Bartosz Wieczorek

THE APPLICATION OF AUGMENTED REALITY TO ENHANCE THE COMMUNICATIVNES OF A USER INTERFACE.

Case: The vehicle terminal

Thesis
CENTRIA UNIVERSITY OF APPLIED SCIENCES
Information Technology
May 2018

ABSTRACT

Centria University of Applied Sciences	Date May 2018	Author Bartosz Wieczorek

Degree programme Degree programme in Information Technology	
Name of thesis THE APPLICATION OF AUGMENTED REALITY TO ENHANCE THE COMMUNICATIVNES OF A USER INTERFACE. Case: The terminal of vehicle.	
Supervisor Dr Grzegorz Szewczyk	**Pages** 44+2
Supervisor Dr Grzegorz Szewczyk	**Working-life supervisor** Jari Isohanni

Augmented reality during recent years gained popularity. The most common use can be found in game market, but there are many fields in ordinary life where Augmented Reality can be used. Application LUUTA (Finnish - broom) has been created to improve an efficiency of workers who are maintaining roads during winter in Kokkola Industrial Park. An application is part of project called BILINE.

The main purpose of the thesis work was to design an application for Android platform which has elements of Augmented Reality in it. It is hard to register work time and tasks, for instance, when employees are working for various customers during a work day in wide area, which is Kokkola Industrial Park. This application can collect a data about current executed work by laborers with all needed information like task, localization, speed or direction. All data is stored in a server database. Moreover, an application provides information about current issues in Kokkola Industrial Park area and can display this information with support of Augmented Reality technology.

Key words
AR, Augmented Reality, vehicle terminal

ABSTRACT

Centria University of Applied Sciences	**Data** Maj 2018	**Autor** Bartosz Wieczorek

Kierunek
Informatyka Stosowana

Temat pracy
ZASTOSOWANIE TECHNOLOGII ROZSZERZONEJ RZECZYWISTOŚCI DO ZWIĘKSZANIA KOMUNIKATYWNOŚCI INTERFEJSU UŻYTKOWNIKA. Przypadek: Terminal pojazdu służbowego.

Instruktor Dr inż. Grzegorz Szewczyk	**Strony** 44+2
Promotor Dr inż. Grzegorz Szewczyk	**Opiekun ze strony zleceniodawcy** Jari Isohanni

Zastosowanie technologii rozszerzonej rzeczywistość jest obecnie coraz bardziej popularne. Stosuje się ją głownie w produkcji gier komputerowych, ale również znajduje szerokie zastosowanie w innych dziedzinach życia codziennego. Aplikacja LUUTA (*[fin]: miotła*) została stworzona, by podnieść efektywność zimowego utrzymania dróg na terenie Parku Przemysłowego w Kokkola (*Kokkola Industrial Park*). Aplikacja jest częścią większego projektu zwanego BILINE. Głównym celem pracy było zaprojektowanie aplikacji, wykorzystującej elementy rozszerzonej rzeczywistości, która byłaby uruchamiana na urządzeniach mobilnych działających pod systemem operacyjnym Android. Na terenie Parku Przemysłowego Kokkola działa wiele przedsiębiorstw i firm. Służby utrzymania dróg wykonują więc prace dla różnych klientów. Rejestracja wykonanych prac, ich czasu trwania oraz zleceniodawcy nie jest więc zadaniem prostym, szczególnie jeżeli weźmie się pod uwagę rozległość terenu. Aplikacja LUUTA jest w stanie zbierać te dane jak i aktualną lokalizację, prędkość oraz kierunek poruszania się. Wszystkie te dane są przechowywane w bazie znajdującej się na serwerze systemu. Ponadto, aplikacja dostarcza informacje o zagrożeniach występujących w Parku Przemysłowym Kokkola za pomocą wyświetlania odpowiedniej informacji z wykorzystaniem technologii rozszerzonej rzeczywistości.

Słowa kluczowe
AR, Rozszerzona Rzeczywistość, terminal samochodowy

ACRONYMS

1. AR – Augmented Reality
2. HTTP – Hypertext Transfer Protocol
3. MQTT – MQ Telemetry Transport

ABSTRACT [English]

ABSTRACT [Polish]

ACRONYMS

CONTENTS

APPENDICES

GRAPHS

FIGURES

TABLES

1 INTRODUCTION

Application LUUTA (fin. broom) presented in this thesis is part of bigger system called BILINE. Idea of whole project is research which can give opportunity to improve job safety and managing of employees in the harbour area of the city of Kokkola. At project website we can find basic information about project - The BILINE project aims at building an ecosystem where the latest digital technologies, promoting the overall safety of identification technology, can be tested and developed (Centria 2017). Designers of project BILINE are planning to reach 3 main objectives:

Creating and developing platform that will improve security and provide authentication of data in industrial environments

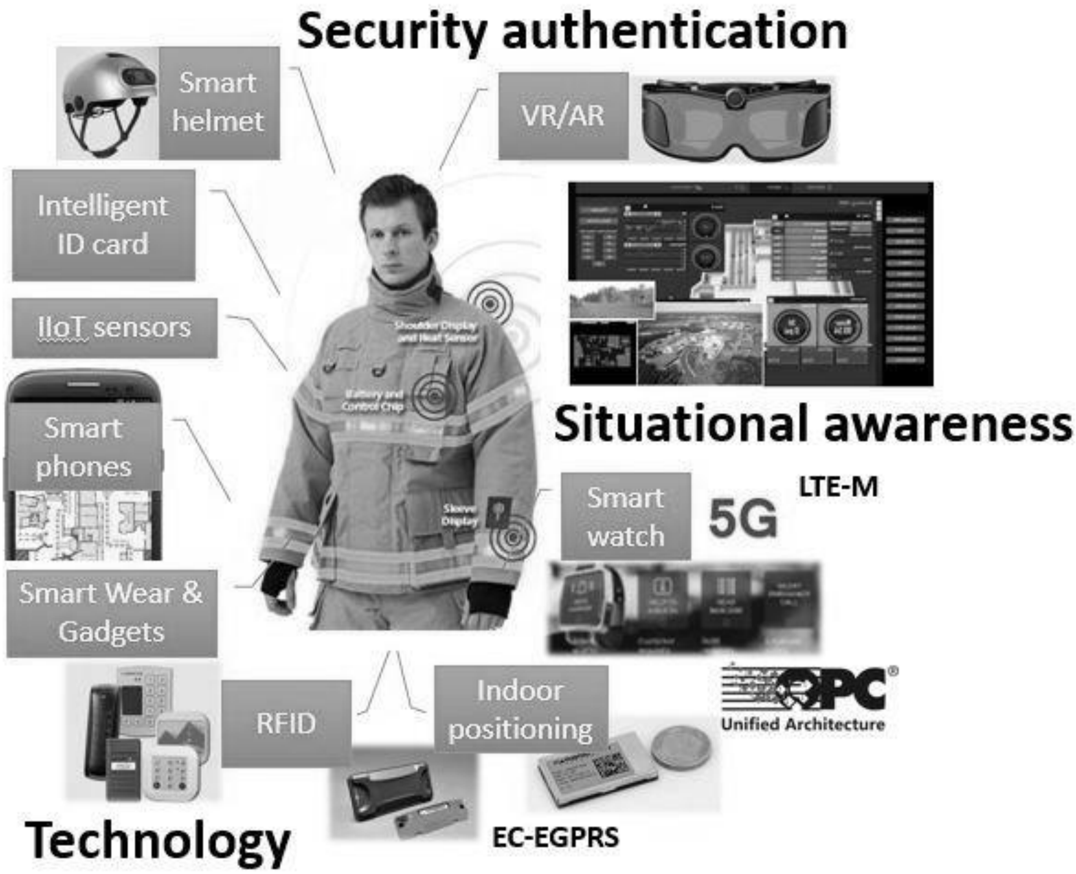
Experimenting, testing and building system which will give opportunity to point out importance of safety and help to find out solutions of problems.

Providing perspective of wider explanation of the industrial security technologies.

The research project is looking to answer the difficulties in the application of reference solutions in challenging application areas as well as to generate new knowledge, know-how and business models using the first pilot customer's feedback and user research (Centria 2017).

Application LUUTA will be subsystem making use of Augmented Reality in mobile phones technologies. Main purpose of application is to support employees working in Kokkola Industrial Park around harbour. Environment of that manufacturing area is characterized by large area, many of companies there, huge amount of people who are working there and heavy traffic during work times. LUUTA based on its technical solutions will help with managing of employees, improve way of informing about tasks and managing its execution, give possibility to track time of workers or provide data what specific worker is doing at certain period. Product is aimed to member of industrial area who are responsible for maintenance of roads during the winter. Work of that people is important on this area during the winter, incorrect and disorganized managing of their job may have an affect workflow of entire industrial estate. Entire system consists some subsystems like: Smart helmet, Intelligent ID card, IoT sensors, Smartphones applications, Smart wear & gears, Indoor positioning, Smart watches technologies, Virtual Reality/Augmented Reality.

LUUTA mobile application will be developed on commission of Centria University of Applied s Research and Development in Kokkola, Finland (APPENDIX 1).



GRAPH 1. BILINE system technologies (Centria 2017)

2 GENERAL IDEA OF AUGMENTED REALITY

This chapter explains what Augmented Reality is and shows general overview of that branch of Information Technology. It also covers types of Augmented Reality and how to distinguish it from Virtual Reality. At the end it shows how Augmented Reality can be used in mobile application development.

