": "Unit 0"
                        "dep": 1,
                },
                {
                        "_id": 1002,
                        ...
                },
                ...
        ]
}
```

The simplicity of generating and processing JSON makes it the preferred choice for this scenario. XML provides more features and is great for many purposes, but for transmitting simple data where the entire document has to be processed, JSON is considered the best choice.

3.2.5 Authenticate the clients

The clients need to identify and authenticate to be able to receive sensitive data. Figure 3.7 shows the login process. The client sends the username and password provided by the user. It also sends the device's ID and the C2DM registration ID to identify the device. The server generates a hash and sends it to the client. The hash is used for identifying the client throughout the session. Without a valid hash, the client will not gain access to any information at the server.

The process at the server after a login request has been received from the client is explained more thoroughly below:

1. First, the server queries the user table in the database to check the username and password.
2. If a match is found, the server queries the device table to check if the received device ID is in the table of approved devices.
3. If the device is approved, the server checks if the registration ID sent by the client matches the registration in the database. The registration ID is updated if it does not match.
4. Then, the server generates a hash based on the current time and the registration ID.
5. The generated hash is returned to the client.

3.2.6 Update requests from clients

When the clients receive notifications about new updates, they connect to the server to download the updates. They also connect periodically to report their positions. The update interval is set to every 60 seconds. Not too often to limit battery usage, but often enough to sustain a fairly updated position. The parameters included in the update request are listed below.

- The hash received from the server during the login process.
- A list of IDs of the assignments currently active at the client (if any).
- The IDs of the newest event and message received by the server.
- The current position (latitude and longitude) and the time when the position was obtained.

These parameters are included in any request made by the client to always check for new updates. The assignment IDs are used to synchronize the assignments. The server looks at the IDs received from the unit and compares them with the IDs attached to the unit at the server. Several events could occur at this point: *1*) If an assignment is missing from the list on the server, the server will include the assignment in the reply to the client. *2*) If an assignment in the list is marked as completed at the server, the server requests the assignment to be archived at the client. *3*) If a assignment ID in the list does not exists at the server, it is requested to be deleted. *4*) Finally, if an assignment in the list is marked as not received at the server, the status is updated to "received".

The IDs of the most recent event and message are used to synchronize events and messages. When a new event or message is sent to the client from the server, the IDs of the most recent event and message are included. This makes it easy to check if there are recent log events or messages. When a new assignment is sent to the client, all log events and messages belonging to that assignment are included. An updated list of other units in the area, containing their positions and number of assignments, is also included in the response. This is used to place the other units on the clients' map.

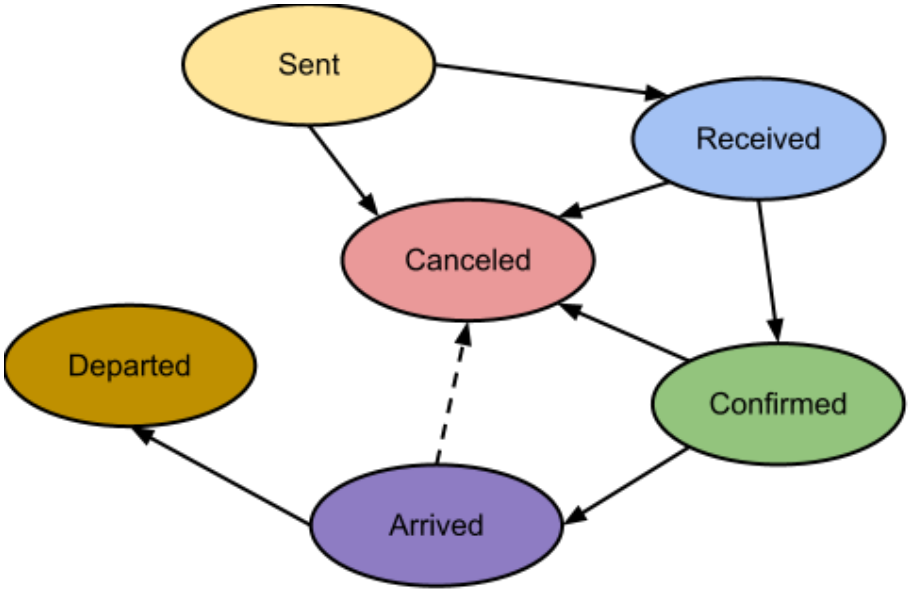

Figure 3.8: Status state-diagram

3.2.7 Status update requests

Figure 3.8 shows the lifetime of an assignment with the status changes that are valid. According to the nature of a response, a unit has to *arrive* before it can *depart* and it has to *confirm* the assignment before it can report *arrival*. The first status is *sent*, which is created when the assignment is attached to the unit. When the unit has received the assignment, the status is changed to *received*. During the process of responding on an assignment, the unit is always allowed to *cancel* the response. When the unit has arrived at a location, it makes less sense to cancel the response, but it is still allowed to cancel the response if, for example, the unit has to leave the location immediately after arrival.

The server is responsible for checking whether or not the status update request is valid. When a status update request is sent to the server, the server checks it according to the state diagram showed in Figure 3.8. The client sends its current status and the server responds with a new (or unchanged) status. Then, the unit updates its status accordingly.

If the client status is ahead of the server, the server will match the client status and return the same status to the client. Table 3.4 shows which statuses the server will return when the client sends a status update. The server may go via states between the client and the server status if the status differs. For example, if the client status says *arrived* and the server status says *sent*, the server will create events for received, confirmed and arrived, as shown in Table 3.4. This should not happen, but could happen in off-line scenarios. The normal status update changes are highlighted in the table.

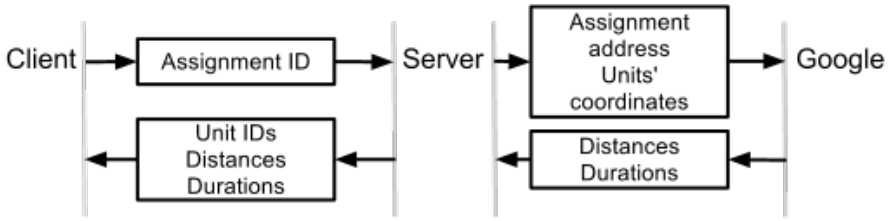

Figure 3.9: The distances and durations look up process

Reporting

When the user has finished the alarm response and is ready to depart from the assignment location, it needs to include a report with the departure status update request. The report contains the most important information in regards of the response: the probable *alarm triggering cause code*, the *measures* taken, *detector numbers* triggered and possible additional *comments* about the response.

The last unit departing an assignment *must* report the probable triggering cause. The other information is only required to be included when needed. The server will only accept an alarm triggering cause and detector numbers from the last unit departing. If several units are attached to an assignment, the remaining units are only allowed to report measures taken and comments. The server treats the measures and comments as a *message* belonging to the assignment. Therefore, remaining units will receive the report as messages when other units depart.

3.2.8 Look up distances and durations

Figure 3.9 illustrates the process of looking up the distances and durations from the other units in the area and to an assignment address. The unit which needs assistance sends the ID (of the assignment it needs assistance with) in an *update request* message to the server. The server uses the assignment ID to look up the address of the assignment. The server also looks up the coordinates of the other available units in the area. The information is sent to the Google API server in an HTTPS request. The Google server responds with a JSON document containing the distance and duration for each coordinate requested, in the order it was requested. The information is then formatted by the server and returned to the unit which requested the information.

3.3 The Client

This section looks into the software developed for the clients and the background information about smartphones and the *Android operating system*. The new system is intended to expand the functionality and increase the user experience by incorporating a more user friendly, more intuitive and better looking application. See Section 3.4 for information

Status at client	Status at server	New status
Sent	Sent	Confirmed
	Received	Confirmed
Received	Sent	Confirmed
	Received	**Confirmed**
Confirmed	Sent	Confirmed
	Received	Confirmed
	Confirmed	**Arrived**
	Arrived	Arrived
	Departed	Departed
Arrived	Sent	Arrived
	Received	Arrived
	Confirmed	Arrived
	Arrived	**Departed**
	Departed	Departed

Table 3.4: Assignment status requests and responses

about the client's features and user interface.

3.3.1 Client device

The client software is developed for smartphones. Smartphones are more advanced than traditional cellphones. In addition to increased functionality compared to traditional cellphones, smartphones are faster, have better and more hardware and more advanced software. Extra hardware found in most smartphones include, a camera, *Global Positioning System (GPS)* and wireless data access via Wi-Fi and mobile broadband.

The first phones that can qualify as being called smartphones were developed in the mid nineties [22]. The term was first used by Ericsson (now: Sony Ericsson) in 1997 when they launched the cellphone "GS 88 Penelope" [53]. The impressive smartphone sales growth started around the year 2009, two years after Apple launched the first iPhone and Google announced the Android platform [19] [37] [2]. The worldwide smartphone sales numbers are shown in Figure 3.10. It shows that the amount has gone up for the last three years, and that in 2012 it is estimated a sale of smartphones is close to 150,000. Smartphones have now a significant share of the world's cellphones. A survey from the beginning of 2012 shows that close to half of the cellphone owners in the United States have a smartphone [36].

The fact that a smartphone is powerful enough to run advanced applications and combines useful hardware like GPS and camera with traditional cellphone features, makes it an ideal device for the purpose described in this thesis. Implementing the system on a smartphone lets the user have both communication and assignments on one device, instead of two devices which is needed with the current PDA system.

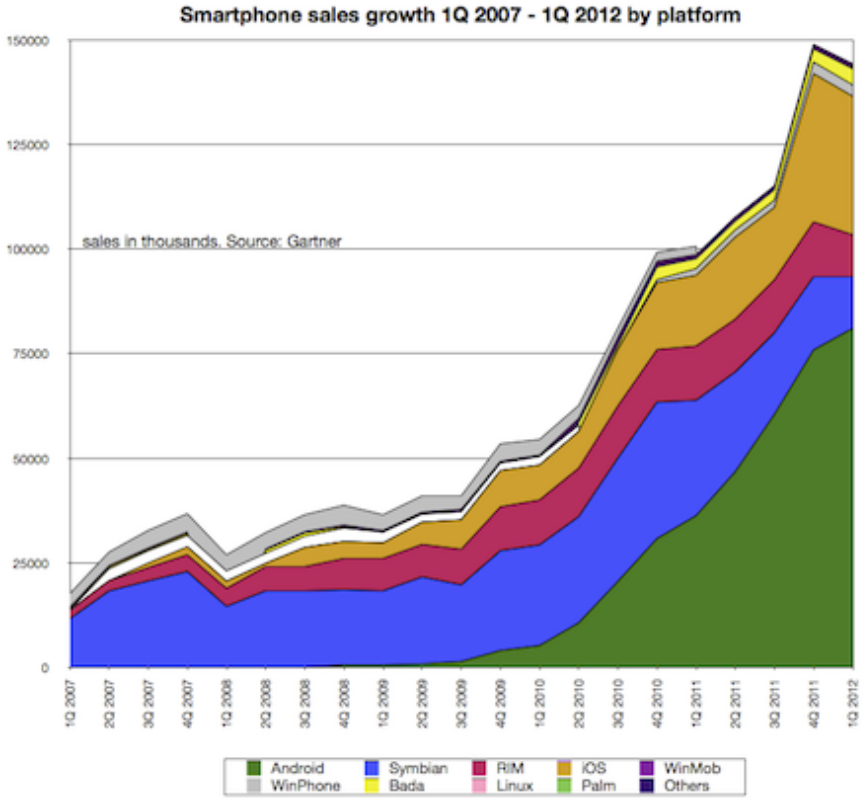

Figure 3.10: Smartphone sales growth from 2007 to 2012, by platform [19]

3.3.2 Client platform

There are several smartphone platform operating systems available: *iPhone*, *Android*, *Windows Mobile* and *SymbianOS*. Android was selected to be used for this system because of the good integration with Google's services, such as maps, push messages and distance calculation, which are needed in this system. Android has, in my opinion, a better and more intuitive graphical user interface than its competitors. *Java* is the programming language used for developing applications for Android. Android also provides a good and powerful API system for developing applications. Furthermore, Android is "an open-source software stack for mobile devices that includes an operating system, middleware and key applications" [5]. Android is based on the Linux operating system and developed by Google [5]. The Android operating system comes with a large selection of smartphones [47]. The large variety of Android phones enables the buyer to select phones based on its preferred criteria, like price, battery capacity or screen size.

3.3.3 Setting up the Android development environment

Before one can start developing Android applications, the Android *Software development kit (SDK)* needs to be installed. The Android SDK can be downloaded from the Android website [12]. When the Android SDK has

been installed, the system image for the *lowest* Android version which one is planning to develop for, has to be installed. The system image contains classes available for that particular Android version (and new versions). Android provides a graphical user interface for installing system images and extra packages. The Android SDK manager is located in the SDK path (*./tools/android*).

Android provides an *Eclipse plugin* for developing Android in Eclipse. Eclipse is a software development environment and is the software recommended by Android for developing android applications [33] [6]. The plug-in provides ways to compile and install the application directly on to the device, watching log outputs and several other tools which can be accessed directly from Eclipse.

3.3.4 Some of Android's development concepts

Android provides a good and powerful API system. This section describes some of the most important principals and features developed by Android and used in the Secdroid application. An brief understanding of these concepts is necessary when reading the following sections.

Android manifest

The Android manifest is a XML file containing important information about the application. The manifest contains the minimum Android version needed to run the application, *permissions* needed and the application's name and icon [10]. The manifest lists all the *activities* used by the application and information about which activity to launch when the application starts up. *Broadcast receivers* and the activity to be launched when a broadcast is received are also listed in the manifest [10].

Permissions

If the application needs to do things that may affect the user's privacy, access the user's personal files or use the device's hardware, it needs permission to do so. The permissions required by the application are listed in the Google play store. The user has to accept the permission before installing the application [15].

Activities

The activity creates a user interface based on a XML coded layout. The activity is what the user sees and interacts with [3]. The developer can extend the Android activity class to develop own activities.

Broadcast receiver

A broadcast receiver is a way of sending information between activities and services or between different applications [11].

Shared preferences

"Shared preferences" is a type of library for storing information that is accessible throughout the application. The information is stored in a text file and will remain stored until the application is deleted or the user explicitly chooses to delete application data [8].

