FINDIT
- Design and Implementation of a Mobile Application for Public Transport

Mille Boström

Master's Thesis

Department of Design Sciences
Lund University
ISRN:LUTMDN/TMAT-5166-SE

EAT 2013

Video available at:

http://www.youtube.com/watch?v=RPtyZSgUWnw

Abstract

The purpose of this master's thesis was to examine existing augmented reality technologies and develop an application for Windows phone. The goal with the application was to find new ways to use augmented reality technologies together with public transport. The end user would use the application to search for and find new journey options.

The development process consisted of four steps – investigation, design, implementation and testing. The Investigation was done by looking at current public transport and augmented reality applications for smartphones, compare them and see which functionality may be used or improved.

Design consisted of scenarios, low-fi prototyping and storyboards where the author followed Microsoft own guidelines for designing and developing a Windows phone application. The application was implemented for both Windows phone 7 and 8 and a comparison was made between the two systems. The last step was testing and it was done by exploration and performance testing.

The result is an application and a video. It became clear that augmented reality can be used and the technology for it is here, but the application still have a long way to go with testing and further consultations from users.

Sammanfattning

Syftet med detta examenensarbete var att undersöka nuvarande augmented reality tekniker och utveckla en applikation till Windows phone. Målet med applikationen var att hitta nya sätt att använda augmented reality på tillsammans med kollektivtrafiken. Slutanvändaren skall sedan använda applikation för att söka och hitta nya resealternativ.

Utvecklingsprocessen bestod av fyra steg – undersökning, design, implementation och testning. Undersökningen utfördes genom att titta på nuvarande applikationer för kollektivtrafiken och augmented reality till smartphones, jämföra dem och se vilken funktionalitet som kan användas eller förbättras.

Designen bestod av scenarios, low-fi prototyper och storyboards där författaren följer Microsofts riktlinjer för design och utveckling av en Windows phone applikation. Applikation implementeras sedan för Windows phone 7 och 8 där en jämförelse mellan båda systemen utförs. Till sist utförs testning som bestod av utforskande testning och prestandatest.

Resultatet är en applikation och en video. Under arbetets gång blev det tydligt att augmented reality kan användas och teknologin finns för det, men att applikationen har en lång väg kvar att gå och att det behövs mer testning och rådslagning med användare.

Acknowledgments

I would like to thank my supervisor at the design institution at LTH, Joakim Eriksson, for his support and guidance with my application and report. I would also like to thank Kristofer Axelsson, Pontus Johansson and Isa Boström for helping me review the report.

Table of Contents

1.0 INTRODUCTION

1.0 Introduction

Augmented reality is a way for the user to combine the real and virtual and to interact with it in real time. It is also a technology that has not found its right place yet and is found more in nice looking demos than in practical situations. Is it because the technology is not ready? Is it because there is no use for it? These were some of the question I asked myself. It is also the start point of my master thesis.

I wanted to explore the possibilities of using augmented reality in our daily life. But instead of inventing something new, I decide to look at things we already do, and how we could do them in a different way with augmented reality. That is why I decided to explore smartphones and public transport. In Sweden a lot of people have a smartphone and one in five use the public transport every day [24]. This make it to a perfect combination to test new ways to use augmented reality.

1.1 AUGMENTED REALITY

Augmented reality (AR) is a technology that allow real-time-fusion of computer generated digital content with the real world [19]. Azumas's (1997) definition of what augmented reality are:

- It combines both real and virtual content.
- The system interact and performs in real-time.
- The virtual content is registered with the real world.

Augmented reality can be applied in a wide range of field, for example marketing, entertainment, education and military. An augmented reality device are assembled by three things – display, input and tracking [20].

Display

The display are used to show the augmented reality and can be optical see-through, video see-through or projector based. Technologies that are used are for example eye glasses, contact lenses or just a simple screen from a hand device.

Input

There is a wide range of technologies to use for interaction with an augmented reality devices. There are for example gloves, wristband, phone as pointing device and gesture.

Tracking

Tracking is used to align the virtual world with the real world. Which technology to use depends a lot on use area and how good precision and accuracy that is needed. But you can separate the technologies to three different kinds – sensor-based, vision-based and hybrid.