2.1 Overview

One of the best definition of augmented reality is that one created by Ronald Azuma in his thesis Making Augmented Reality a Reality:

Augmented Reality (AR) is an immersive experience that superimposes virtual 3D objects upon a user's direct view of the surrounding real environment, generating the illusion that those virtual objects exist in that space. While Virtual Reality (VR) completely replaces the user's view of the real world, AR supplements it [1]. In the long term, AR potentially has a much larger market than VR, because it improves the user's understanding of and interaction with the real world. AR connects users to the people, locations and objects around them, rather than cutting them off from the surrounding environment. AR is the most likely route by which wearable systems replace smartphones, because of its potential to provide a large visual display in a compact, head-worn form factor (Azuma 2017).

Ronald Azuma is comparing Augmented Reality with Virtual Reality. Placing Augmented Reality between Virtual Reality and Reality allows to get better understanding of problem. Whole purpose of Augmented Reality is to extend real world with virtual elements. It gives opportunity to improve people interacting with reality. In example people are cannot define their exact localization with longitude and latitude. With adequate technologies they can do it in few seconds.

Another definition of Augmented Reality from Handbook of Augmented Reality: We define Augmented Reality (AR) as a real-time direct or indirect view of a physical real-world environment that has been enhanced/augmented by adding virtual computer-generated information to it. AR is both interactive and registered in 3D as well as combines real and virtual objects (Carmiganini; Furth, 2011).

Important information emphasized in this definition is about processing 3D virtual elements in real-world in real-time. Affecting human senses real-time gives for user feeling that virtual elements are existing in real world. Moreover of 3-dimensional view enhance that feeling because it is natural way of seeing object for people. It allows to lose the limit between reality and virtual reality.

Place of Augmented reality was defined for the first time in publication Augmented Reality: A class of displays on the reality-virtual reality continuum. There we can find a diagram of Reality – Virtual Reality continuum (Paul; Haruo; Utsumi; Kishino,1994).

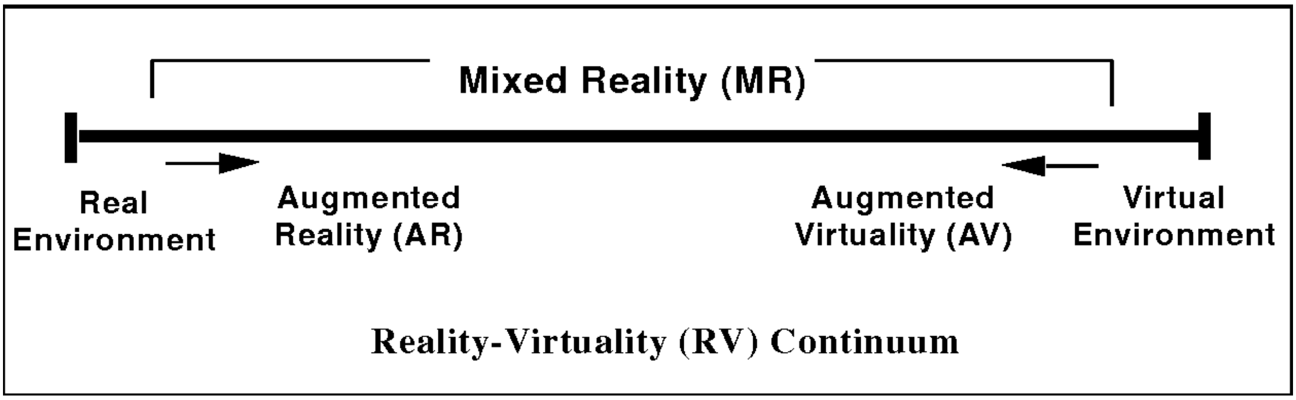


FIGURE 1. Reality – virtual reality continuum (Paul, et al. 1994)

Diagram above is showing concept of reality and virtual reality. It shows process of transmuting real environment to virtual environment and revers. As we see on diagram everything what between reality and virtual reality is Mixed Reality. Additionally, author divided mixed reality to Augmented Reality it means reality with virtual elements and Augmented Virtual Reality it means virtual reality with real elements.

2.2 Types of Augmented Reality

In Augmented Reality: A class of displays on the reality-virtual reality continuum publication authors divided Augmented reality systems to two basic subclasses:

“See through” AR displays

> This class of displays is characterized by the ability to see through the display medium directly to the world surrounding the observer, thereby achieving both the maximal possible extent of presence and the ultimate degree of "real space imaging".3 Most commonly display augmentation is achieved by using mirrors to superimpose computer generated graphics optically onto directly viewed real-world scenes (Paul, et al. 1994).

Monitor based AR displays

> We use the term monitor-based (non-immersive), or "window-on-the-world" (WoW), AR to refer to display systems where computer generated images are either analogically or digitally overlaid onto live or stored video images. Although the technology for achieving this has been well-

> known for some time, most notably by means of chroma-keying, many useful applications present themselves when this concept is implemented stereoscopically (Paul, et al. 1994).

Examples of first group can be AR googles, head up display, or any system fixed to sight organ. Second group is more connected to devices situated in some places or carried by human. Perspective of watching monitor-based AR displays can be different. In this situation allegory of eyeglasses (first subclass) and window (second subclass) can be used to compare how AR system is affecting user's way of seeing. In eyeglasses user almost always have same distance and angle between eye and glasses. In other hand window can be observed from different distance and on different angle and in changes our perspective of seeing elements "outside" (Paul, et al. 1994).

2.3 Augmented Reality in mobile technologies

Nowadays when smartphone became more popular than computers. They are more compact and flexible. They fit in our pockets and almost allows us to do same things like with normal computer. Moreover, producers are adding to mobile devices a lot of sensors like GPS sensor, compass, accelerometer or altimeter. It is great opportunity to provide for every smartphone user feeling of augmented reality. Augmented Reality in mobile phone technology is situated in second subclass of AR – Monitor based AR displays. Thanks to camera it can work as "window" to augmented reality world (Paul, et al. 1994).

Already there are two most popular mobile operating systems. Android provided by Google and iOS provided by Apple. Android had 87.7% and iOS had 12.1% of global mobile OS market share in sales to end user in second quarter in 2017 (statista.com 2018). To see how augmented reality is working mobile technologies we should notice that both companies in last years were developing their own frameworks for Augmented Reality. In case of Android it is called ARCore and in case of Apple – ARKit. What shows that Augmented Reality became important part of mobile technologies.

Success of mobile game Pokémon GO changed way of seen Augmented Reality in mobile technologies. It was first that big success of application using AR, only in Google Play number of downloads reach already (February 2018) about 10 million. It is important that it was also distributed in Apple Store (unknown number of downloads). After that Augmented reality gained huge popularity in mobile phone technology. Game allows user to move around cites and collecting Pokémon's and combating with another player on arenas. It was using basic processing of camera image, and additionally GPS localization to track person position. In my opinion this proved that is the best way of combination smart phones with Augmented Reality technology. Besides AR can improve storytelling in mobile games it can be also used in many more ordinary cases like education, marketing, tourism or interior design.

3 APPLICATION DESIGN

This chapter covers process of designing LUUTA. It shows functional and non-function requirements related to the application, which describe basic purposes and features of LUUTA. Rest of chapter provides information about how application works, using UML diagrams with descriptions.

3.1 External Interface Requirements

The analysis of requirements was done to describe basic features of application. The requirements for a system are the descriptions of what the system should do— the services that it provides and the constraints on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information. (Sommerville 2011)

3.1.1 User Interfaces

Application will be used as a vehicle terminal, it should be possible to use some basic functionalities during the driving. User Interface elements, that allow to interact with them should be sizable for easier handling. Spaces between clickable components should be also bigger than in normal applications to avoid miss clicks. Application should not distract user while it is driving. User Interface will be designed in dark and pale colours, because application will be used mostly in winter, when most of the day time is dark.

3.1.2 Hardware Interfaces

LUUTA using augmented reality needs to receive some data from device hardware components. Camera and GPS sensor. Application can run on device without GPS sensor or camera, but it will be not able to provide AR features. Additionally, application is using compass and accelerometer to provide better handling of displaying virtual objects in AR mode.

3.1.3 Software Interfaces

Application will work on Android platform. Above Android API 22 users can grant permission for "risk permissions" during application runtime. LUUTA need 2 risk permissions: camera and localizations. Application shall handle runtime permissions for these features.

3.1.4 Communication Protocols

Application will be communicated with external server. Server will be hosted on Azure platform. LUTTA shall use 2 protocols to communicate with Azure. MQTT for Azure IoT Hub communication and HTTP for Azure REST service.

3.2 Functional Requirements

FR001 Application shall be able to provide list of companies which can place an order.

FR002 Application shall be able to provide the list of tasks can be performed.

FR003 Application shall be able to send the database data on: task, commissioning company, location of user and speed of user.

FR004 Application shall be able to display current position of the user on map.

FR005 Application shall be able to display speed, temperature of the user in area and actual time.

FR006 Application shall be able to fetch information about nearest Points of Interest.

FR007 Application shall be able to display information about Point of Interest using camera.

3.3 Non-functional Requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. (Sommerville 2011)

3.3.1 Reliability

Application will be work based on network connection and GPS localization, it shall manage state of access and manage in situation of signal lacks.

3.3.2 Availability

Application will work with web server. Request time shall not be longer than 5 second. Application shall be able to send information about user when is working on foreground and background. Information shall be send every 30 seconds. GPS position displayed on map shall be updated every 50 meters, and at least every 60 seconds.

3.3.3 Security

Application shall be able to recognize user phone id.