Alarm manager

The *AlarmManager* is a class which provides access to Android's alarm service. The alarm manager is used to set an "alarm" at a specific time in the future. When the time comes, the code file provided– when initializing the alarm– is run. The alarm manager can also be set up to run periodically with a specified time interval [7].

Services

A service is a part of the application that runs in the background. A service can be run not only while the user is navigation in the service's application's activities but also while the phone is idle and when the user is using other applications. There are two types of services: one which runs from when it is started until it is stopped again. The other type starts when the application needs it and stops automatically when it is done executing its code [16].

Both service types can be started in the foreground, as a *foreground service*. If the service is started in the foreground, the system will create a *notification* visible for the user for as long as the service is running. A service started in the foreground is considered to be more important and something the user is actively using and aware of. Therefore, a service started in the foreground is less likely to be killed by the system to save memory [16].

Notifications

Applications can send notifications to the user. Notifications are always visible in the notification bar at the top of the phone's screen. Notifications can be set to play a sound or make the phone vibrate when they are created, to get the user's attention even when the screen is turned off [14].

Intents

Intents are messages that can be sent between different parts of the application. *Activities*, *broadcast receivers*, *services* and the *alarm manager* are all started by creating Intents. Intents are constructed with a class name as the parameter. Then, the desired class is started by issuing commands such as *startActivity* or *startService*. Intents may also hold custom values for the starting intent [13].

Intent filters are entries in the manifest file that tell the application the class to run when intents are received from outside the application [13].

Location Manager

LocationManager is a class that gives the user access to Android's localization service. The location service is shared among the applications and every application can use locations obtained. The location can be obtained either by the device's GPS or by using information from the device's network connection, where GPS locations are more accurate. The location manager can fall back to using network-based locations if GPS is unavailable [17].

3.3.5 Custom made background classes of importance

These classes are background classes which perform operations for the user interface classes, such as communication with the server and handling the database.

Database Adapter

A database adapter class, *DatabaseAdapter*, is shared between the other classes. The class is used to send queries to the application's database. Android provides *SQLite* as the default database type. Each applications gets an own database if needed. SQLite is an embedded SQL database engine which reads and writes to disk files [54]. Android provides a SQLite package with a class for managing the SQLite database [9].

Server Connection

ServerConnection is a class used to connect to the server. The class has a public *connect* method which takes the *type* and a *LinkedList* as parameters. The linked list holds additional key-value pairs needed by the server.

The information intended for the server is included in the URL. For every request (except to sign in and out), the IDs of the assignments currently stored in the client's database, the authentication hash and the position of the client are sent to the server. Read more about this in Section 3.2.6. The Java class *HttpURLConnection* is used to send a HTTPS GET request to the server. A *JSONObject* is created from the JSON document returned from the server. The JSONObject contains methods for reading the different JSON variables, which were stored in the JSON document.

The different reasons to connect to the server are: *1*) signing in, *2*) fetching updates, *3*) reporting status changes, *4*) getting information about other units, *5*) forwarding assignments and *6*) signing out.

Periodic Updater

When the user has signed in, a periodic *AlarmManager* is set to run some code every *60 seconds*. The code is the *PeriodicUpdater* class. It uses *ServerConnection* to check for updates and to report the device's position to the server.

C2DM Receiver

The *C2DMReceiver* is launched when a *C2DM Registration ID Intent* is received from the Google server. The C2DMReceiver stores the C2DM Registration ID in the shared preferences when it is received from Google's server.

Location Updater

The Location Updater also starts when the user has signed in. This is an *foreground service* used to obtain the position of the device. The location updater uses the shared preferences to store the positions obtained. The preferences are accessible by the Periodic Updater, which reports the device's position to the server.

3.3.6 Application life cycle

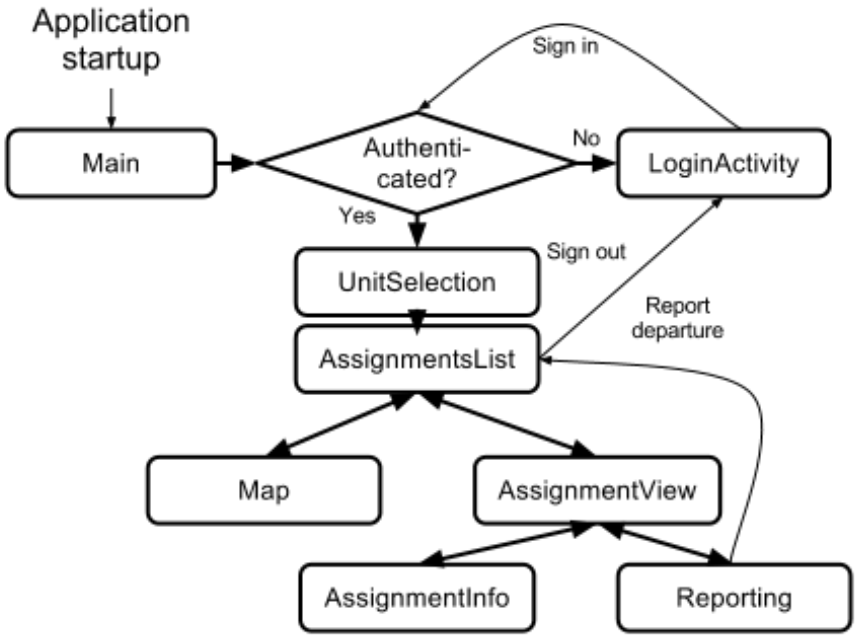

Figure 3.11: The application's activities and how the user can navigate between them

This section describes activities used in the Android application and how the user can navigate between them. An illustration of the application's activities is showed in Figure 3.11 with an explanation below.

Main This is the first activity to be launched when the application is started. The main activity starts by checking if the application has acquired a *C2DM Registration ID*. The activity request a new C2DM ID if no ID is stored in the shared preferences. See Section 3.3.8 for information about how to acquire a C2DM Registration ID.

The main activity is essentially just a blank screen and it starts a new activity immediately. The activity checks the shared preferences if the user is signed on or not. If the user is signed on, the *AssignmentsList* is started, if not, the *LoginActivity* is started.

LoginActivity In the LoginActivity, the user has to enter his or her username and password to sign on. The authentication happens at the server and is handled by the *ServerConnection* class, see Section 3.2.5 for information about how the authentication process is handled by the server. The LoginActivity and the ServerConnection classes are also used to sign the user out, when the user selects it from the *AssignmentsList* activity. The sign-out needs to be registered at the server. The sign out is approved once it has been registered. When it is approved, the user details are deleted from the shared preferences.

UnitSelection This is where the user selects which unit to sign in to after it is authenticated.

AssignmentsList This is the first activity the user sees throughout the session, after it is authenticated. It has options for viewing the map, for signing out and for opening assignments. A more detailed explanation of the usage of this activity can be found in Section 3.4.2.

If the user wants to open the map, the *Map* activity is started. If the users wants to signed out, the *LoginActivity* is started to take care of the sign out process. If an assignment is selected from the list, the user is taken to the *AssignmentView* activity.

Map The map activity shows a map containing units and assignments. See Section 3.4.1 for a more detailed description about the Map activity.

AssignmentView This activity shows details about an assignment and options for administrating it. It also contains log events and messages for the particular assignment. See Section 3.4.3 for the usage of this activity.

If the user selects to open the full assignment details, the *AssignmentInfo* activity is started and if the user wants to change the status of the assignment, the *Reporting* activity is started.

AssignmentInfo This is a simple activity displaying information about an assignment. See Section 3.4.3.

Reporting When the user wish to update the assignment status, this activity is started. It utilizes the *ServerConnection* class to send status updates to the server. When the user wants to report *departure*, the report activity input fields that needs to be filled out, see Section 3.4.4. For any other status change, the activity will only display a progress bar. Section 3.2.6 explains how the status update request is handled at the server.

3.3.7 Localization

The unit's location is used for two purposes: first, to determine the distance to assignment addresses and other units, and second, to make the unit traceable for safety reasons.

Each mobile device is equipped with GPS and can get an almost accurate location. There are, however, a few drawbacks with getting the location from the GPS device. The device needs a clear view towards the sky and will therefore not work indoors. GPS also looses accuracy when the device is inside vehicles, especially in areas surrounded by tall buildings. Lastly, GPS usage has a high battery consumption.

To remedy those things, the Android API provides some built-in solutions. The GPS is put into sleep mode between each poll. The polling frequency is set by the developer. A better solution if the usage does not require a very accurate position, is to use network acquired positions. The location manager can fall back to use approximate location data from the access point or base station it is connected to. This application does not need an accurate position signal from GPS and will use positions solely acquired from the device's network connection. This method has a significantly lower battery consumption.

3.3.8 Registering for C2DM

To be able to use Android's push to device system, which is described in Section 3.2.3, the application needs to get a C2DM Registration ID. The registration ID is stored in the application's *shared preferences*. When the application is started, the main class checks if a registration ID exists. If not, the process of acquiring one is started.

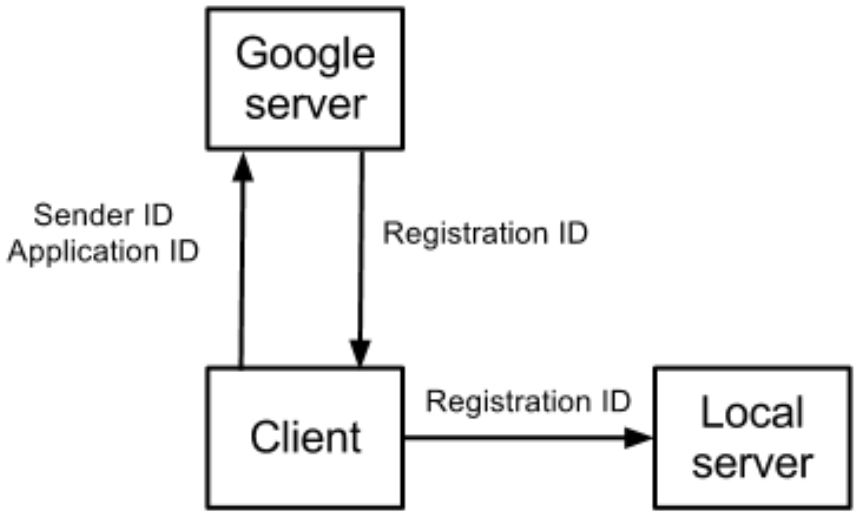

Figure 3.12: The process of registering for C2DM

Figure 3.12 illustrates the process of acquiring a registration ID. The

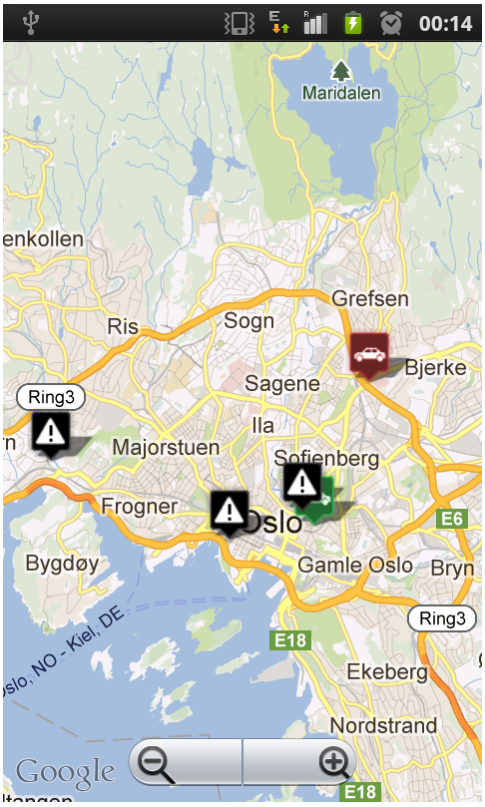

Figure 3.13: Map activity showing assignments and units

application sends an *intent* containing the application ID and a *sender ID*. The sender ID, which has a value of *c2dmsender*, tells the Google server that the application registers for the C2DM service.

After receiving the request, Google's server sends an intent containing the registration ID back to the application. An *intent filter* in the manifest file starts the application's *C2DMReceiver* class when the application receives the intent from Google. The registration ID is extracted from the received intent and stored in the shared preferences.

3.4 Client Design

This application is designed with a goal of being powerful enough to perform the operations needed. In addition to this, the application is designed to intuitive and easy to use. A survey conducted among a group of test users, described in Section 4.2.3, shows that the average user were happy with the design of the application.

3.4.1 Map activity

The map is accessible from the Assignment list activity, see Section 3.4.2, and from the Assignment view activity, see Section 3.4.3. The map uses the Android provided Google maps library to display the units' own

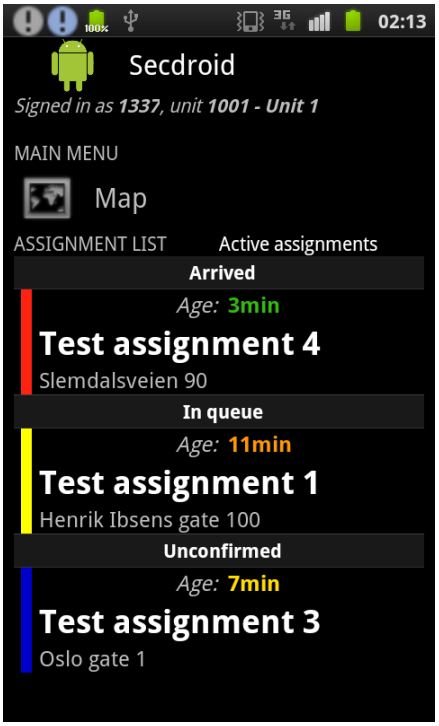

Figure 3.14: The application's home screen containing a list of active assignments

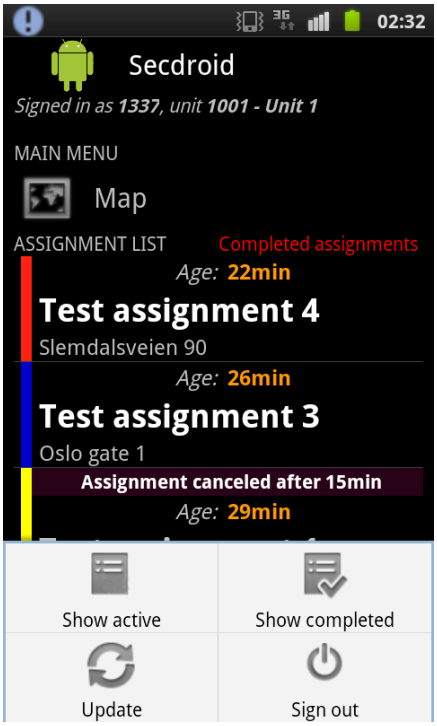

Figure 3.15: The application's home screen containing a list of completed assignments

assignments and other units. Figure 3.13 shows the map with assignments symbolized as explanation marks. The unit using the map and other units are symbolized with green or red car symbols, depending on their availability. The map activity remembers the last map position and zoom level. That makes it convenient to switch between the map and other activities.