With the sensor-based technology you use the accelerometer, Wi-Fi or GPS.

1.0 INTRODUCTION

With vision-based technology you use either a marker-based or marker-less technology. With the marker-based technology you have a marker as a target and it work as midpoint while the marker-less technology use the surrounding in different ways.

A hybrid technology combine sensor and vision based in different ways.

1.2 GOAL

The purpose of this master's thesis is to investigate existing augmented reality technologies. I will use this knowledge to design and implement my own application to improve the user experience in public transport.

My goals are:

- To investigate what's possible to do with augmented reality and current technology.
- To investigate current applications for public transport.
- To investigate how augmented reality can be used to improve the user experience in public transport.
- To design and implement an application for Windows phone.

2.0 THEORETICAL BACKGROUND

2.0 Theoretical background

2.1 TECHNOLOGY

I start with an analysis of available frameworks for augmented reality to Windows phone to get an idea of what was technically possible. I analyzed them with these questions in mind:

- Which techniques are used?
- On which platform can I use it?
- Is it open source and free use?
- Is it easy to start working with?

I found three different frameworks that I looked deeper into.

2.2 SLARTOOLKIT

SLARtookit [1] is a flexible framework and marker-based. The framework work on the phone and computer and have a strong community with a lot of examples. The framework is originally based on two other frameworks – NyARToolkit [2] and ARToolkit [3]

Which techniques are used?
SLARToolkit use Silverlight, which is an application framework made by Microsoft to run and write Rich Internet Applications (RIA) in Windows or the web browser. Silverlight is comparable to Adobe flash that have similar functionality and purpose. [22]

The benefits of Silverlight is that it is easy to create applications for Windows, web browser and phone at the same time. The only difference is where the image stream is from. The disadvantage of Silverlight is that Microsoft stopped supporting it to work with other technologies.

On which platform can I use it?
The framework works on both Windows and Windows phone.

Is it open source and free use?
Everything is open source and free use and is under the GNU general public license. The license say that everything is free use and that anyone should have access to the source code upon request. [26]

Is it easy to start working with?
SLARToolkis is easy to start work with and there are many examples and projects to run. On the computer and web browser it was fast and efficient, but unfortunately I saw a big delay with the phone and it was difficult to get hold of the marker.

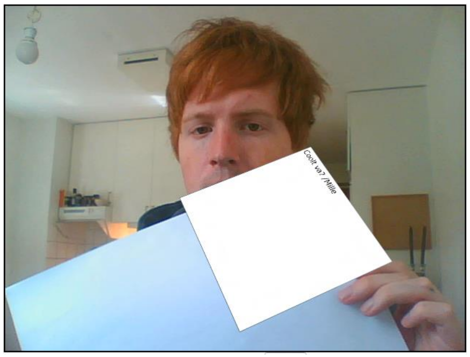


Figure 1 - Test with SLARTookit

2.3 GART

GART [3] is a framework that are made to create easy augmented reality application to Windows phone 7. GART is not marker-based as SLARToolkit, but instead sensor-based. GART places information on top of real places in the world around you. It do this by tracking where you are and in which direction you are facing.

This can easily be used with other services. For example, you can search restaurants with Bing, which returns both latitude and longitude coordinates. You can then use them to show restaurants that are close through the camera.

Which techniques are used?
GART is made just for Windows phone 7 and use a lot of Microsoft own motion API. This limit the framework to Windows phone.

On which platform can I use it?
GART is only available to Windows phone 7.

Is it open source and free use?
The framework goes under Microsoft Limited Public License (MS-LPL), and that means it is okay to develop and use the framework to create software for Microsoft. This means, by definition, that it is not open source (as it must be technologically neutral [27]).

Is it easy to start working with?
As with SLARToolkit there is a lot of good examples and projects for GART, but their examples was a bit messy. The idea was that the application would set out points on the map and an arrow would point in which direction to walk to. But the arrow was quite confusing and it was hard to see what it points toward.