3.3.4 Maintainability

Application should be divided in 4 modules based on basic views (dashboard, companies, map, camera). Is should also extract modules representing data models, networking, background services.

3.3.5 Portability

Application will be prepared for android platform and shall support all APIs above API 21 with their specific requirements.

3.3.6 Performance

Application shall not do too much work on main thread, that it will work without "screen freezes". APK file size shall not be bigger than 80 Megabytes.

3.3.7 Organizational requirements

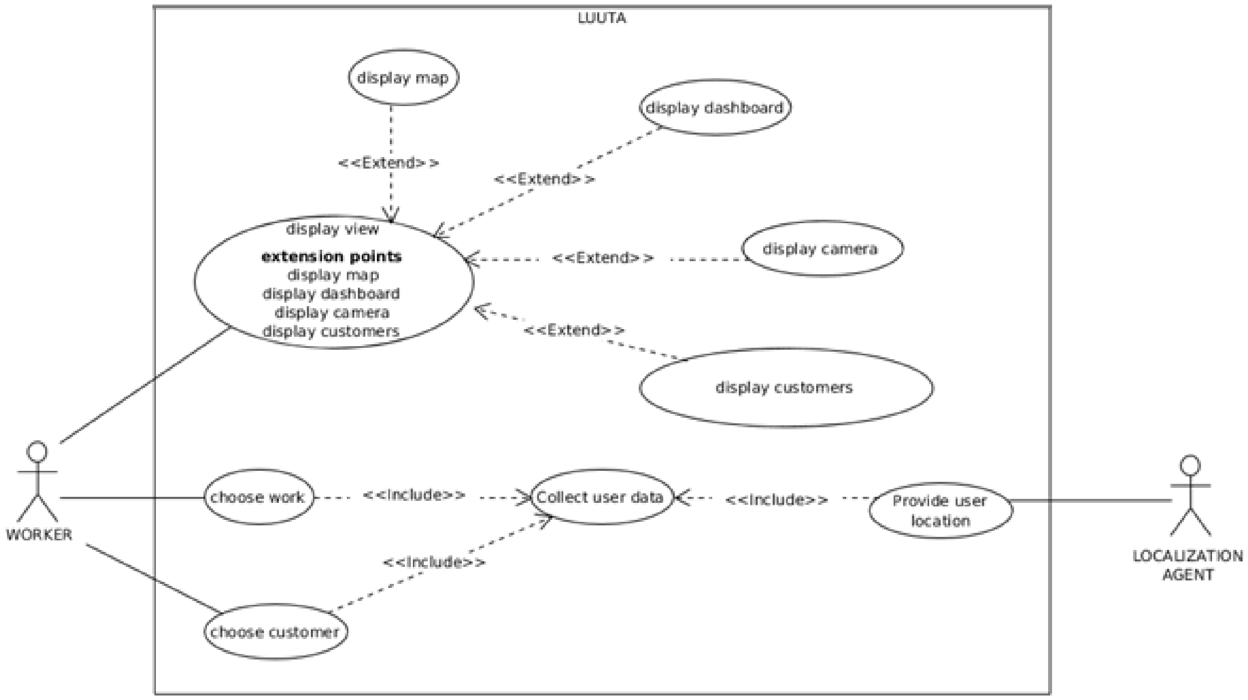
Development process shall be managed at Trello platform. Source codes shall be stored in repository on DEVEO platform.

3.4 Use Cases

The set of use cases represents all the possible interactions that will be described in the system requirements. Actors in the process, who may be human or other systems, are represented as stick figures.

Each class of interaction is represented as a named ellipse. Lines link the actors with the interaction. (Sommerville 2011)

3.4.1 Use Cases Diagrams



GRAPH 2. Use cases diagram

3.4.2 Use Cases Descriptions

Use Case ID:	**001**
Use Case Name:	**Provide user localization**
Actors:	Localization agent
Description:	Application is using GPS sensor to get information about longitude and latitude.
Trigger:	Run service which is listening for changes at GPS sensor
Pre-conditions:	User must allow for using localization services
Normal Flow:	GPS sensor is providing localization. Provided localization is set as user localization. After localization change event is raised
Alternative Flows:	
Exceptions:	

Use Case ID:	**001**
Use Case Name:	**Provide user localization**
Post-conditions:	Current localization is saved as a user position
Includes:	Display Points of Interest on Camera
Frequency of Use:	Every 30 seconds or 50 meters
Special Requirements:	User must allow for using localization services
Assumptions:	
Notes and Issues:	

Use Case ID:	**002**
Use Case Name:	**Display view**
Actors:	Worker
Description:	Application displays information about user current state or information which can help with its work.
Trigger:	
Pre-conditions:	
Normal Flow:	User selects a view from the list: display customers, display dashboard, display camera, display map
Alternative Flows:	
Exceptions:	
Post-conditions:	
Includes:	
Frequency of Use:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	**003**
Use Case Name:	**Choose customer**
Actors:	Worker
Description:	When data is fetched from server database, list of customers can be displayed
Trigger:	Fetch complete event
Pre-conditions:	List has been downloaded
Normal Flow:	Application displays list of customers. User can choose customer or no service
Alternative Flows:	
Exceptions:	
Post-conditions:	List of customers is displayed
Includes:	
Frequency of Use:	Every time when user starts list of customer view
Special Require-ments:	
Assumptions:	
Notes and Issues:	

Use Case ID:	**004**
Use Case Name:	**Choose work type**
Actors:	Worker
Description:	Application shall display 4 types of work. Snow plow, salt, gravel or other. User can check current executed work
Trigger:	Triggered by user
Pre-conditions:	
Normal Flow:	User is checking toggle button with name of work, applications links current executed work to user data
Alternative Flows:	
Exceptions:	
Post-conditions:	Data is saved and can be sent to server.

Use Case ID:	**004**
Use Case Name:	**Choose work type**
Includes:	
Frequency of Use:	Depends of user
Special Require-ments:	
Assumptions:	
Notes and Issues:	

Use Case ID:	**005**
Use Case Name:	**Collect data**
Actors:	Worker
Description:	Data about user is collection in one package of information. It contains location, speed, heading, task type, company id, user id
Trigger:	Sending service event
Pre-conditions:	
Normal Flow:	Application collects data about user and preparing it to send to Azure Hub
Alternative Flows:	
Exceptions:	
Post-conditions:	Data is prepared to send
Includes:	
Frequency of Use:	Every 10 seconds when user speed is greater than 0 and 60 seconds when user speed is equal zero
Special Require-ments:	
Assumptions:	
Notes and Issues:	

3.5 Dictionary

Based on created use cases is possible to process class selection. Every use case provides crucial information about actions and its executors. Normal flow gives important information about classes and its methods needed in application.

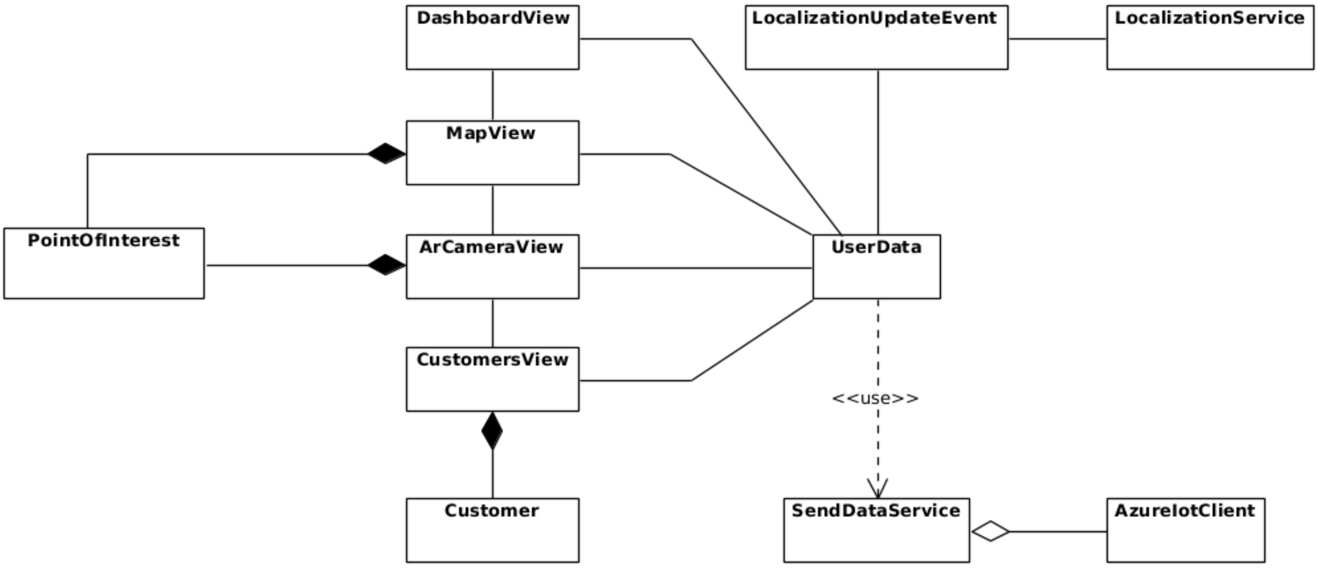
TABLE 1. Lexical analysis of use cases.