Units and assignments can be selected at the map. Selecting a particular unit displays the unit ID and the current number of assignments attached to the unit. Selecting a particular assignment displays the customer name and address. The menu options in the map activity are listed below.

My position which zooms the map into the user's current position on the map.

My area which zooms out to display an overview of the area the users operate in.

3.4.2 Assignment list activity

This is the application's "home screen" and the first activity displayed to the user when it is logged in. The activity displays user and unit information, a menu, and a list of assignments. The login ID and route name is displayed during the login-session and written under the logo and name, as shown in Figure 3.14.

List color	Example alarm types
Red	Robbery, fire
Orange	Elevator, customer assistance
Yellow	Burglary
Blue	Signal loss, power failure, water, high or low temperature

Table 3.5: Alarm event priority guide, including client interface color codes

An important feature of the application is to give a good overview of the assignments in the queue. A list is implemented, showing the current assignments in queue, with their names and addresses. Each item in the list also comes with the time in minutes since the assignment was received. That is useful for the user to help estimate the response time. In addition to this, each list item has a colored left-border, symbolizing the priority of the assignment. There are five color categories, representing the assignment priority. The different events and the categories corresponding to the events are shown in Table 3.5.

Figure 3.15 shows a list of completed assignments. The user has access to the log, messages and information about completed assignments. Completed assignments are only viewable, and the user can not change the status or report of the assignment. Figure 3.15 also shows the home screen's menu with the following options:

Show active which shows the list of active assignments, as shown in Figure 3.14.

Show inactive which shows the list of completed assignments, as shown in Figure 3.15.

Refresh which refreshes the screen in case of new incoming assignments.

Log out which signs the user out.

3.4.3 Assignment view activity

When the user selects an item from the list on the home screen activity, the assignment view activity starts. The assignment view activity contains an event log, messages and more detailed information about the assignment.

Overview

The assignment view activity includes the priority color as explained in Section 3.4.2 and shown in Table 3.5. It's displayed as a line at the top of the activity. The remaining parts of the activity consist of information and options summarized in Figure 3.16 with the numbers explained below.

1. The name/logo of the alarm company the assignment belongs to is displayed.

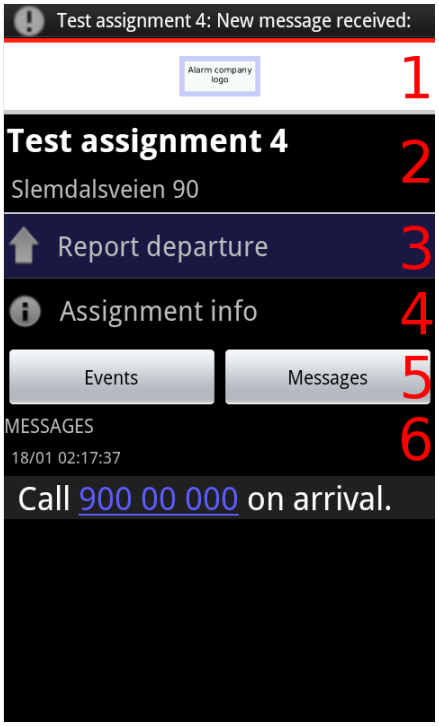

Figure 3.16: Assignment view showing messages

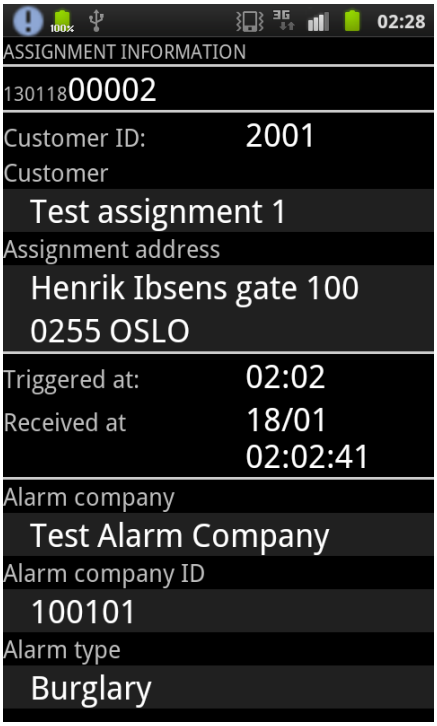

Figure 3.17: The assignment information activity

2. The name of the customer and the location of the assignment is displayed.

3. An option for reporting arrival at or departure from an assignment location is displayed. In Figure 3.16, the button will report arrival when selected. After arrival has been selected, the button text and background color changes; as does the function of the button, as shown in Figure 3.18. When clicked again, it takes the user to the reporting screen shown in Figure 3.24.

4. An option for opening an activity which contains the full information about the assignment is displayed. The information activity is shown in Figure 3.17 and explained further below.

5. Buttons are available for switching between viewing messages and log events.

6. The lower part of the activity, below the buttons for switching between viewing messages and log events, shows additional information selected by the user. The information can be accessed by selecting one of the menu options available.

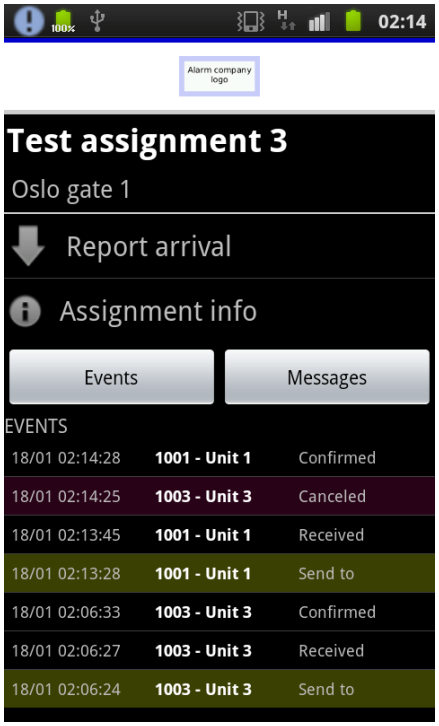

Figure 3.18: Assignment view showing log events

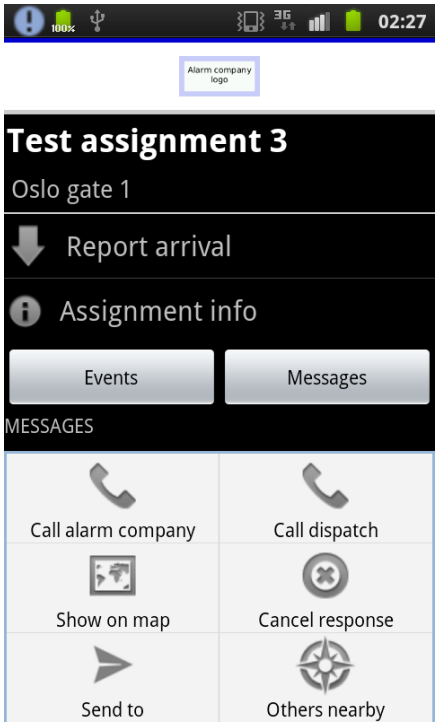

Figure 3.19: Assignment view showing the options menu

Assignment information activity

Figure 3.17 shows a screenshot of the activity containing the full information about the assignment. The following information is included:

- A unique assignment identification number, assigned by the security company.
- The costumer number assigned by the alarm company.
- The customer's name and address.
- When the alarm was triggered and when it was received by the security company.
- The company that ordered the assignment and their unique identification number.
- The assignment type.

The information is used by the security guard to perform a correct response to the assignment.

Assignment view menu options

There are two buttons for switching between message view and log view, as shown as number five in Figure 3.16. Message view is shown in Figure

3.16. Messages are used when the alarm company or the security company dispatch want to include additional information with the assignment. Messages are linked to the assignment and show up for every unit which have received the assignment. Messages can be sent anytime during the assignment life cycle. When another unit reports using a partial reporting scheme, as explained in Section 3.4.4, the information is sent as messages to each of the units attached to the assignment.

The log view shows a log of all the events made by the units attached to the assignment. The log view is shown in Figure 3.18 and also in the background in Figure 3.22. The log view offers an easy way for the units to follow the time and order of the events relevant to the assignment. Important events, such as arrival and departure times, are highlighted using different colors.

Figure 3.19 shows the assignment-view-options menu with additional information and actions that are available to the user. The options are, in the order of appearance:

Call alarm company An option for calling the alarm company which requisitioned the assignment. When the option is selected, the application automatically launches the Android dialing application and calls the phone number linked to the alarm company.

Call security company dispatch This option starts a call to the security company dispatch.

Show on map An option for launching the map activity and zooming in to the location of the assignment.

Abort response This option aborts the unit's response to the assignment. The assignment will still be active, but gets deactivated for the particular unit. This option is useful if the user chooses to abort the response to the assignment. The security company will get notified and can reassign the assignment to another unit.

Send to This option displays other units in the department for the purpose of forwarding the assignment. The "send to" option is explained further below.

Others nearby This option performs the same as the previous option in addition to also calculating the time and distance from other units to the assignment location.

Assignment forwarding

To achieve efficient collaboration between the units, assignments can easily be sent between the units if the need arises. A list of other units can be generated in two ways: either with or without the units' distance to the assignment address. The latter is useful if the designated recipient is already known, to avoid wasting time calculating distance.

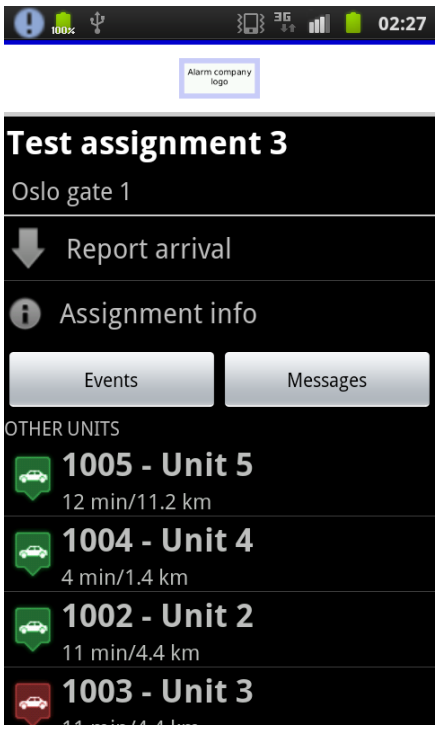

Figure 3.20: Assignment view showing the distance from other units

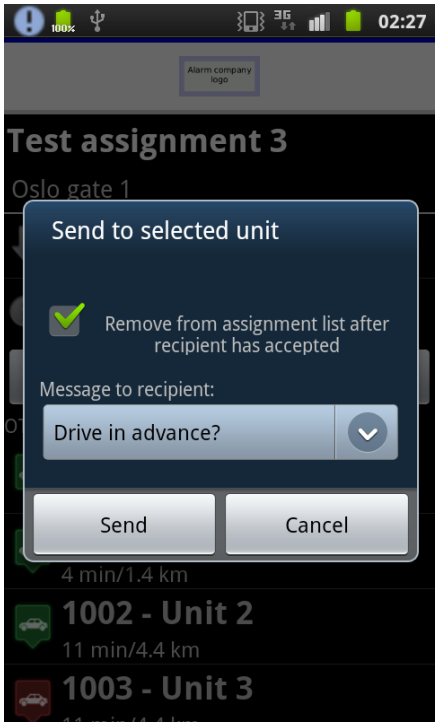

Figure 3.21: Assignment view showing selection for sending the assignment to other units

Figure 3.20 shows a list of other units with the time and distance calculated. The symbol indicates whether or not the unit is attached to other assignments. To send an assignment, the user simply clicks on one of the units in the list. Then a form for sending assignments will appear, as shown in Figure 3.21. The sender's form includes an option for removing the assignment from its list of assignments after the recipient has chosen to accept the assignment. The sender also has to choose a message to send to the recipient from a predefined list of messages. The list consists of the most common reasons for forwarding assignments and lets the receiver know why the assignment was sent to him or her.

Figure 3.22 shows an assignment sent from another unit, which has been opened for the first time. The receiving guard gets information about the sender, with message attached prompting the receiving guard to either accept or decline the assignment.