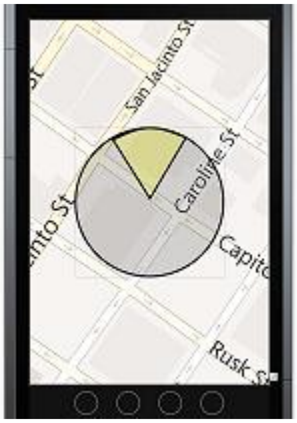


Figure 2 - GART example project

2.4 GOBLIN XNA

Goblin XNA [5] is a framework that is designed to make 3D interfaces, augmented reality and virtual reality. Unlike the other frameworks, Goblin XNA put much emphasis on gaming as it is made from Microsoft own framework to make games, XNA [6]. This gives a couple of advantages such as it is optimized for both 3D graphic and easy to work with physics. It is also marker-based

Goblin XNA just started to support Windows phone and because of this the documentation and information is very limited.

Which techniques are used?
Goblin XNA use ALVAR [7] which is a library to create virtual and augmented reality. This library can create both marker- and feature-based augmented reality.

On which platform can I use it?
Windows and Windows phone.

Is it easy to start working with?
Goblin XNA was the hardest framework to start work with. There were a couple of examples to work with 3D, but it lacked good example to test it with augmented reality. It also forced you to download and sign up on a couple of different websites just to get started.

2.5 CURRENT PUBLIC TRANSPORT APPLICATIONS

In Sweden

To get an overview of was already available in all the current public transport applications, and too see what was considered “standard functionality”, I did an analysis of common features. I inspected their own applications but also third parts.

Most of the application used the same functionality as their homepages and just added functionality as save favorites and history of earlier searches. Nor was there any application that worked across the whole country. The closest was an application named MobieTime [8] that worked over 14 different counties, but this application did just have buss schedule and no other functionality.

Skånetrafiken [28] was the one that listed their applications functionally best. Their functionality was:

- Price information
- Map presentation
- Filter of traffic type.
- Indication of delays
- Traffic messages
- History of latest start and goal points.
- Search with GPS.

Outside of Sweden

I then analyzed applications outside of Sweden.

London tube
London tube [9] is an application that find all subway stations and give the optimal route in London. It also have functionality as notification on delays, time schedules and history.

But the most interesting thing with this application is its augmented reality part. It show close restaurants, cycle parks and bus stops through your camera. It show information as distance, name and a matching icon.

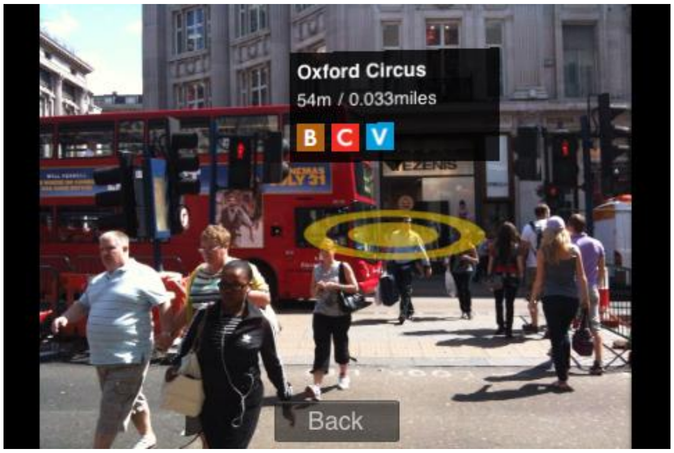
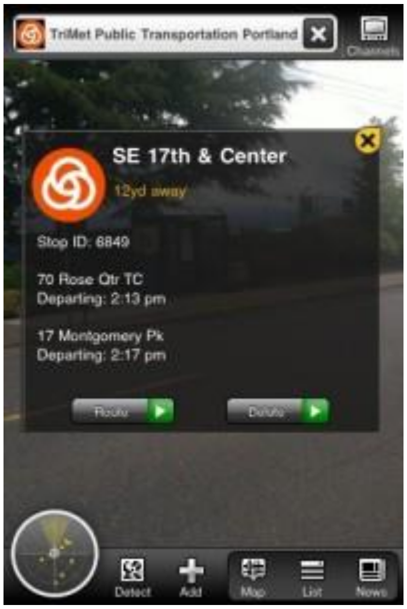


Figure 3 - London tube[9]

2.0 THEORETICAL BACKGROUND

Junaio browser

Junaio Browser is an application that can show close bus stops, distance and when the next bus will arrive in real time. You will be able to get all of this information by looking with your camera. The user will also be able to put out notes and comments that other users can find. Junaio is made from the company Metaio[10] that is specialized in augmented reality.