Use case number	Use case name	Use Case Description
001	Provide user localization	GPS sensor is providing localization. Provided localization is set as user localization. After localization change event is raised.
002	Display view	User chooses to view and can display: dashboard (displayed by default) with speed, time and weather information, list of customers, map with points of interest or camera view with augmented reality.
003	Choose customer	Application displays list of customers. User can choose customer or no service.
004	Choose work type	User is checking toggle button with name of work, applications links current executed work to user data.
005	Collect data	Application collects user's data and preparing it to send to Azure IoT Hub

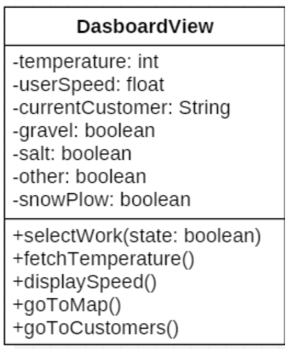
Lexical analysis of use cases results with the following nouns: Weather, Localization, User Localization, Event, Dashboard, Customer, Camera View, Map, Points of Interest, Customer, User's Data, Toggle Button.

Analyse of observed nouns, gave information what classes and methods will be used in application. Application will be divided for 4 view classes which will represent: dashboard, companies list, map and augmented view. Additionally, application will contain classes for handling sending statistic and updating data from GPS sensor. They will be represented as Android service class. All updates of localization will be sent by events. First view(dashboard) class will contain class which can help with displaying information about user state. Moreover, view will use HTTP request and response to fetch data about weather form Weather API. Next view (companies list) will contain widgets which can display list of customers. On this view will be sent HTTP request to fetch customers. Third view will display map with user position on it and all points of interest in area. Last view will be camera. Camera view will have implemented Augmented World and will be able to display icons which will symbolize Points of Interest. In application will be used one extra component which will be responsible for communication with Azure IoT Hub. Communication with Hub allows to sending statistic about user to server. It will be represented by Azure Client.

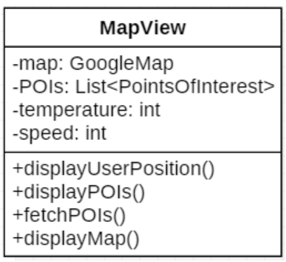
3.6 Class Diagram



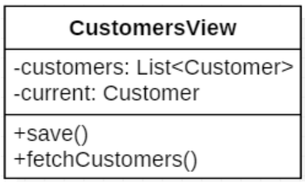
GRAPH 3. Class diagram



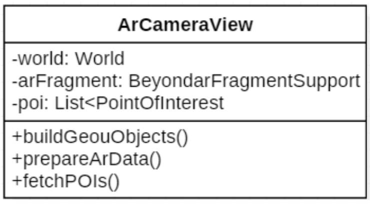
This class represents dashboard view. It displays current temperature, speed of user, and allows to user to define what type of work is doing. There are 4 types: gravel, salt, snow plow, other. It means what type of stuff is used for road maintaining.



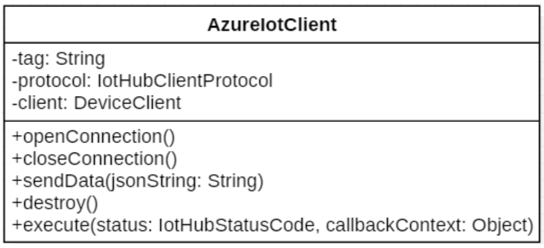
This class represents Map view, which is displaying map with all points of interest and user position. On this view also, temperature and speed is shown.



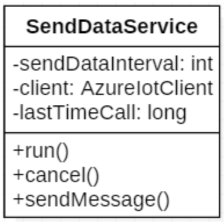
This class is showing list of customers for user and no service option. User can pick one of the options and then save choice.



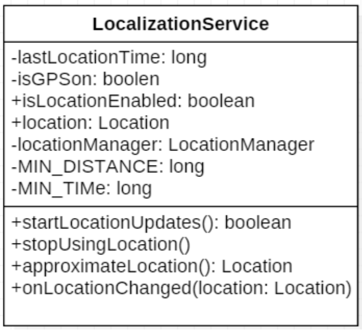
This class represents Camera view with implemented Augmented Reality world. It displays virtual elements based on user localization an



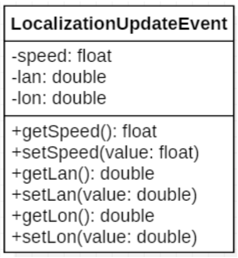
This class is responsible for sending data to Azure IoT Hub. It is starting connection, sending data and closing connection.



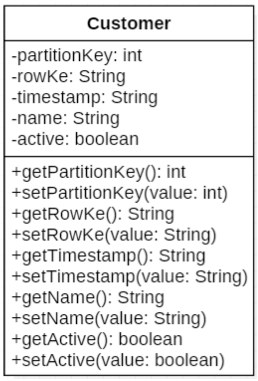
This class is scheduling sending data. It contains instance of Azure Client which is handling it.



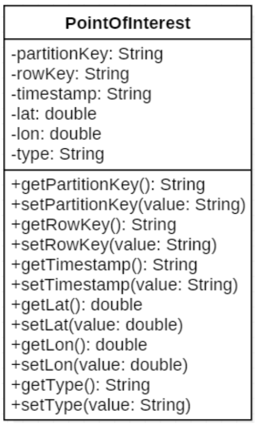
This service is responsible for reading current information about localization from sensors. It has specified minimum distance and time for update. Every time when localization change is detected event is rising.



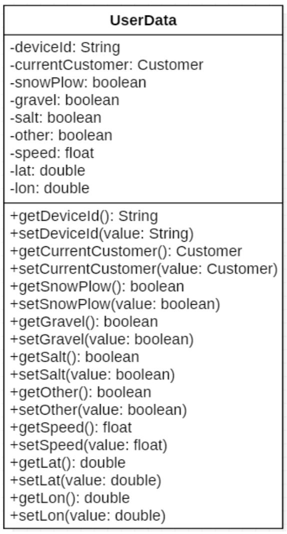
This class representing update of localization, using it allows to send information about localization changes to across application.



This class is model of Customer. It contains all information need to send it to Azure server.



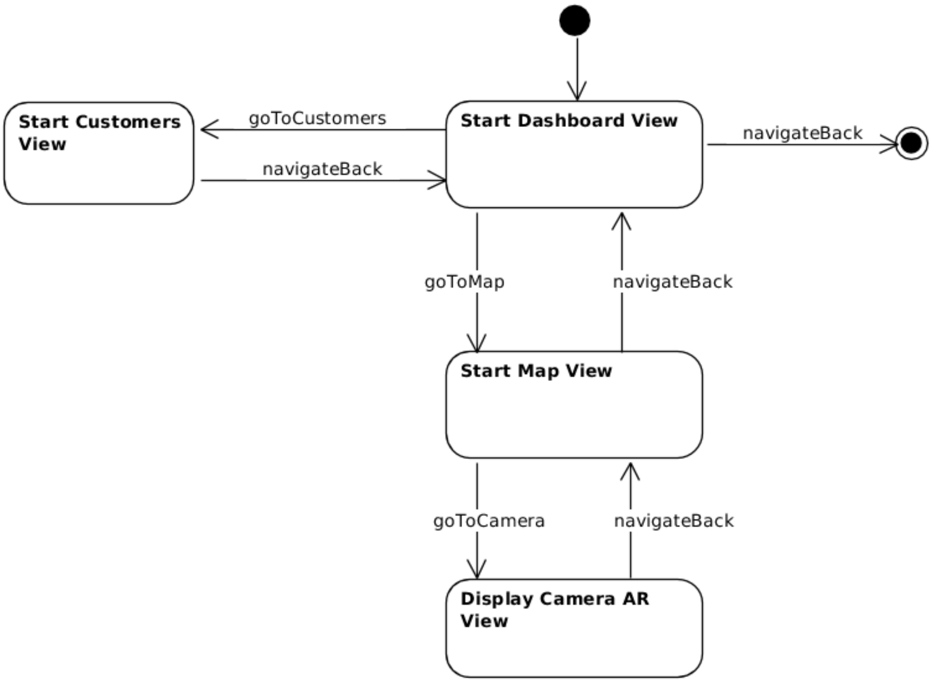
This class is model for Point of Interest. It contains all information need to display it on map or AR camera.



UserData is storing all important information about current state of user.

3.7 Activity Diagram

LUUTA will be working with 4 views. Worker can change between views. Every view provides information for worker. Below activity diagram shows flow of navigation in application.



GRAPH 4. Activity Diagram

3.8 State diagrams

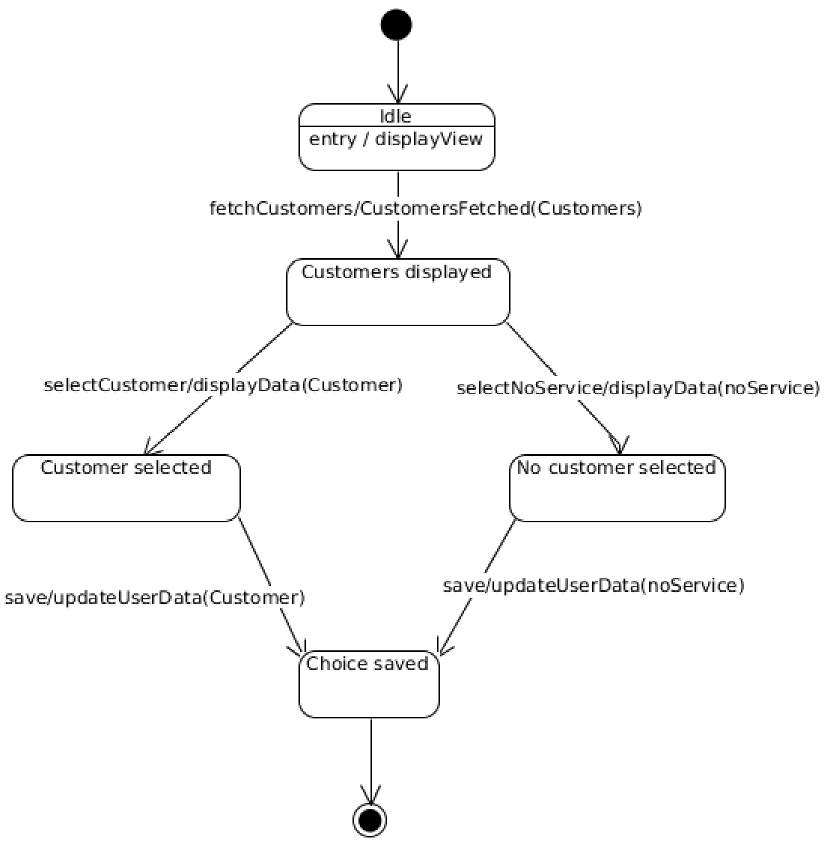
Excluding navigation which are independent from states of views, all views are working with several states. Diagrams below describes state flow of every individual view.