Canceled assignment

If the alarm company choose to cancel the assignment, the user receives a notification, as described in Section 3.3.4. When an assignment has been canceled, the time elapsed since it got canceled is included with the assignment in the assignment list, as shown in Figure 3.15. When a canceled assignment is opened, the user is presented with three options as

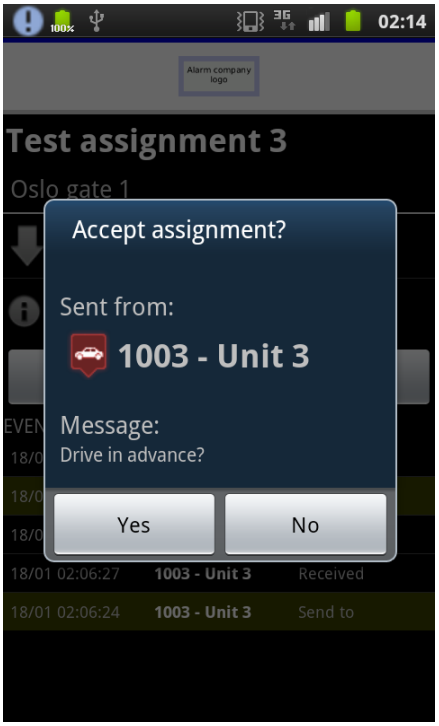

Figure 3.22: Assignment view showing a newly received assignment

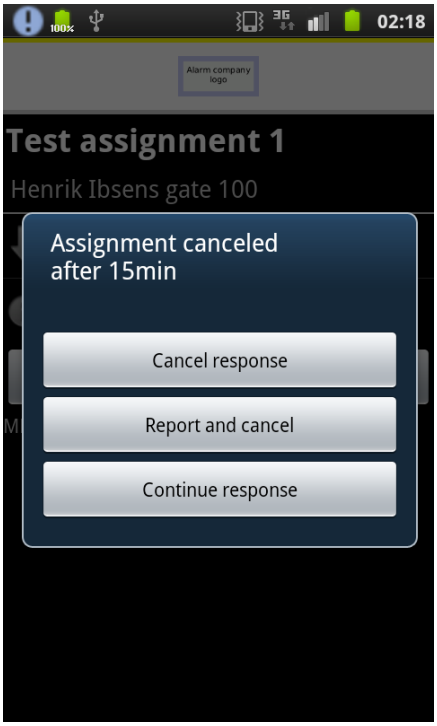

Figure 3.23: Assignment view showing options when an assignment has been canceled

shown in Figure 3.23 and explained below:

Abort response The unit accepts the cancellation, aborts the response and the client archives the assignment.

Report and abort Aborts the response and include a report to the alarm company saying that it was canceled after the cancellation time threshold.

Continue response If the user thinks the assignment was mistakenly canceled based on information not known to the alarm company, the user can continue, normally after clarifying with the alarm company.

3.4.4 Reporting activity

When the user is finished at an assignment and clicks the "departure" button, the reporting activity is launched. The user reports back the most important information after the investigation of the assignment. Figure 3.24 shows the reporting activity. The form consists of the following fields:

Cause code The probable alarm triggering cause. There is a full list of available cause codes and a search box to let the user filter the list, as demonstrated in Figure 3.25.

Detector numbers To list the triggered alarm detector numbers.

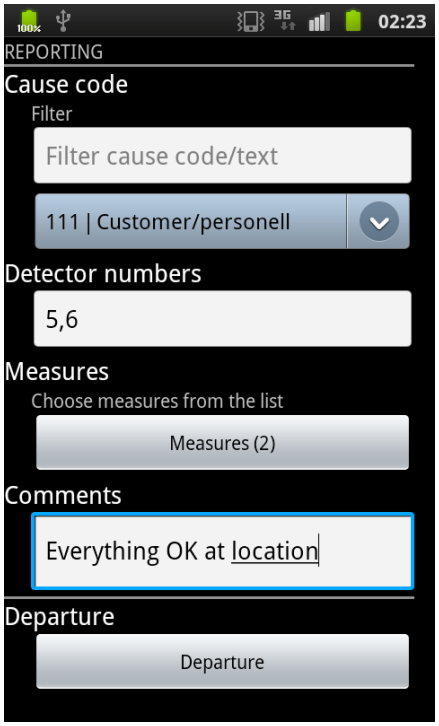

Figure 3.24: Reporting activity, full reporting screen

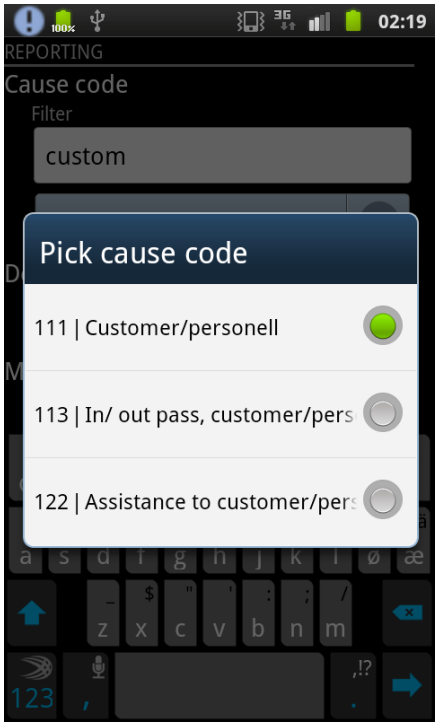

Figure 3.25: Reporting activity, alarm triggering cause selection

Measures The user can select multiple measures performed at the assignment location, as demonstrated in Figure 3.26.

Comments Lets the user, if needed, type any comments to be included with the report.

Departure Button to send the report and finish the assignment.

To avoid duplicate alarm triggering cause codes and ambiguous triggered section listings, if several units are attached to the assignment, only the last unit to report gets the full reporting screen shown in Figure 3.24. Additional units get a simplified reporting screen, as shown in Figure 3.27.

3.5 Secdroid Assignment Management Interface

The security company has a system for distributing assignments. In order to distribute assignments and to control the units and assignment statuses, an assignment management interface has been developed. The interface is only meant to be used for testing purposes.

The interface is a dynamic website, written mainly in PHP and HTML. Figure 3.28 shows a screenshot of the interface. It contains the features needed for distributing assignments and monitoring units and

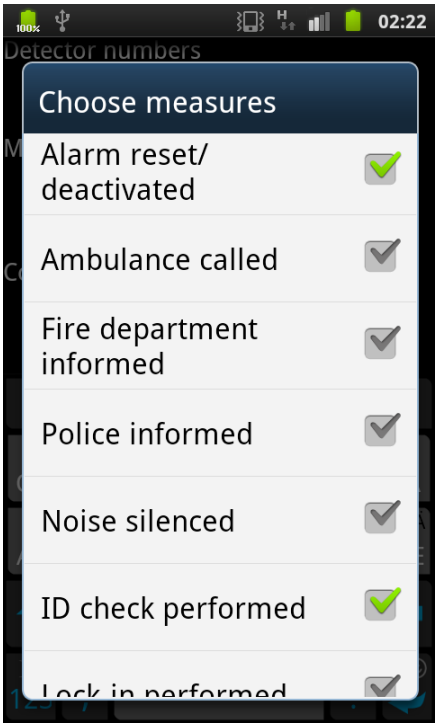

Figure 3.26: Reporting activity, measures selection

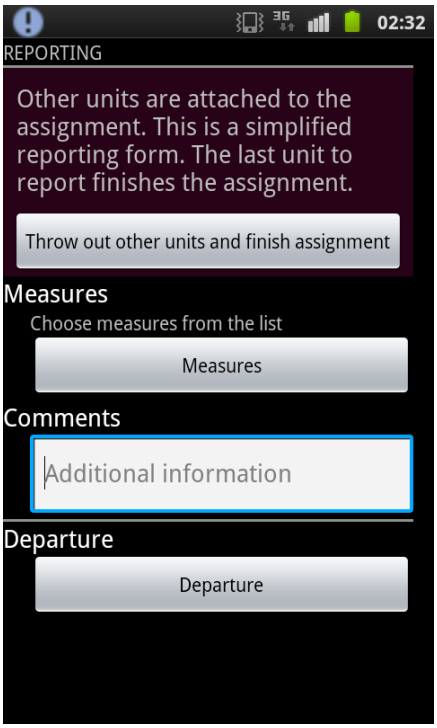

Figure 3.27: Reporting activity, partial reporting screen

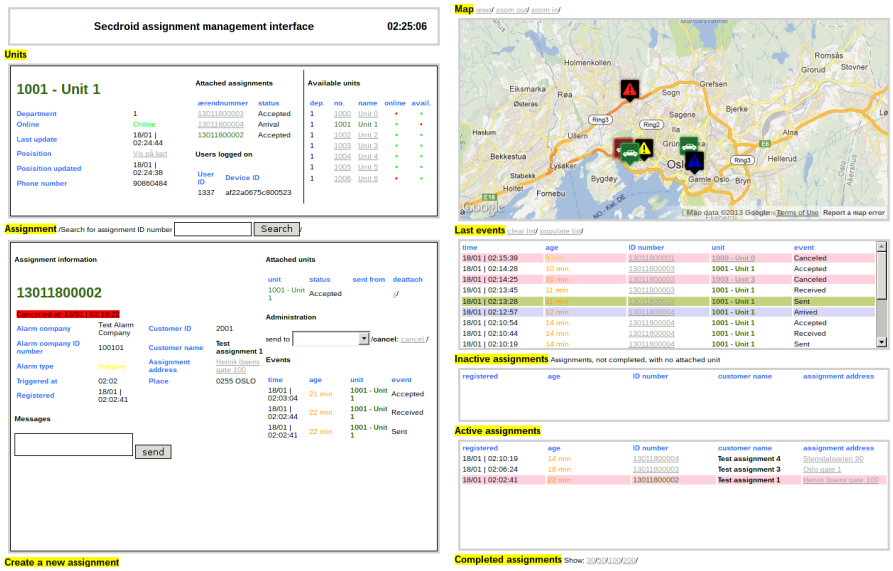

Figure 3.28: An overview of the Secdroid assignment management interface

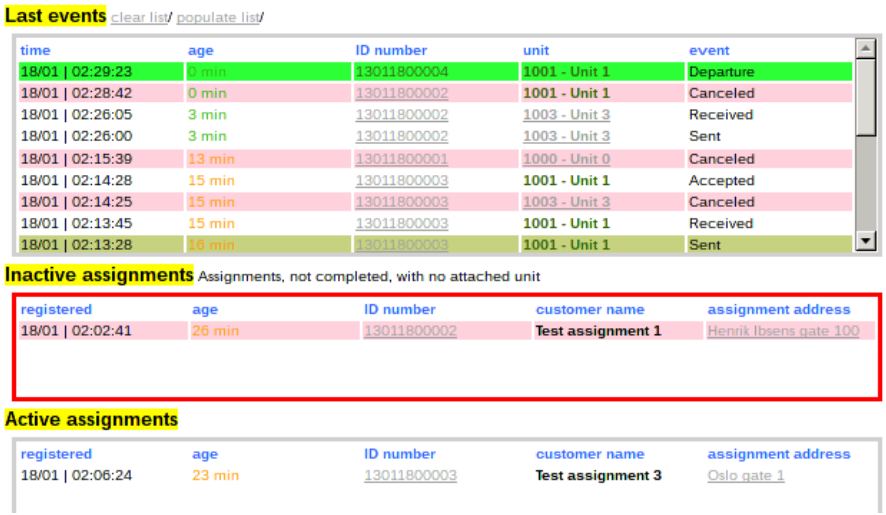

Figure 3.29: Secdroid assignment management interface: Events and assignments lists

assignments. *jQuery* is used to periodically reload sections of the map to give a close to real-time overview of events.

3.5.1 Map

The map gives the user of the interface a complete overview over the locations of the active units and assignments. The map is located at the top right corner of the interface, as shown in Figure 3.28. Addresses and locations throughout the interface can be clicked on to automatically show the location zoomed in on at the map.

3.5.2 Events and assignments lists

The event list shows events reported by any units on any assignment, in the order the events are reported by the units. The event list is the list on the top in Figure 3.29. Important events like cancellations, arrivals and departures are highlighted.

There are three lists showing assignments, regarding the current state of the assignment: *1*) Inactive assignments, *2*) active assignments and *3*) completed assignments. Inactive and active assignments lists are shown as list number two and three from the top in Figure 3.29.

The inactive assignments list shows not completed assignments with no attached units. An assignment should not stay in that state for too long. Thus, the list border turns read when the list is populated, to get the user's attention.

The active assignments list shows the assignments currently active and the completed assignments list shows the last completed assignments.

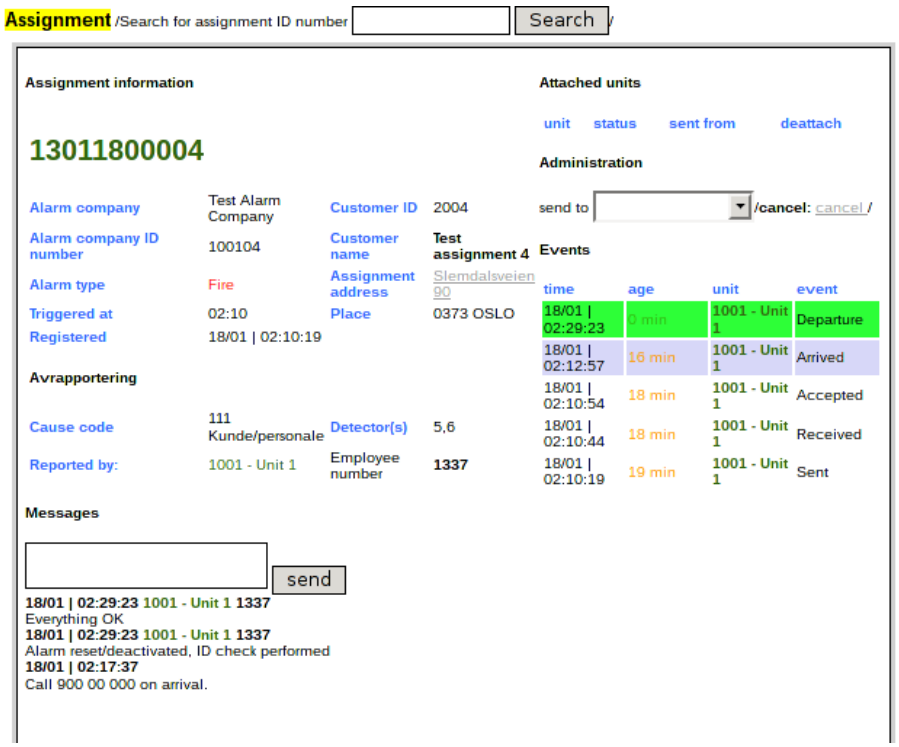

Figure 3.30: Secdroid assignment management interface: Assignment information

3.5.3 View assignment

The view assignment area of the interface shows relevant information about a particular assignments. A screenshot is shown in Figure 3.30. It shows the alarm and customer information, log events regarding the selected assignments, units attached, messages and the report if the assignment is completed. It also includes options for attaching new units and cancelling the assignment.

3.5.4 View unit

The view unit area of the interface shows information about the selected unit, as shown in Figure 3.31. It contains information about the last update, position, attached assignments and user(s) signed on.