3.8.1 Dashboard

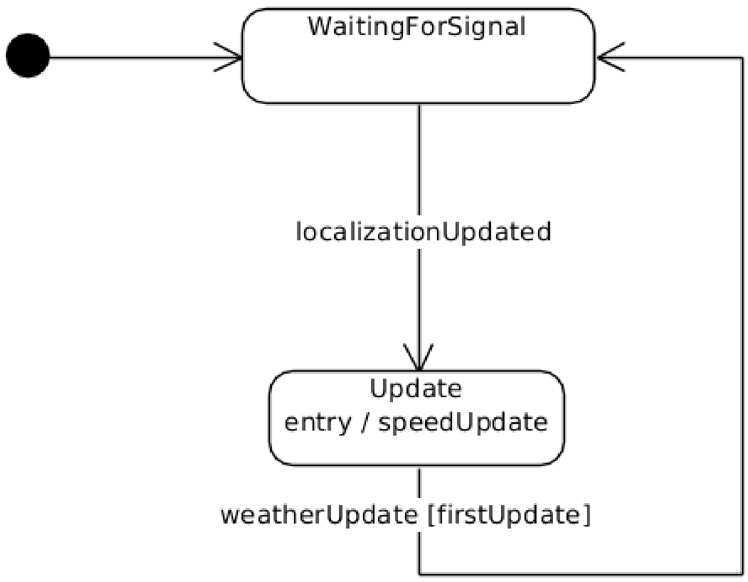
First view will 4 states. (GRAPH 4.) At beginning idle state, and it will be wait for signal from GPS sensor. When localization is updated, then 2 events will happen. Updating speed information on screen and requesting data from weather API.

3.8.2 Customers

Customers list view provide 4 states, but single flow of states contains 3 states. At beginning view is prepared. Then application sends request for Customers data. After customers list is displayed, user can choose one of company or no service. After Choice user is saving data.



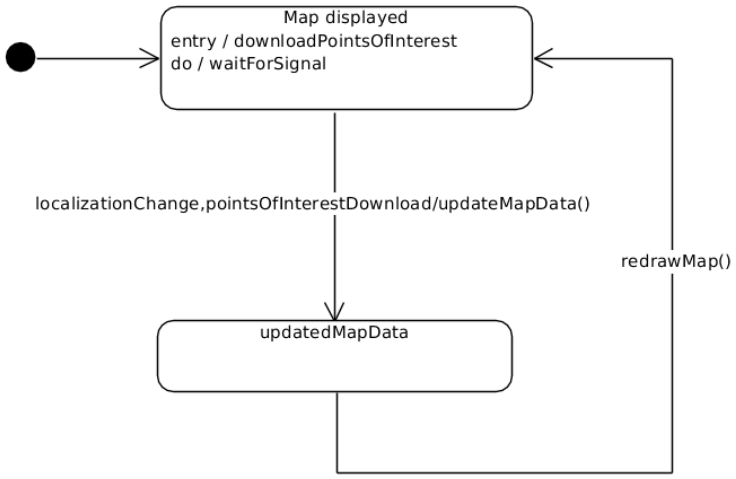
GRAPH 5. States of the dashboard view



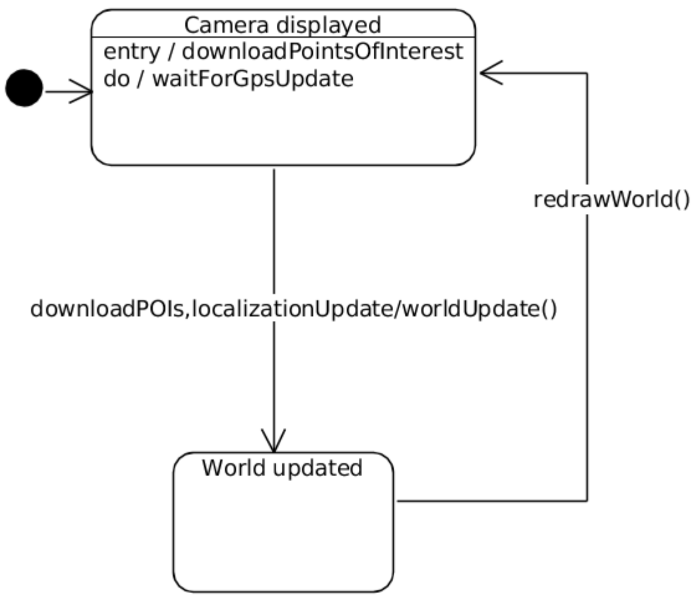
GRAPH 6. States of the customers selection view

3.8.3 Map

Map view contains 3 states. At start map is setting up. When map is displayed, view is waiting for localization event from GPS sensor, when it is raised localization of user is displayed on map. Also, after map set up, view is requesting data about Points of interest from server. Data contains localization fields about POIs, what allows map to display point on it.



GRAPH 7. States of map interactions



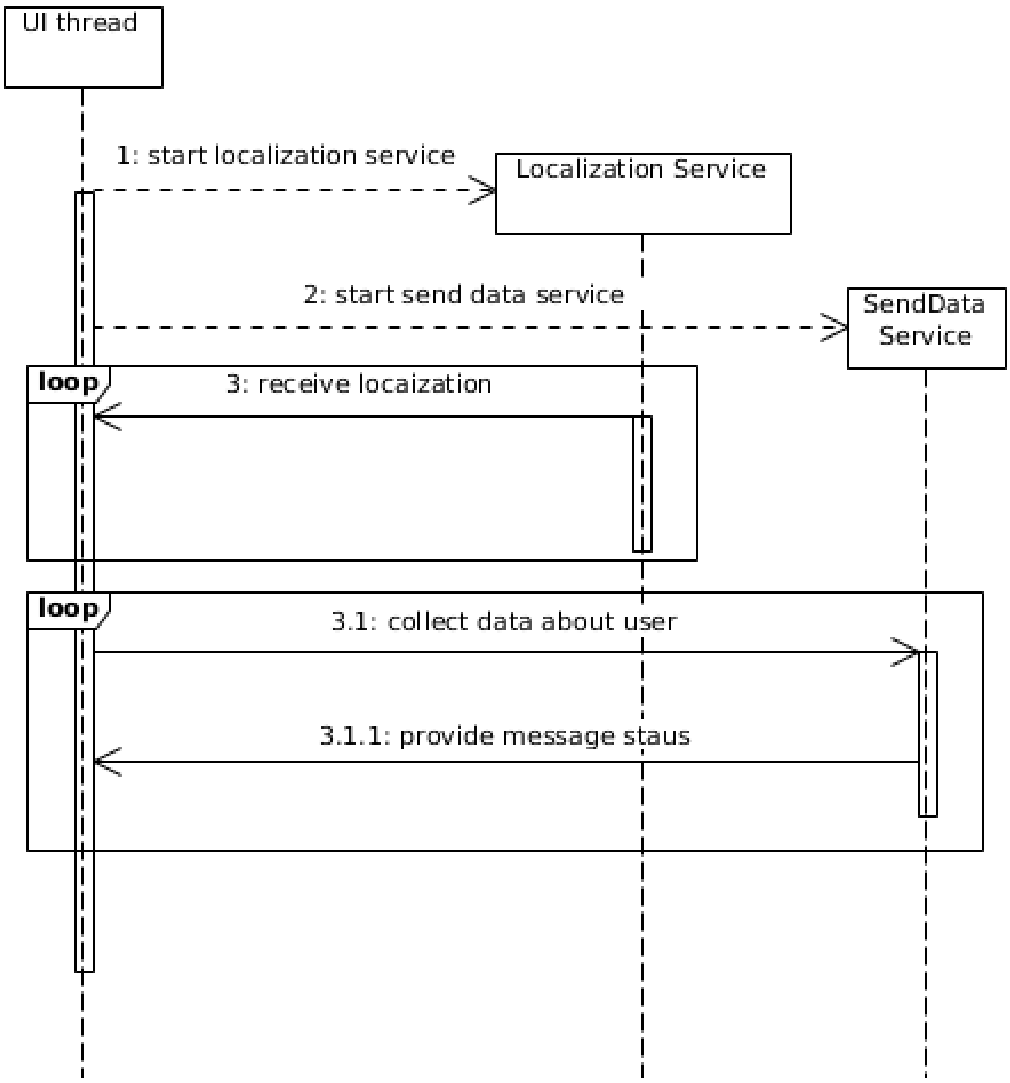
GRAPH 8. States of the camera interactions

3.8.4 Augmented Reality camera view

Augmented Reality camera view has similar flow to map states flow. It contains 3 states. But most important difference is that effects of fetching Points of interest and localization event merged. Especially every localization update affects Augmented Reality objects redraw.