3.6 Summary

Secdroid, a new system, has been created with several improvements and added functionality compared to the system currently in use by the security company. Secdroid utilizes maps and calculates distance and duration from units to assignment addresses to improve the users' efficiency. A new and improved list of assignments and reporting section have also been implemented.

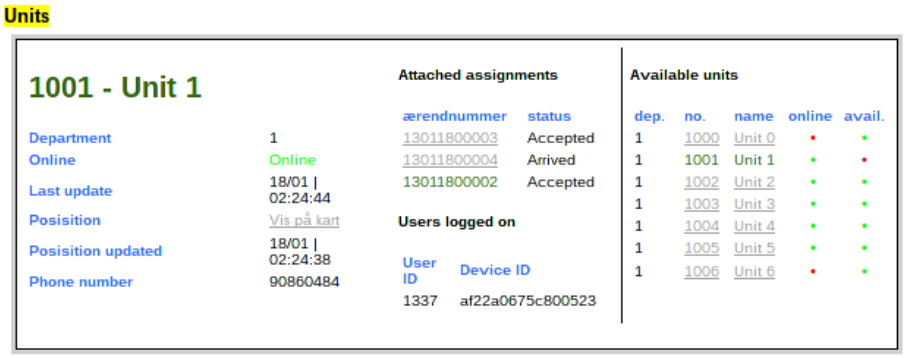

Figure 3.31: Secdroid assignment management interface: Unit information

The clients are developed for smartphones using the Android operating system. The system uses Google services where it is suitable. Google Geocoding API is used to calculate distance and duration from units to assignment addresses and Google push-to-device is used to distribute assignments and messages.

An increased efficiency among the users of the system will likely decrease response times. The usage of push technology to distribute assignments and messages will likely limit the distribution times. In the next chapter, experiments regarding these questions are presented and discussed. A survey conducted among a group of test users will also reveal the users' thoughts about the new system compared to the previous systems used.

Chapter 4

Experiments & Discussion

4.1 Real-Life Testing

In the previous chapter, the design and implementation of Secdroid was described. The Secdroid system is targeted to improve an existing system by fulfilling the requirements in Section 1.2. In order to evaluate the system, several tests have been conducted using actual alarm assignments and alarm response units. The tests have been conducted over three types of works shifts: daytime, evening and weekend daytime. Each user testing the system was asked to fill out a questionnaire. In the questionnaire, questions about Secdroid and questions about Secdroid compared to the security company's previous and current systems were asked. The questionnaire and the results from the survey can be found in Section 4.2.

Throughout this section, results from the test periods have been compared with data from the usage of the current PDA system. The data is from June 1, 2012 to January 1, 2013 and contains about 15,000 assignments. The total number of assignments included in the data from the tests are 167. This includes the assignments from the test days and also assignments from a few days when individual units tested the system.

4.1.1 The testbed

The server

The server used for testing was a *web server* publicly accessible for the clients. The clients need to authenticate with the server to gain access to information on the server. Table 4.1 lists the relevant software and versions installed on the server used for testing.

The clients

The client software was installed on *Samsung Galaxy S Plus* smartphones which the users used during the tests. Figure 4.1 shows a picture of the smartphone type that was used. The relevant software and software versions are listed in Table 4.2.

Software	Version
Linux	2.6.32
Apache	2.2.16
MySQL	5.5.28
PHP	5.3.3
libcurl	7.21.0

Table 4.1: Server software versions

Figure 4.1: The Samsung Galaxy S Plus smartphone [52]

4.1.2 Summary of the test days

The system has been tested for five shifts using every unit in the alarm response department. During those shifts, the system has been tested for a total of 43 hours. Table 4.3 shows a summary of the test days. The table contains the day, time of the day, number of test users, the number of assignments for each day, the number of assignments per hour and the number of assignments per hour per test user. The number of assignments per hour range from 1.75 to 5.00, while the number of assignments per hour per user range from 0.44 to 1.00. The average time spent on one assignment during the test period is 18.5 minutes.

4.1.3 The data basis

To get an understanding of the data basis acquired during the test period, the alarm types from the second half of 2012 have been compared with the types from the test periods. The distribution of the alarm types should be about the same independent of which system being used to distribute the assignments. Figure 4.2 shows the ratio for the different alarm types. It clearly shows that the majority of assignments are of the burglary type, with about 70% of the total assignments. High prioritized alarm types, such as robbery, fire and assistance have a significantly smaller ratio of occurrences.

Table 4.4 shows the percentages of the different alarm types for the

Software	Version
Linux	2.6.35
Android	2.3.5
Android API	8
SQLite	3.7.2

Table 4.2: Client smartphones software versions

#	Day	Hours	# users	# A	A/h	A/user/h
1	Thursday	7am - 3pm	4	19	2.38	0.59
2	Wednesday	3pm - 11pm	4	28	3.50	0.88
3	Friday	3pm - 10pm	4	30	4.29	0.94
4	Sunday	8am - 8pm	5	60	5.00	1.00
5	Wednesday	7am - 3pm	4	14	1.75	0.44
avg				**30.20**	**3.38**	**0.87**
sum				**151**		

Table 4.3: Summary of the test days

second half of 2012 and for the test period. The deviation is smallest for the alarm type with the highest number of assignments (burglary, 10% deviation). Technical and service assignments are are assignments which typically more common at nighttime than during daytime. The fact that the testing was performed during daytime and evenings only, may contribute to explaining the high deviation for those assignment types. Robbery, elevator and assistance alarm types have a very low number of assignments, and do not, by themselves, provide a good data basis for comparison.

4.1.4 Response times

The response time is the time it takes from the moment the assignment is sent out the first time until the first unit arrives at the location. It does not include potential processing time at the alarm company or security company dispatch, since those are not included in the scope of the thesis.

In order to cope with the low number of assignments for some of the assignment types, a graph is created presenting the different assignment

Alarm type	Second half 2012	Test period	Deviation
Robbery	0.9%	0.6%	33%
Fire	4.4%	3.6%	18%
Burglary	69.0%	76.0%	10%
Technical	3.5%	1.8%	49%
Elevator	3.0%	3.6 %	20%
Assistance	2.6%	3.0%	15%
Service	16.7%	11.4%	32%

Table 4.4: Deviation between the second half of 2012 and the test period

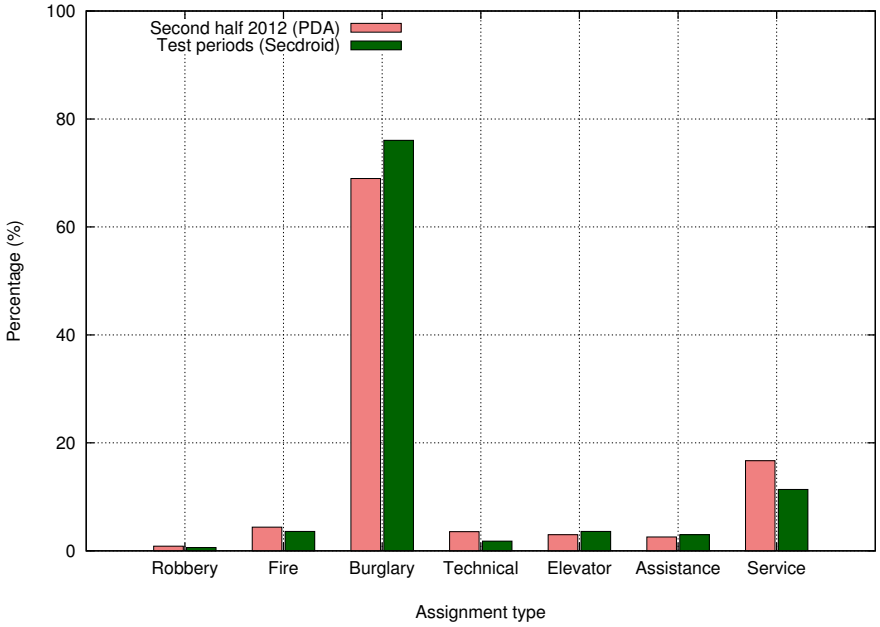

Figure 4.2: Assignments by alarm type

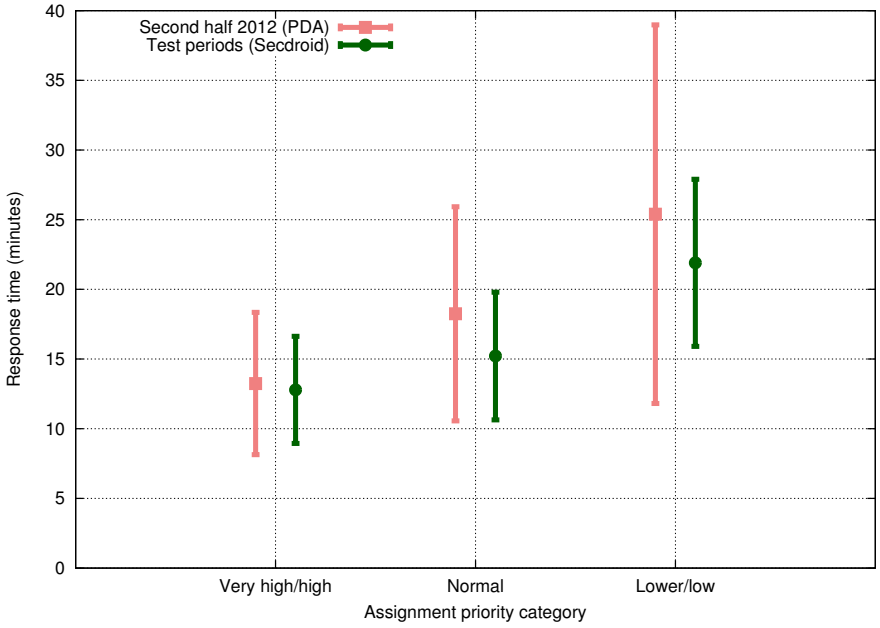

Figure 4.3: Response times for the test periods and second half of 2012 (while using the PDA system) compared

	Response time in minutes		
Category or priority	Second half 2012	Test periods	Improvement
Very high/High	13.2	12.8	3.45%
Normal	18.2	15.2	16.61%
Lower/Low	25.4	18.0	29.28%
Average	**19.3**	**15.7**	**18.93%**

Table 4.5: Improved response times by assignment category

priority categories, as shown in Figure 4.3. The categories are based on the values in Table 2.1. The lines in Figure 4.3 show the average response times by category based on data from the second half of 2012, while the PDA system was in use, and the response times from the test periods. The error lines show the standard deviation of the results.

Figure 4.3 shows that the response times for every assignment priority category are shorter during the test periods than the response times during the usage of the PDA system. As expected, the response times increase for both periods for the lower prioritized categories. However, the increment was greater for the second half of 2012, while using the PDA system, than for the test periods. While only 0.4 minutes separate the two averages for the very high/high categories, 7.4 minutes separate the averages for the lower/low categories. The response times for the PDA system increase with 91.8% from the very high/high categories to the lower/low categories, while the response times from the test periods only increase by 40.5% between the same categories of priority.

Table 4.5 summarizes the values from the graph in Figure 4.3. It shows the average response times in minutes from the second half of 2012 and from the test periods and the test periods improved response times in percentages. The improvement is greatest for the lower/low priority categories, with 29.28% Those are the categories which had the highest response times for the second half of 2012 period, and thus had the greatest room for improvement. The improvement was significantly lower for the very high/high categories, but even for those categories, we see an improvement of 3.45%. With the normal category, which includes about 70% of the assignments, the response time improvement is 16.61%. The average improvement for every assignment is 18.93%.

The numbers suggest that high priority assignment categories are prioritized by the users. The response times for the very high/high categories are low both for the Secdroid test periods and for the PDA usage period. The improvement while using Secdroid becomes visible in the normal and lower/low categories. This is likely to be caused by the users' assistance requests rate. While using the Secdroid system, which gives the user a good overview over the other units' workloads, the users are more likely to ask for assistance. The numbers are higher for the period while the PDA system was in use, which suggests that lower prioritized assignments were put in the receiving unit's queue and responded to after the response to higher prioritized categories. Improved response times was

one of the goal with this thesis, and these results show that the goal has been accomplished.

4.1.5 Users requesting assistance

This section looks into when and how often users request assistance from other units. This has been done to try to reveal if an improved overview and an easy forwarding function has an impact in the assistance request rate. Assignments may be forwarded to another unit and let the other unit respond to the assignment instead of the unit who originally received the assignment. Units may also assist on one assignment, where one unit drives to the location to perform an initial check and then wait for the unit responsible for that particular sub-area. Due to lack of information in the database received from the PDA system, only the latter scenario is available.

Period	Percentage
Second half 2012 (PDA)	6.3%
Test period (Secdroid)	9.2%

Table 4.6: Percentage of assignment with more than one attached unit

Table 4.6 shows percentage of assignments where more then one unit responded. There is an increase of 46% during the usage of the Secdroid system. It is likely that the increase is a result of a better overview over other units in the area and an easy assignment forwarding. This supports the theory from the previous section.

Example assistance request scenario

Figure 4.4 shows a map with units and assignments at one point during the test period. One unit (the second to the left) was occupied at the first assignment when it received a new assignment. The new assignment was a confirmed burglary, and needed a fast response. The unit that received the new assignment needed assistance, and used the other units' distance to assignment function. The map shows the other units. The unit in the top left corner was not available, while the other two units were available.

Table 4.7 shows the units' estimated time to the assignment address and their availability. This is the same information that unit 1 received in this scenario. Help was requested from unit 3, which was found to be the closest unit to the assignment where help was needed.

4.1.6 Assignment distribution times

The assignment distribution time is the time it takes from when the assignment has been sent out from the security company dispatch until it is received and downloaded on the users device. Unfortunately, the current system does not keep track of when the assignment has been received on

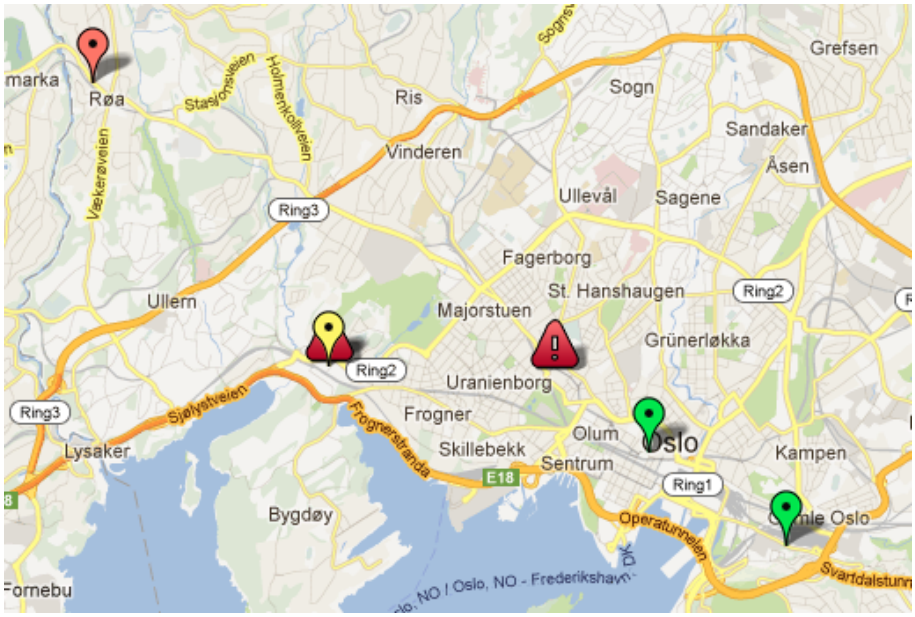

Figure 4.4: Example of a assignment forwarding scenario

Unit #	Duration to assignment	Available
1	12min 24sec	No
2	9min	No
3	**4min 55sec**	**Yes**
4	7min 44sec	Yes

Table 4.7: Information about units shown in Figure 4.4 where units are numbered from left to right in the figure

the PDA, see Section 2.3.2, and data for comparison with Secdroid is not available.

Assignments should, in a normal scenario, be distributed and downloaded to the device within few seconds. However, there are off-line scenarios and times when no one is signed into the unit to which the assignment is sent to. Those scenarios delay the transmission, but not because of technical limitations. This section focuses on the time it takes to distribute assignments in an ideal scenario, where the recipient is online and signed on. Figure 4.5 shows average distribution times. To exclude high distribution times caused by scenarios described in the previous paragraph, the graph in the figure shows average distribution times with increasing percentage omitted when calculating the average.

With none of the long distribution times omitted, the average distribution time is on 17.1 seconds. The graph starts out even with about 12% of the high values omitted, where the average distribution time is 4.8 seconds. This suggests that around 12% of the distribution happens during temporarily off-line scenarios, which corresponds with the experience from the test periods.

Figure 4.6 shows the percentage of assignments distributed in each

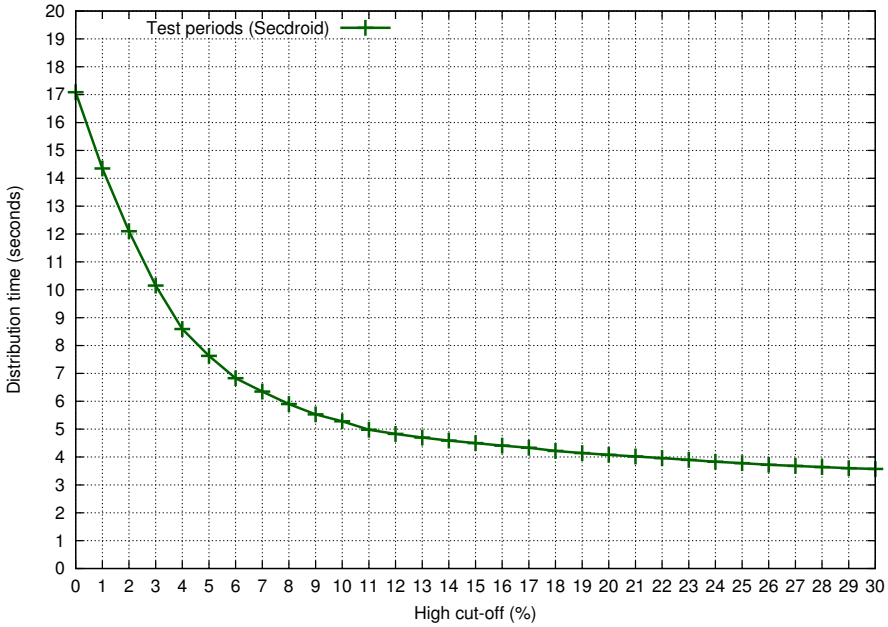

Figure 4.5: Assignment average distribution times

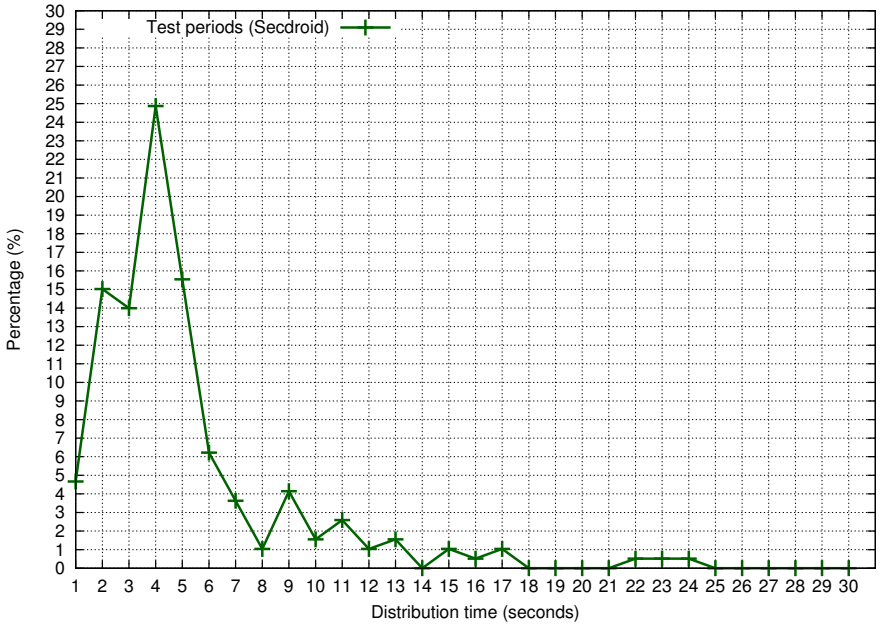

Figure 4.6: Assignment distribution times by seconds

second up 30 seconds. The graph shows that 80% of the assignments are distributed in less than seven seconds.

4.2 Feedback

A survey was conducted among the users testing the new system. This was done to find out which functions the users found most helpful and to look at their experience with the new system compared to the current and previous systems. The results of the survey and the questionnaire given to the testers are presented in this section.

4.2.1 Questionnaire

The users who tested the new system were asked to fill out a questionnaire. The questionnaire asked the users about the different systems, and let the user fill out comments about their experiences during the test period. 14 users answered the questionnaire. That is about half of the department's employees. The questions asked in the questionnaire are listed below.

1. New function: Other units' distance to the assignment address
   This function helped me performing my tasks more efficiently.

2. New function: Send assignments to other units
   This function helped me performing my tasks more efficiently.

3. New function: Map showing assignments and other units
   This function helped me performing my tasks more efficiently.

4. User friendliness
   The client solution/software is easy to use and understand.

5. Assignment list overview
   The client solution/software gives me a good overview over my active assignments.

6. Reporting
   It is easy to report back after completing an assignment using the client solution/software.

7. Mobility
   The device easy to handle and carry around.

8. Reliability
   The solution/software is reliable (few/none errors, crashes or unexpected behaviour).

9. Comments

10. Suggestions

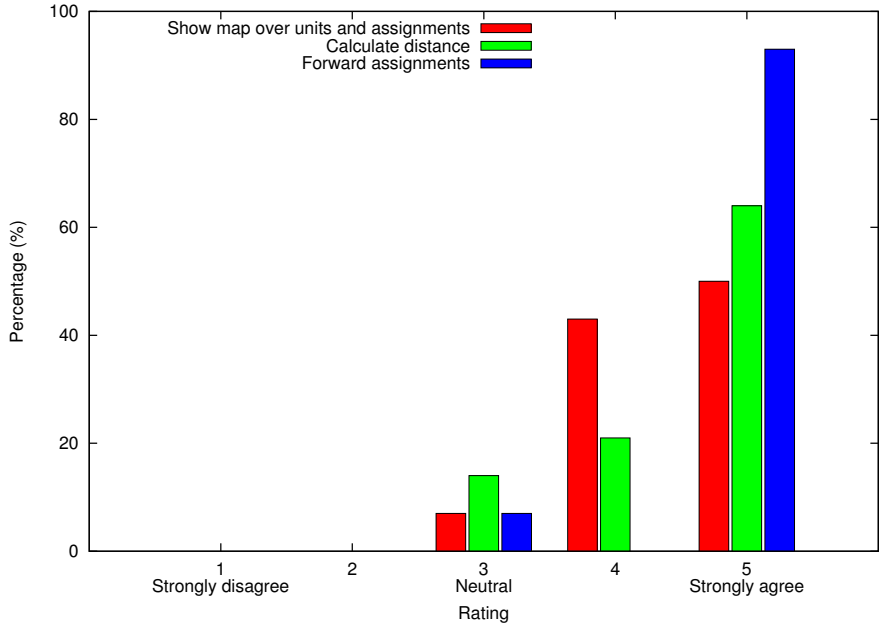

Figure 4.7: The new functions were helpful in performing the tasks more efficient

Function	Average score $\pm$ std. dev.
Show map	4.43 ± 0.7
Calculate distance	4.50 ± 0.8
Forward assignments	4.86 ± 0.5
Overall	$\mathbf{4.60 \pm 0.7}$

Table 4.8: Rating of the new functions' popularity

In question 1 through 8, the users were given statements and were asked to rate their level of agreement: strongly disagree, disagree, neutral, agree or strongly agree. The scale is a five-point *Likert scale* [41]. The options are numbered from 1 (strongly disagree) to 5 (strongly agree), where a higher number is a better rating. For question 4 through 8, the users were asked to compare *Secdroid* with the system currently in use by the security company (*PDA*) and the system previously used by the security company (*text messages*).

4.2.2 New functions' helpfulness

The users were asked to rate some of the added functionality in the Secdroid system. The rating is based on how helpful the users found each function to be, in performing alarm responses more efficiently. The functions mentioned in the survey were:

- Show map over units and assignments (see Section 3.4.1).
- Calculate distance from other units to assignments (see Section 3.4.3).

- Forward assignments to other units (see Section 3.4.3).

Figure 4.7 illustrates a summary of the responses. The figure clearly shows that the users found the new functions to be helpful. Table 4.8 lists the average rating of each function. The overall rating of the new functions is 4.60 out of 5.00.

The most helpful function according to the ratings is the function which lets the user forward assignments. 93% of the users strongly agreed to the statement that this function was helpful and it received a rating of 4.86. This is a function which has been highly anticipated requested for in the PDA system currently in use.

The second highest rated function for helpfulness is the function which lets the users calculate distance from other units to assignments. 64% of the users strongly agreed to the statement that this function was helpful, and it received a rating of 4.50. This function is more helpful with a higher number of users. Since the system was tested with only four or five users, the need of the function was not as big as it would have been with a larger number of users.

The map function gives a nice overview of units and assignments, but it is not the most helpful function when it comes to performing efficient responses. The distribution of answers were almost half and half divided between score 5 (strongly agree) and 4 (agree), respectively 50% and 43%. This function got a rating of 4.43.

4.2.3 Usability

A software's usability describes easily a person could use the software. The definition of usability can be split up into five components [1]:

Learnability How easy it is for the user to learn using the software. Preferably without training.

Efficiency How efficiently the user can use the software once he or she has learned how to use it.

Memorability Does the user remember how to use the software after a period of not using it?

Errors Does the user have any user generated errors? If so, how often and how severe are the errors which occurs?

Satisfaction How satisfied is the user with the software? A more user friendly software generally increases the user's satisfaction.

The PDA system has received a high amount of negative feedback about its usability:

- The user interface provides a high degree of non-intuitive solutions which require the user to be told how to accomplish tasks.

- The PDA provides complex ways of performing tasks which could be solved in simpler ways, for example the reporting part (see Section 2.3.4).
- The PDA frequently crashes (see Section 2.3.5). This is not only frustrating, but it also takes up time and decreases the efficiency.
- The users are generally not satisfied with the PDA and have called for improvements, updates and new functionality since its release.

The old system which utilizes text messages for sending out assignments is a simpler system, but it also lacks of functionality.

- The user needs to be taught how to use the system, as there is no user interface for it.
- The system is simple, but requires a high degree of manual reporting and information flow, which is time consuming.
- The user needs to remember different codes needed for reporting, which are easily forgotten when they are not used on a regular basis.
- The system has a high amount of user errors as the information flow is manual and can easily be misinterpreted.
- The system gave experienced users a sense of achievement, and feedback from the users indicate that experienced users were more satisfied with the old system than inexperienced users. A drawback for both user groups was, however, the lack of functionality in this system.

When designing Secdroid, drawbacks of the previous systems were improved, missing functionality features were added and useful parts of the previous systems were improved. Good usability was one of the goals when initially designing Secdroid.

User friendliness

System	Average score $\pm$ std. dev.
Text messages	3.36 ± 1.6
PDA	3.64 ± 0.9
Secdroid	4.57 ± 0.6

Table 4.9: Rating of the systems' user friendliness

The test users were asked to rate how user friendly they found the different systems. The results of the rating are presented in Figure 4.8 and the ratings are listed in Table 4.9. For the text messages system, experienced users rated it higher than inexperienced users did and Figure 4.8 shows divided opinions about the system's user friendliness. The text messages system got a rating of 3.36 out of 5.00.