3.8.5 Services in application

LUUTA is doing some tasks in background which a repeatable. First of it is updating localization information of user. Second one is sending user data to Azure database about user. This operation is executed during application lifetime.



GRAPH 9. Services in LUUTA

3.9 User interface design

During design process client provided graphical representation of 3 views. On all views are dominating dark colours. Graphical User Interface is designed in that way to assure comfort of driving for worker. Moreover, all view contains only "one click" components for easier usage during driving.

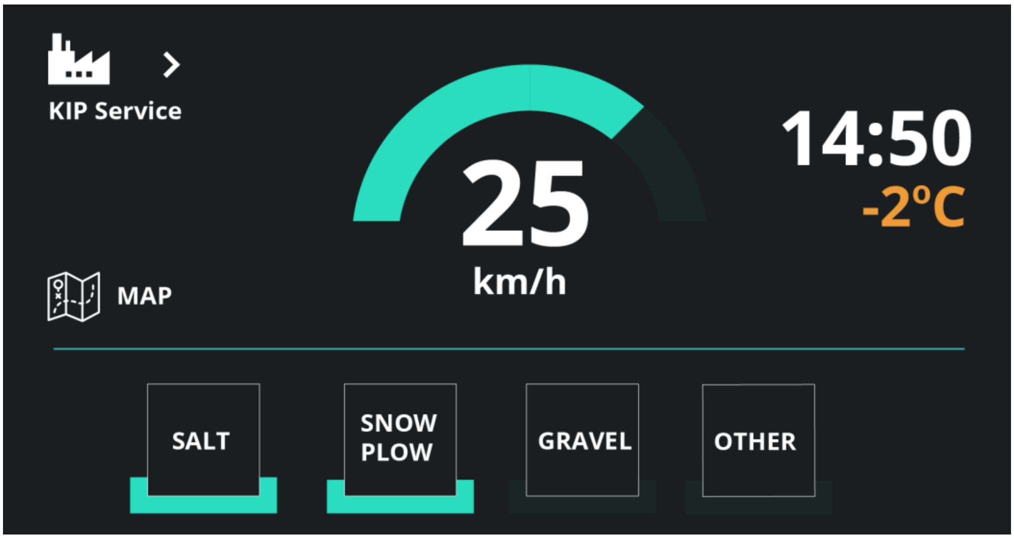


FIGURE 2. Mock-up of dashboard

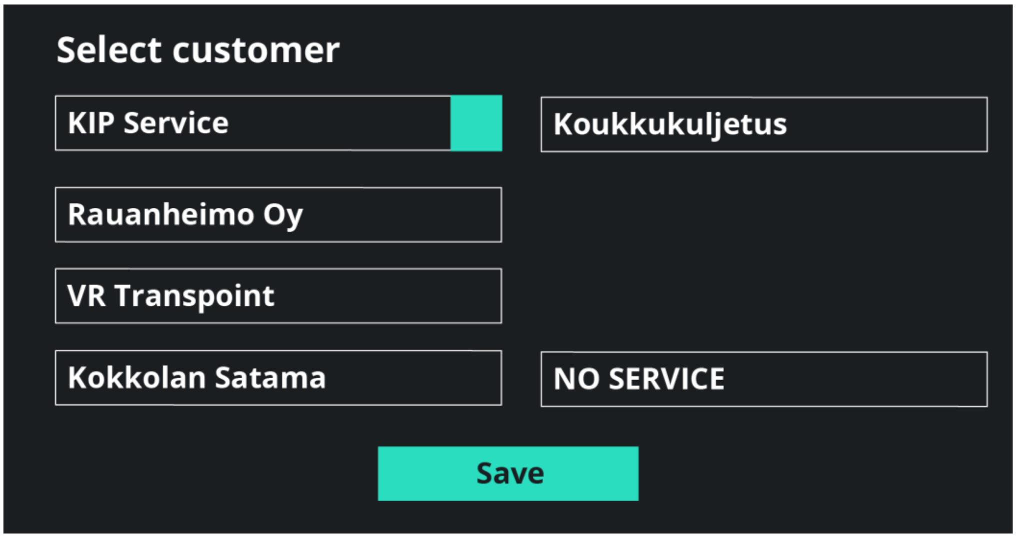


FIGURE 3. Mock-up of customers selection

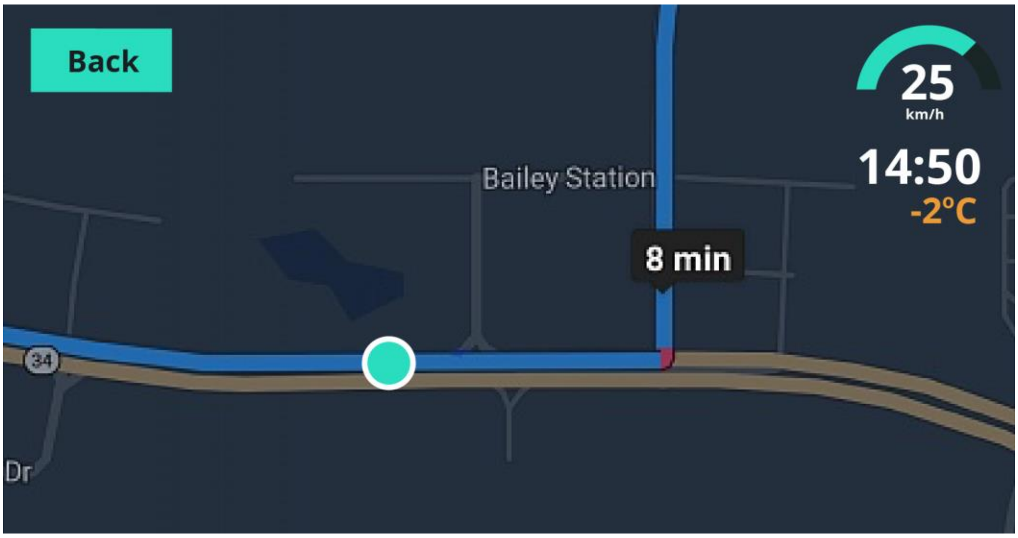


FIGURE 4. Mock-up of map

4 APPLICATION DEVELOPMENT

This part covers some wider description of Android application process which contains: briefly review of used tools, external libraries descriptions, design patters used in applications and final class diagram showing all components of application.

4.1 Tools

Developers like other specialists need some tools for work. In some cases, it is possible to reach something without tools, but in most cases, it is extremally hard or impossible. This subchapter describes all necessary tools for creation LUUTA application.

4.1.1 Android Studio – Integrated Developer Environment

Official Integrated Developer Environment for developing Android application. Application was developed using Android Studio version 2.4. All newer releases should be backward compatible and support LUUTA project.

4.1.2 GitKraken – Client Control Version

During process of development was using git version control system. GitKraken is Graphical User Interface git client which is compatible with Linux Ubuntu. It is very convenient and intuitive git client.

4.1.3 Deveo – Platform for Control Versioning

Finnish repository management platform. Repository of application is stored on Deveo server. Repository was provided by client.

4.2 External Libraries

All libraries in project are managed by Gradle tool. On gradle project site we can find wider information about project:

Gradle is an open-source build automation tool focused on flexibility and performance. Gradle build scripts are written using a Groovy or Kotlin DSL.

Gradle supports many major IDEs, including Android Studio, Eclipse, IntelliJ IDEA, Visual Studio 2017, and XCode. You can also invoke Gradle via its command line interface in your terminal or through your continuous integration server. Gradle build scans help you understand build results, improve build performance, and collaborate to fix problems faster. (gradle.org 2018)

4.2.1 GreenRobot EventBus

Project address: https://github.com/greenrobot/EventBus

Library provides Java implementation on project pattern of Event Bus. Allows to use event bus in very convenient and easy way in Android Application. Basic purpose of using event bus in android development is to make easier communication between independent components. Basic advantages mentioned by developers about this library are that it implifies the communication between components, decouples event senders and receivers, performs well with Activities, Fragments, and background threads, avoids complex and error-prone dependencies and life cycle issues, makes your code simpler, is fast, is tiny (~50k jar), is proven in practice by apps with 100,000,000+ installs, has advanced features like delivery threads, subscriber priorities, etc. (github.com/greenrobot 2018)

4.2.2 android-location-tracker

Android Simple Location Tracker is an Android library that helps you get user location with an object named LocationTracker. (github.com/quentin7b 2018)

This library helps with handling GPS sensor work. It is very convenient tool and it is improving readability of code. Good managing of Location tracking in Android can be hard, but this library simplifies development to maximum.

4.2.3 Retrofit

Retrofit is the most popular networking framework for Android. It is created and maintained by Square. In my application I use to handle working with REST service interface by HTTP protocol. All data was exchanging using JSON format. Retrofit is also compatible with GSON library which allows to converse JSON string to Java Objects.

4.2.4 Beyondar

This framework has been designed to offer some resources to those developers with an interest in working with Augmented Reality based on geolocalization on Smart Phones and tablets (github.com/BeyondAR 2018). In application is used module for displaying virtual elements in camera view.

4.2.5 IoT device Client

The Microsoft Azure IoT device SDK for Java facilitates building devices and applications that connect and are managed by Azure IoT Suite services (Microsoft 2018). This library is necessary to connect to Azure IoT Hub and sending data about user to it.