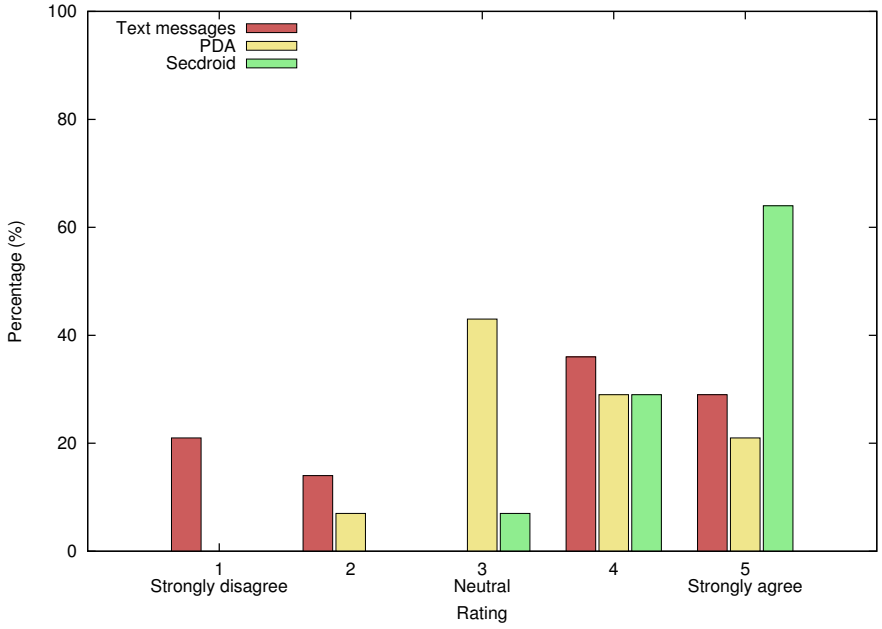

Figure 4.8: The client solution/software is user friendly

The PDA system, which provides a user interface for performing the tasks, gets a more coherent rating. The largest amount of the test users, 43%, rated it to be "neutral" user friendly. The PDA got a rating of 3.64, a little bit higher than the text messages system. The PDA has a lot to go on to be considered user friendly.

The majority of the test users, 64%, rated Secdroid to be very user friendly (5, strongly agree). Secdroid scores high when it comes to user friendliness and is the best rated system out of the three systems tested, with a rating of 4.57. It can still be programmed better, but the goal of increasing the user friendliness has accomplished.

Assignment list overview

System	Average score $\pm$ std. dev.
Text messages	2.43 ± 1.7
PDA	3.71 ± 0.7
Secdroid	4.36 ± 0.7

Table 4.10: Rating of the systems' assignment list overview

The test users were asked to rate how user friendly they found the The assignment list is connected to the system's usability. It is an important feature and should help the user keep track of what is happening with all units in the best possible way. Figure 4.9 presents the test users' opinions about the system's overview of active assignments and Table 4.10 lists the ratings. The text messages system scores badly when it comes to the assignment overview, with a rating or 2.43. Half of the test users rated

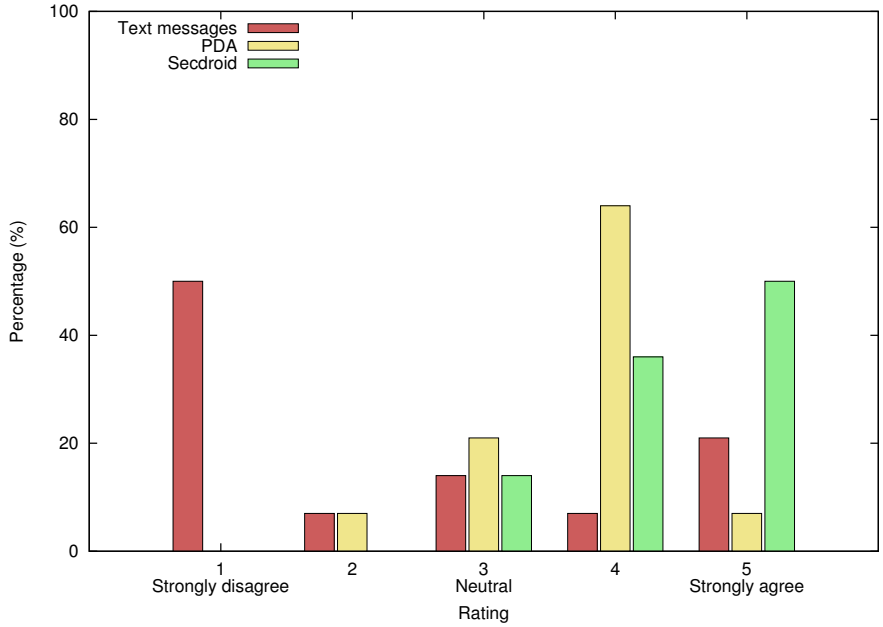

Figure 4.9: The client solution/software gives me a good overview over my active assignments

it 1 out of 5, which represents a "bad overview." The remaining ratings are spread out among the other options. The results are not surprising: New assignments using this system end up in the user's text message inbox and to view the information about a different assignment, the user has to navigate through the messages. The text messages are static and assignment status updates are not updated.

The PDA and Secdroid systems both score well when it comes to assignment overview. Both system have a good overview over active assignments, where assignment status changes are updated in the user interfaces. The rating of the PDA is 3.71, while the rating of Secdroid is 4.36.

Reporting after finishing an assignment

System	Average score $\pm$ std. dev.
Text messages	3.14 ± 1.5
PDA	3.64 ± 1.0
Secdroid	4.57 ± 0.5

Table 4.11: Rating of the systems' reporting solution

The test users were asked to rate how user friendly they found the The reporting section of the system is where severe user errors can occur. The reports must be accurate and the user interface should be intuitive and easy to use. Figure 4.10 shows that the responses for the text messages system and PDA system are spread out over the different rating options. Table

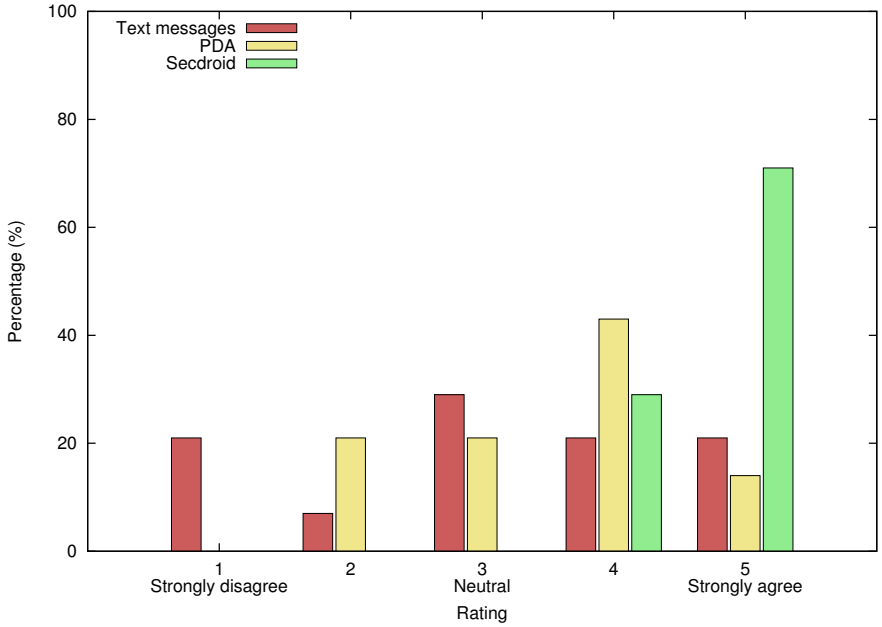

Figure 4.10: It is easy to report back after completing an assignment using the client software/solution

4.11 lists the ratings of the systems. The text messages system got a rating of 3.14 while the PDA got a rating of 3.64. This suggests that the users disagree in how good the reporting systems are, but none of two systems can be viewed as "good," – based on the test users' ratings.

For Secdroid, the results are distributed differently. 71% of the users strongly agree to the statement that it is easy to report back after finishing an assignment and Secdroid got a rating of 4.57, higher than the two other systems. According to the test users, Secdroid has achieved the goal of having a good reporting section.

The devices' mobility

The devices' mobility is not a very important part of the users' work efficiency. Nevertheless, the users were asked to rate the mobility to reveal if they noticed a difference between having a relatively big PDA device and using a smaller smartphone. A smaller device is easier to handle and to carry around.

System	Average score $\pm$ std. dev.
Text messages/Secdroid	4.93 ± 0.3
PDA	2.42 ± 0.9

Table 4.12: Rating of the systems' mobility

The test users were asked to rate how user friendly they found the The text messages system and Secdroid questions are merged for mobility, since they both utilizes mobile phones with a similar size. The results

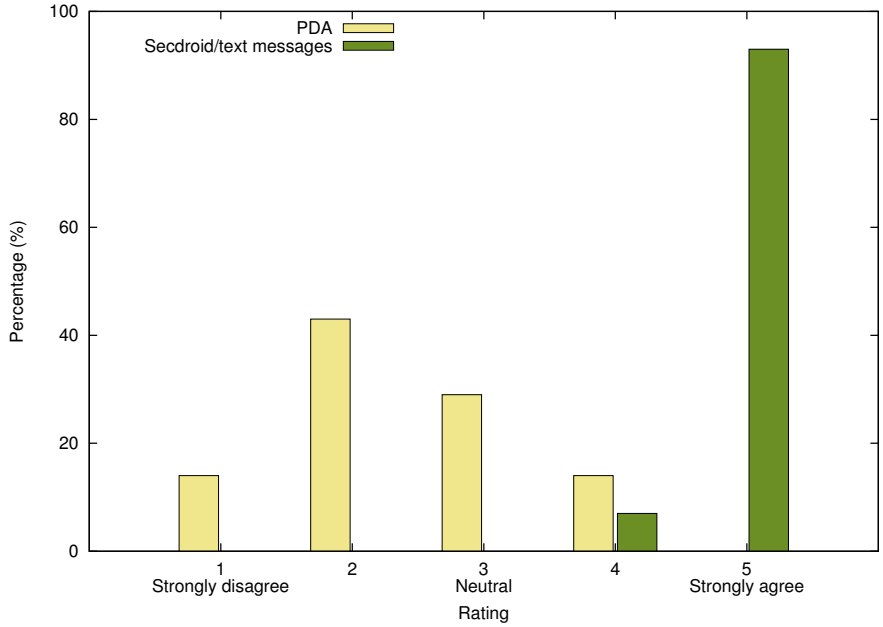

Figure 4.11: The device is easy to handle and carry around

are presented in Figure 4.11 and shows that the majority of the test users strongly agreed to the statement saying that the smartphone/mobile phone is easy to handle and carry around (93%). Table 4.12 lists the rating of the systems' mobility. Text messages/Secdroid got a score of 4.93, significantly higher than the PDA's score of 2.42.

The systems' reliability

Reliability is one of the most important factors when it comes to the general user satisfaction. It can also affect the efficiency when crashes and downtimes limit the user in performing its tasks.

System	Average score $\pm$ std. dev.
Text messages	3.86 ± 1.1
PDA	1.64 ± 0.6
Secdroid	4.79 ± 0.4

Table 4.13: Rating of the systems' reliability

The test users were asked to rate how user friendly they found the The text messages system and Secdroid questions are merged for mobility, The PDA has a history of frequent crashes and downtimes, which is reflected in the users' review. Figure 4.12 and Table 4.13 presents the test users' view on the different systems' reliability. The PDA comes out worst, with a rating of 1.64. The text message system is considered more reliable, with a rating of 3.86. Secdroid is considered to be the most reliable system, with a rating of 4.79.

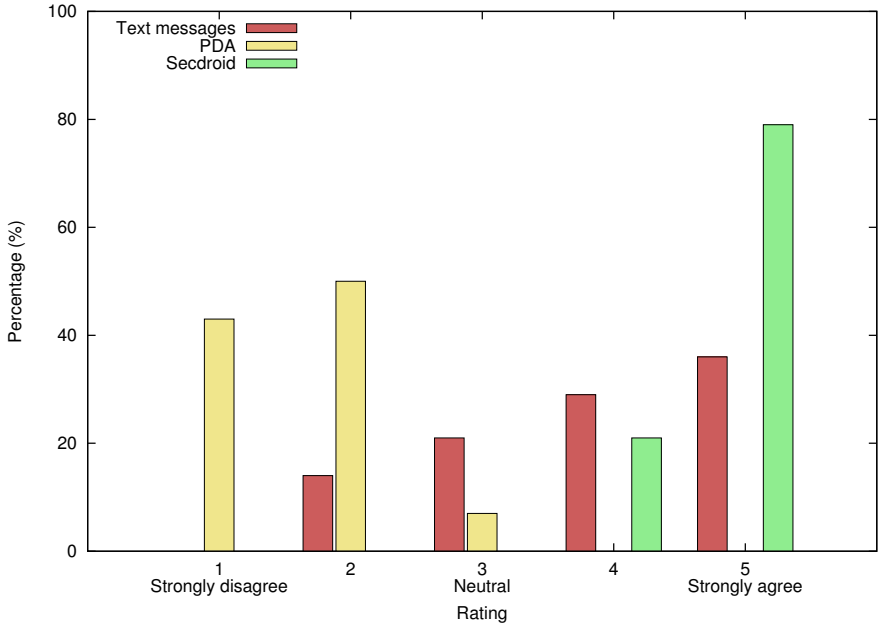

Figure 4.12: The solution/software is reliable

To develop a reliable system was one of the goals when designing system. The feedback from the users shows that the system was reliable during the test periods. The results also supports the information in Section 2.3.5, where the PDA system is described as being unstable.

4.2.4 Comments & suggestions

Test users' comments

The test users were given the opportunity to write down comments about their experience. The comments received are listed below:

- "The map gives a nice overview over available units. User friendly program. It is also nice with the opportunity to use the device for calling, to only have one unit to focus on. Color codes for prioritization is nice for newly employed employees. The smartphone responds faster when navigating through the menus."
- "I'd use the Android rather than the PDA any day"
- "Extremely user friendly. Fast in reporting/finishing assignments. Forwarding between units is very fast."
- "Number 1: Android. Number 2: Text messages. Last place: PDA."
- "Very good. No bugs. User friendly."
- "No errors/crashes during the shift."
- "Android the best by far. I want to use it more!"

- "Android works very good!"

The comments suggest that the test users where satisfied with the Secdroid system, especially when comparing it to the PDA.