4.3 Design patterns

Design pattern can be understood as a universal solution related to specific problem. It is true that probably design pattern is not only one solution for specific problem, but often it is recommended for solving it. Mostly design patterns allow to solve programming problems in efficient and clear way.

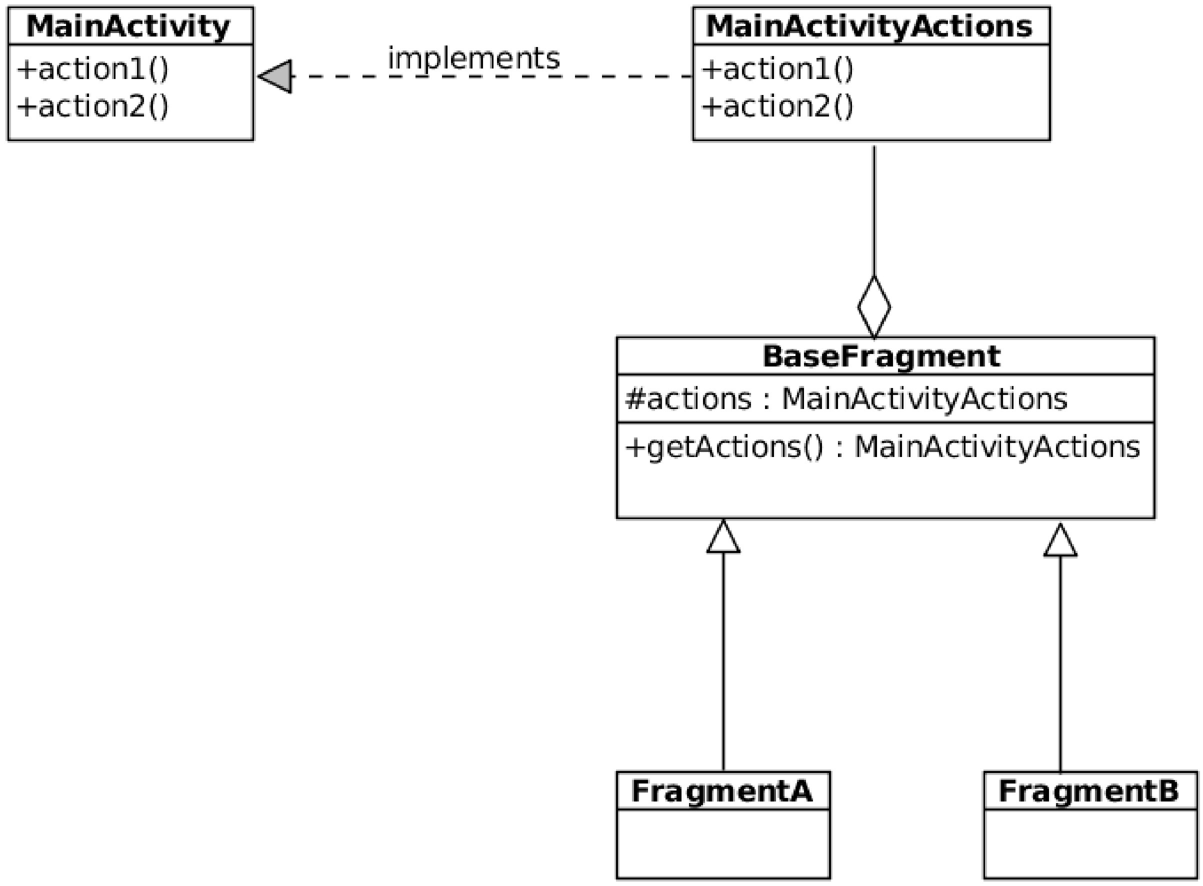
4.3.1 Fragment communication pattern

It is popular pattern using to handling communication between Fragments and Activities in Android application. Often you will want one Fragment to communicate with another, for example to change the content based on a user event. All Fragment-to-Fragment communication is done through the associated Activity (see GRAPH 10). Two Fragments should never communicate directly (developer.android.com 2018).

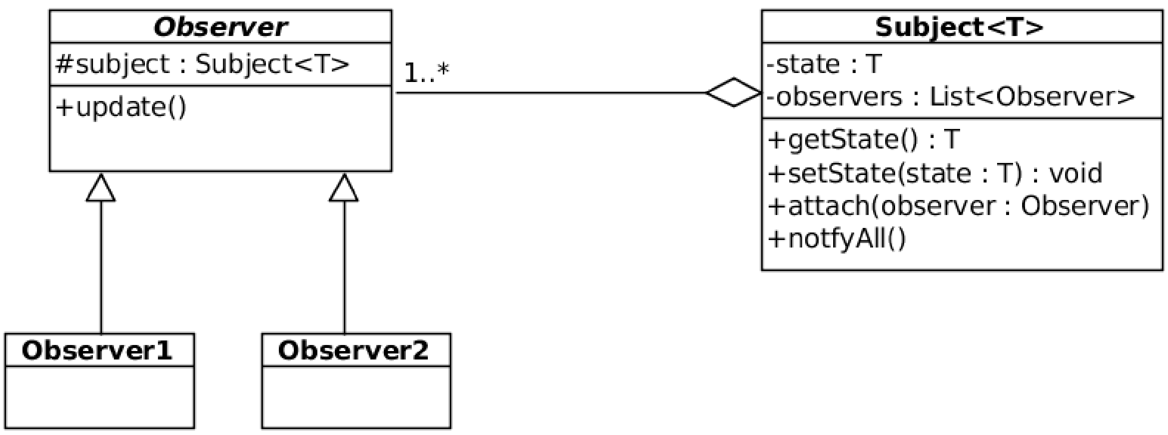
4.4 Observer pattern

Android application are characterized by event-based flow. Most of user interactions causes some reaction. Very common and convenient pattern used in Android development is observer pattern and its variations.

Consider, for example, a case where data is kept in a database and can be displayed in multiple formats, as a table or a graph. The Observer pattern (see GRAPH 11) suggests that the display classes register themselves with the class responsible for maintaining the data, so they can be notified when the data changes, and so they can update their displays (IBM 2018).



GRAPH 10. Class diagram - fragment communication pattern



GRAPH 11. Class diagram - observer pattern

5 TESTS

Test of application was divided for 5 parts. During test all views were tested and sending data module. Tests were driven in end-to-end tests. End-to-end testing is a methodology used to test whether the flow of an application is performing as designed from start to finish. The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various system components and systems (techopedia.com 2018).

TABLE 2. Test procedure of the dashboard.

User path	Expected behaviour	Test Result	Comment
1. Start view **2. Accept localization permission** **3. Wait for temperature information**	Application displays current temperature	Passed	
1. Start view **2. Increase speed** **3. Wait for speed information**	Application displays information about current speed	Passed	
1. Start view **2. Toggle work button** **3. Remember user choice**	Application remembers user choice. Application is sending data to server	Passed	

TABLE 3. Test of customer selection

User path	Expected behaviour	Test Result	Comment
1. Start customer view **2. Wait for customer list**	Customers list is fetched from server. 4 companies are displayed.	passed	No service item is marked by default. In other cases, last chosen company
1. Start customer view	Customer item is marked on list.	passed	

User path	Expected behaviour	Test Result	Comment
2. Wait for customer list **3. Select customer** **4. Save choice**	Saved choice is send in data package to server.		
1. Start customer view **2. Wait for customer list** **3. Select no service** **4. Save choice**	No service item is marked on list. Saved choice is send in data package to server.	passed	

TABLE 4. Test procedure of the interaction with a map.

User path	Expected behavior	Test Result	Comment
1. Start map view **2. Wait for GPS signal**	Position of device is displayed on map. Position is represented by blue dot.	passed	
1. Start map view **2. Wait for GPS signal** **3. Change localization**	Position of device is displayed on map. Every change of position is updated on map.	passed	Localization should be updated every 30 seconds or 50 meters
1. Start map view **2. Wait for GPS signal** **3. Click on POI pin**	Position of device is displayed on map. Points of interest are displayed. Points are represented by map pins	passed	

TABLE 5. Test procedure of the interaction with camera.

User path	Expected behaviour	Test Result	Comment
1. Start camera view	Camera view is displayed. Correct orientation and right proportion of screen.	passed	

User path	Expected behaviour	Test Result	Comment
1. Start camera **2. Wait for localization data**	Point of interest symbol is displayed on map	passed	
1. Start camera **2. Wait for localization data** **3. Change device localization** **4. Change localization position**	Camera view is displaying point of interesting symbol. Position of symbol is changing relatively to device position	passed	Quality of mapping position on screen depends on device sensors. Observed not stable position of virtual objects.

5.1 Release notes

There are no issues affected work flow of application, which should be included to release notes.

6 MANUAL

This chapter provides all important information for user of LUUTA application. It covers application's requirements, instruction of installation and user manual which describes way of using application.