Test users' suggestions

The test users could also give suggestions about improvements to the system:

- "Get rid of the PDA."
- "Start using the Android system ASAP."
- "New incoming assignments should fill the entire screen."
- "Frequently used cause codes should be place at the top of the cause code list."
- "Statistics."
- "Include a message when forwarding assignments."

A future improvement could be to make it more visible in the application's user interface when new assignments are received. To put frequently used cause code on top of the list could be a solution for more efficient reporting. However, an improved search mechanism could achieve the same result. Statistics will probably not be a part of the application, as it is not needed to perform the tasks. The last suggestion, to include messages when forwarding assignments, was implemented after that test day and available in the subsequent version.

4.3 Summary

In order to evaluate Secdroid, real-life tests have been conducted using actual units and assignments. The system has been tested with every unit in the alarm response department, four or five units, depending on the test day. The primary goal of the tests has been to measure differences in response times between the PDA system and Secdroid and to determine the Secdroid application's degree of usability.

The average response time has been significantly improved; by almost 19%. Assignments can be split into categories dependent of the priorities of the assignments. Response times very calculated for three different categories: very high/high, normal and lower/low. The results reveal a smaller difference for assignments in the very high/high priority category and a larger differences for the normal and lower/low categories. The differences are respectively 3.5%, 17.0% and 29%. The results show that high priority categories were prioritized by the units independent of the system in use. The larger difference in response times improvement for the two other categories can be explained by a lower threshold for asking for

assistance while using Secdroid than with PDA, since the workload should be the same independently of which system being used. This theory is supported when we look at the percentage of assignments where more than one unit arrived at an assignment. The percentage is 46% higher while using Secdroid than while using the PDA. A good overview, easy forwarding and distance calculations are factors which are likely to have contributed in the greater extend of assistance requests.

The assignment distribution time also affects the response times. The PDA system does not keep track of distribution times, so there is no data to compare the two systems. The average distribution time while using Secdroid is about 17 seconds. However, this number includes distributions while the device has been off-line. When omitting the longest distribution times, caused by off-line scenarios, the average is about four seconds. A distribution time on four seconds is a short time, considering the mobility of the units.

A survey were conducted among a group of test users. The test users were asked to rate some of the new functionality with their view of how it increased their efficiency. The functions rated were: *1*) map showing units and assignments, *2*) distance from units to assignments calculation and *3*) assignment forwarding. Most of the users found the functions to be very helpful in terms of improving efficiency. The users rated the functions from 1 to 5, and the functions got an average rating of 4.6.

The test users were also asked to compare Secdroid's performance and usability with the currently used PDA system and the previously used text messages system. The users were asked about the applications user friendliness, assignments overview, reporting section, mobility and reliability. These questions add up to form a view of the systems' usability. Each category could be rated from 1 to 5. The average ratings from every question comparing these systems are: 3.0 for the PDA, 3.5 for the text messages and 4,7 for the Secdroid system. To determine a "winner" based on the test users ratings is easy. Secdroid clearly stands out from the other systems, in a positive way. The results also show that the test users preferred the previously used text messages system over the currently used PDA system.

Chapter 5

Conclusion

5.1 Summary

Alarm systems are installed in buildings to protect people and assets. When an alarm has been triggered, a response to the alarm is needed. Normally, a security guard is sent to the location where the alarm went off to check the premises and try to find the cause of the alarm. If the security at the location has been breached, caused by for example a burglary or a fire, a fast response may limit the potential damages and loss of assets.

The security company mentioned in this thesis currently use a PDA system for distributing alarm assignments to the security guard units. There are multiple issues and limitations with the PDA system. The PDA has a low usability and a history of frequent system crashes and unreliability. Other similar systems exists, but no systems have been found with the functionality needed.

A new system, called Secdroid, has been developed in the context of this thesis with the goal of increasing the efficiency for each unit and between units and to have an increased usability.

Secdroid has added functionality which the PDA system does not have:

- Secdroid can, if the need of assistance arises, measure distance and durations from other units to an assignment address.
- Secdroid lets the users forward assignments between each other.
- Secdroid implements a prioritization scheme and visually separate assignments based on their type and severeness.
- Secdroid includes a map with information about other units' workload and positions.

With Secdroid in use, less time will be used to administrate assignments and more focus can be given to doing a good job at alarm responses. It would also help finding the best solution when assistance on assignments is needed.

Secdroid is proven to provide faster responses and more efficient work flow. It is easier and requires less effort to get assistance with Secdroid.

The improved overview available with Secdroid resulted in a higher rate of assistance on assignments. The average response time was also found to be almost 19% faster compared to the PDA system.

With Secdroid, lower prioritized assignments were distributed better among the units available, resulting in almost 30% faster responses on those types of assignments. This is likely to be a result of the improved overview which makes it easier for the users to distribute the workload.

The users testing Secdroid were truly satisfied with the system. In a questionnaire, the users rated Secdroid best in every category, and almost exclusively the new system was given a high rating.

5.2 Contributions

A new assignment distribution system has been developed. The system has been developed to improve a security company's efficiency when it comes to alarm assignment responses. By giving the users of the system an increased overview, it will make it easier for them to request assistance when needed. If the units are unaware of the other units in the area's position and workload, the threshold for asking for assistance increases. The new system developed utilizing the units' locations to determine their positions. Each client used by the units regularly connects to a server to report its position and download the positions of the other units. The clients of the system has been developed for Android smartphones.

Server software has been developed to distribute assignments, messages, positions and log events to the clients. The server is responsible for keeping the units' information up-to-date. The clients report the newest message and event ID numbers, and the server checks if newer messages or events exist. The clients also report the current assignment in their databases and the status of the assignments. This enables the server to check if the client holds the assignments it should and that the status of each response, for example: arrived at assignment or departed from assignment, match. The server uses Google's API services to retrieve coordinates for addresses, calculate driving times and to push assignments and messages from the server to the units.

In addition to the client and the server software, a test interface has been created. The interface is integrated with the server software and enables distribution of assignment and messages and monitoring of units for test purposes.

The response times were lower when using the Secdroid system compared to the PDA system. The improvement was bigger for lower prioritized assignments than for high prioritized assignments. This suggests that an increased overview and easier assignment forwarding encouraged the users to involve other units to a higher degree when using the Secdroid system than with the PDA system.

5.3 Further Work

5.3.1 Upgrade to the newest Android cloud-to-device messaging framework

GCM was launched after the completion of the client application, and the system is therefore still using Google's deprecated C2DM service. See Section 3.2.3 for a more detailed explanation of C2DM and GCM. The method for sending GCM messages differs from the method for sending C2DM messages. Software, both at the server and the client, needs to be changed to upgrade to the newer version. With the usage of GCM, application gets more battery efficient, since the phone does not need to connect to the system's server to download updates. A migration to GCM will also decrease the message sending latency. Google provides a guide for migrating from C2DM to GCM [4].

5.3.2 Image upload support

The current solution for sending data from the client devices to the server is by using the HTTPS GET method. By using HTTPS POST for submitting data, image and media uploading can be implemented. This will give the customer an even better response with images of incidents at their building.

5.3.3 Electronic reporting

An improved importing activity can be implemented where the unit add all the details of the operation at the customer's location. A customer portal can be implemented where the customer can view and download the response reports, or the reports can be send to the customer by email.

5.3.4 Automatic distribution

The technology developed for the Secdroid system can easily be used to automatic distribute assignments. Now, each unit is responsible for its own area, and each assignment is distributed to the unit responsible for the assignment's location. With minor changes to the system, assignments can, for example, automatically distributed to the closest available unit to the assignment location.

5.3.5 Utilize tablets

The client software can be implemented specifically for Android tablets. This will give the users opportunity to report and administrative their assignments on a device with a bigger screen. The advanced reporting opportunity mentioned in Section 5.3.3 would be easier with a bigger screen.

Bibliography

[1] Jakob Nielsen's Alertbox. Usability 101: Introduction to usability, February 2013. http://nngroup.com/articles/usability-101-introduction-to-usability/.

[2] Open Handset Alliance. Industry leaders announces open platform for mobile devices, November 2007. http://openhandsetalliance.com/press_110507.html.

[3] Android. Android developers: Activity, February 2012. http://developer.android.com/reference/ android/app/Activity.html.

[4] Android. C2DM-to-GCM Migration document, November 2012. http://developer.android.com/guide/google/gcm/c2dm.html.

[5] Android. What is android?, February 2012. http://developer.android.com/guide/basics/ what-is-android.html.

[6] Android. ADT plugin, January 2013. http://developer.android.com/tools/sdk/eclipse-adt.html.

[7] Android. Alarmmanager, January 2013. http://developer.android.com/reference/android/app/ AlarmManager.html.

[8] Android. Android developers: Using shared preferences, January 2013. http://developer.android.com/guide/data-storage.html#pref.

[9] Android. android.database.sqlite, January 2013. http://developer.android.com/reference/android/database/sqlite/package-summary.html.

[10] Android. The androidmanifest.xml file, January 2013. http://developer.android.com/guide/topics/manifest/manifest-intro.html.

[11] Android. Broadcastreceiver, January 2013. http://developer.android.com/reference/android/content/ BroadcastReceiver.html.

[12] Android. Get the Android SDK, January 2013. http://developer.android.com/sdk/.

[13] Android. Intents and intent filters, January 2013. http://developer.android.com/guide/components/intents-filters.html.

[14] Android. Notifications, January 2013. http://developer.android.com/guide/topics/ui/notifications.html.

[15] Android. Permissions, January 2013. http://developer.android.com/guide/topics/security/ permissions.html.

[16] Android. Services, January 2013. http://developer.android.com/guide/components/services.html.

[17] Android. Using the location manager, January 2013. http://developer.android.com/training/basics/location/ locationmanager.html.

[18] Apache. The Apache software foundation. http://apache.org.

[19] Charles Arthur. Android over 50% of smartphone sale as Nokia and RIM feel strain, May 2012. http://guardian.co.uk/technology/2012/may/16/ android-smartphone-market-50-percent.

[20] Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernandez, Michael Kay, Jonathan Robie, and Jerome Simeon. XML path language (XPath) 2.0 (second edition), 2010. http://www.w3.org/TR/xpath20/.

[21] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform resource locators (url), December 1994. http://tools.ietf.org/rfc1738#section-3.3.

[22] BloombergBusinessweek. Before iPhone and Android came Simon, the first smartphone, June 2012. http://businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone.

[23] CenCom. Trafikkdirigeringssystemer, 2012. http://http://cencom.no/produkter.asp?meny=6,44.

[24] ADT Security Choice. ADT home security benefits. http://securitychoice.com/adt-home-security-benefits.html.

[25] World Wide Web Consortium. Extensible Markup Language (XML). http://w3.org/XML/.

[26] cURL. cURL and libcurl. http://curl.haxx.se.

[27] Maged Dessouky, Randolph Hall, Lei Zhang, and Ajay Singh. Real-time control of buses for schedule coordination at a terminal. *Transportation Research, Part A: Policy and Practice*, 37:145–164, 2003.

[28] Google Developers. The Google geocoding API. https://developers.google.com/maps/documentation/geocoding/.

[29] Google Developers. Using OAuth 2.0 to access google APIs. https://developers.google.com/accounts/docs/OAuth2.

[30] Google Developers. Android Cloud to Device Messaging Framework, September 2012. http://developers.google.com/android/c2dm/.

[31] Evry. Alystra, March 2013. http://alystra.com.

[32] Evry. Evry, March 2013. http://evry.no.

[33] Eclipse Foundation. The Eclipse Foundation open source community website, January 2013. http://eclipse.org.

[34] Google. Google play. https://play.google.com/store.

[35] Scania Group. Scania fleet management, September 2012. https://play.google.com/store/apps/details? id=se.scania.fm.

[36] Devindra Hardawar. The magic moment: Smartphones now half of all U.S. mobiles, March 2012. http://venturebeat.com/2012/03/29/the-magic-moment-smartphones-now-half-of-all-u-s-mobiles/.

[37] Matthew Honan. Apple unveils iPhone, January 2007. http://macworld.com/article/1054769/iphone.html.

[38] ITnews for Australian business. Google launches fleet management service, June 2012. http://www.itnews.com.au/News/305908,google-launches- fleet-management-service.aspx.

[39] jQuery. jQuery: The write less, do more, JavaScript library. http://jquery.com.

[40] JSON. Introducing JSON. http://json.org.

[41] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):44–55, 1932.

[42] Wei-Hua Lin and Jian Zeng. An experimental study on real time bus arrival time prediction with gps data. *Transportation Research Record*, 1666:101–109, 1999.

[43] Johannes G. Lomsdalen, Jon Morten Owen, Tom Erik Iversen, and Jon Andersen. Oslo taxi trafikkdirigeringssystem, 1998. Prosjektrapport i IN166 - Universitetet i Oslo.

[44] MySQL. MySQL: The world's most popular open source database, February 2012. http://www.mysql.org.

[45] GPS Business News. Google enters fleet management market, June 2012. http://www.gpsbusinessnews.com/ Google-Enters-Fleet-Management-Market_a3714.html.

[46] Perl. The Perl programming language. http://perl.org.

[47] phandroid. Android phone fans, February 2012. http://phandroid.com/phones/.

[48] PHP. PDO - introduction. http://www.php.net/manual/en/intro.pdo.php.

[49] PHP. PDO drivers. http://www.php.net/manual/en/pdo.drivers.php.

[50] PHP. PHP: Hypertext preprocessor. http://php.net.

[51] PHP. What can PHP do?, February 2012. http://www.php.net/manual/en/intro-whatcando.php.

[52] Samsung. Galaxy S Plus. http://www.samsung.com/ie/consumer/mobile-devices/smartphones/android/GT-I9001HKDO2I.

[53] Stockholm Smartphone. Stockholm smartphone history, 2010. http://stockholmsmartphone.org/history/.

[54] SQLite. SQLite home page, January 2013. http://sqlite.org/about.html.

[55] RT Traffic. Device fleet management, July 2011. https://play.google.com/store/apps/details?id=com.cg.android.devicemanagement.beta.