6.1 Application's requirements

Application was created to run on devices powered by operating system Android, at least version 5.0 Lollipop API 21. Also, device should have turned on GPS provided, and it have active internet connection during application use. To ensure that application will be working properly device should fulfil following hardware requirements:

1. Camera
2. GPS sensor
3. Accelerometer
4. Compass

6.2 Installation manual

Application is not distributed to Google Play Store, so installation is only possible using apk file. Before beginning installation, process make sure that device fulfil all requirements above:

1. Allow for installing applications from unknown sources. Follow steps below:
    a. Open Settings
    b. Tap Security section
    c. Tap Personal section
    d. Check unknown sources box
    e. Tap OK on information dialog
2. Download .apk file
3. Click install on installation dialog

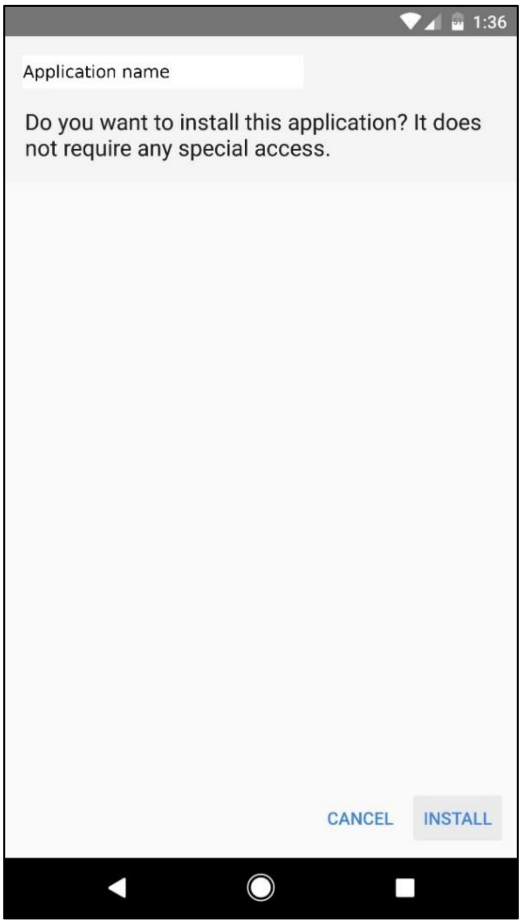


FIGURE 5. Installation dialog

6.3 User manual

User manual contains description of all interactive and important components at every view. Components are marked by red dots with numbers. In the text related to specific view, information about specific component is marked by number from view screen.

6.3.1 Dashboard

To choose commissioning company tap to icon in left top corner (1). Then application will display list of companies. On the centre of screen is displayed speed information (2). Data is displayed in kilometres per hour. On right side is displayed common information about current time (3) and current temperature (4). On bottom screen are displayed current tasks (5) as a toggle button. All buttons have 2 states checked and unchecked. If bottom of toggle button has aquamarine colour, it means that current work is preforming. To navigate to map view tap on map icon on left side of screen.

FIGURE 6. Dashboard of the application.

6.3.2 Customer's selection

On right side is displaying list of customers (1). User can pick one of company from list or no service (3). By default, option no service is marked. When company is chosen user can save (2) its choice.

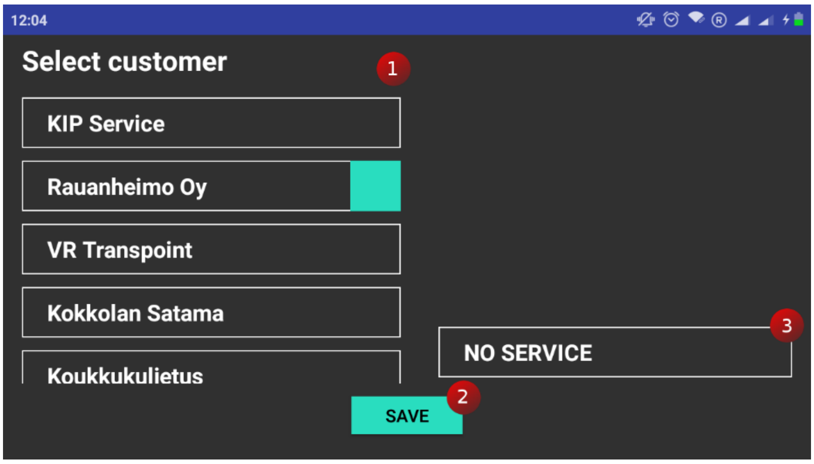


FIGURE 7. Customer selection screen.

6.3.3 The use of maps

On top right are placed navigation buttons. To go back to dashboard, tap on back button (1). To navigate to Augmented Reality camera (2). Every point of interest is displayed as a pin on map (3).

On right top is placed container (4) with current information like speed time, and temperature. User position is represented as aquamarine dot on map (5).

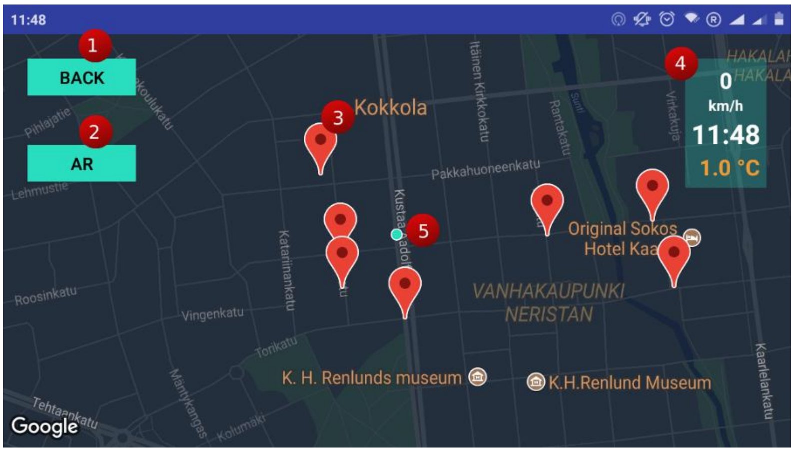


FIGURE 8. Map view

REFERENCES

Azuma, Ronald T. 2017. “Making Augmented Reality a Reality.” Intel Labs. 20 12. http://ronaldazuma.com/papers/OSA2017_invited_paper_Azuma.pdf.

Carmiganini, Julie, and Borko Furht. 2011. *Augmented Reality: An Overview.*

Centria, TKI. 2017. *tki.centria.fi.* 12 11. https://tki.centria.fi/hanke/biline-turvallisuuteen-liittyvat-digitaaliset-ratkaisut/6248/6248/6248.

developer.android.com. 2018. 13 02. https://developer.android.com/training/basics/fragments/communicating.html#DefineInterface.

github.com/BeyondAR. 2018. 03 03. https://github.com/BeyondAR/beyondar.

github.com/greenrobot. 2018. 03 03. (https://github.com/greenrobot/EventBus).

github.com/quentin7b. 2018. 03 03. https://github.com/quentin7b/android-location-tracker.

gradle.org. 2018. 03 04. https://docs.gradle.org/4.5.1/userguide/userguide.html?_ga=2.81753303.1558757124.1519677497-776725113.1519233690.

IBM. 2018. 03 03. https://www.ibm.com/developerworks/java/tutorials/j-patterns/j-patterns.html.

Microsoft. 2018. 30 03. https://github.com/Azure/azure-iot-sdk-java/tree/master/device.

Paul, Milgram, Takemura Haruo, Akira Utsumi, and Fumio Kishino. 1994. *Augmented Reality: A class of displays on the reality-virtuality continuum.* Japan: ATR Communication Systems Research Laboratori.

Sommerville, Ian. 2011. *Software engineering.*

statista.com. 2018. *statista.* 03 04. https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/.

techopedia.com. 2018. 03 03. https://www.techopedia.com/definition/7035/end-to-end-test.

APPENDIX 1

Centria
AMMATTIKORKEAKOULU

THESIS CONTRACT

Author(s) of the thesis Bartosz Wieczorek	Date of starting the thesis project 19.9.2017
Degree programme	COU department
Principal lecturer of the degree programme	Proposed supervisor

Initial subject of the thesis
In this thesis Bartosz will be implementing a smart driver's device for heavy vehicles. This device will be in charge of proving data about task driver is doing and which client driver is working with. Device will also have augmented reality feature where user is able to see important information and map.

Research problem/assigned development task
- Develop reliable application (Android) that can send data accurately and often to Azure server
- Augmented reality feature to be used in mobile device to provide information driver

Objective of the thesis and delimitation/expected research outcome
- Usable Android application for heavy vehicle driver with augmented reality

Centria
AMMATTIKORKEAKOULU

THESIS CONTRACT

Initial schedule for thesis project (dates in months)

Start-up meeting (supervisor, thesis author, working-life supervisor)	[September 2017]
Presentation of implementation plan	[______________]
Interim report	[______________]
Review of thesis by supervisor and/or final meeting	[______________]
Submission of thesis	[______________]
Seminar presentation of thesis	[______________]
Maturity test	[______________]

Contact information of commissioner

(company, name of contact person, address, telephone, email)

Centria University of Applied Sciences, Jari Isohanni, RDI-coordinator, Talonpojankatu 2 67100 Kokkola FINLAND, +358 40 669 0690, jari.isohanni@centria.fi

This contract has been issued in three copies. The copies of the contract shall be delivered to the author, the commissioner, and the supervisor. The student of Central Ostrobothnia University of Applied Sciences, following the commissioner's assignment, commits to complete a thesis on the subject above by ____________ (date).

The commissioner commits to make necessary information available for the author of the thesis, and, after the thesis is completed, to assess the usability and pragmatic value of the thesis. The commissioner shall be responsible for paying for material, postage, travel, and other costs according to invoice. The commissioner may pay a compensation to the author of the thesis.

The undersigned shall be responsible for not disclosing any information gained in connection with the commission in so far as this information can be regarded as a business secret of the contracting party. The thesis shall be reviewed at a thesis seminar and it shall be made a public document. The commissioner shall request for non-disclosure of the thesis separately.

The University of Applied Sciences shall not be responsible for any harm or damage caused by the author of the thesis.

Date	
Signature of working-life supervisor	[signature]
Signature of student	[signature]
Signature of thesis supervisor	[signature]