TriMet Public Transportation Portland

SE 17th & Center

12yd away

Stop ID: 6849

70 Rose Qtr TC
Departing: 2:13 pm

17 Montgomery Pk
Departing: 2:17 pm

Figure 4 - Junaio brower [25]

Catch the bus

Catch the bus [11] is an application that show where all the buses are located in real time. By using the GPS on every bus it can show where the bus is right now and determine exactly how much time is left before the bus arrive.

3.0 DESIGN

3.0 Design

3.1 AUGMENTED REALITY DECISION

Since there were a couple of different augmented reality technologies I had to decide which of them I wanted to use. They could all be used in different ways, but the problem with SLARTookit and Goblin XNA was that they were marker-based.

The problem with marker-based is that the user must either to print out the marker or it need to be provided. For public transport there could be markers at the bus stops, but it's hard to see how that would be used. There was also problem with SLARToolkits efficiency that made it hard to use.

That's why I decided for GART. As GART is sensor-based I had a lot more freedom and I could use it as London Tube or Junaio browser. I could for example use GART to show close bus stops with the camera and GPS.

3.2 SCENARIOS

The next step was to create scenarios to help me define my applications functionality.
I built my scenarios around normal use and from situations I thought a person could be in.

Task and goal	Sub target
Gustav has come to a new city with the train, and only know the address of where he is going. Now he wonders where the nearest bus stop is and which bus he should take.	- Se close bus stops. - Be able to search for buses from a position or bus stop that goes to a specific address.
Stig is visiting the city and getting tired. He cannot walk further and wonder where he can take the next bus home.	- Be able to search buses and bus stops that take you home with the help of different priorities.
Lars is playing games at a friend's house. Buses run less frequently here and he feels that's easy to easy miss them when he is focusing on other things.	- Be able to get notifications and reminders from different bus stops.

3.0 DESIGN

Lars would like a reminder when there is ten minutes left to the bus arrive.	
Frans just missed the bus to his job interview and feel extremely stressed. He now needs to know if there is another way for him to travel that may give him a chance to get there on time.	- Be able to search from current position to another with priority on time.
Jakob are going to party, but he does not remember the address or which bus stops that are close.	- Be able to click on the map to get the optimal route from current position. .
Magnus usually travel from different communes with the bus. He therefore feels that is start to get tiresome to always write his home address when he goes home.	- Be able to save home position. - Be able to find the closest way home.
Kent are out in the woods and feel a bit lost. He just want to get home and are looking for a way out and to a good bus stop. But he are not that good with maps and just want a direction to walk in.	- Search for close bus stops. - Be able to see direction to walk in.
Olle is out and thinking about taking the bus home. The problem is that he doesn't know when the next bus go.	- Be able to see time schedule from different bus stops.
Rolf is at the beach with his children and thinking about taking the bus home. He know that there is a couple of different buses that go, but some of them have more changes than others. He would like to do as few changes as possible because the children.	- Be able to search for the best route him with priority on bus changes.

Agnes know that the buses at this bus stop always come a bit too early. She therefore feel a bit sorry for those who don't know this and miss their bus.	**-** Be able to put out notes that other can read.

3.3 FUNCTIONALITY

From my scenarios and investigation of current technology I created a list of functionality that I wanted my application to have. I chose to priorities them all from one to three - one means "shall be implemented" and three means "implement if there is time".

- Be able to see close but stops on the map [prio 1]
    - Shall be able to click and see close bus stops.
- Be able to see close bus stops through the camera[prio 1]
    - Shall be able to click on them for time schedule and more information.
- Five different kind of searches.
    - Search with address [prio 2]
    - Search with bus stop[prio 2]
    - Search with current position/position on map[prio 1]
    - Search with home position [prio 1]
    - Search with favorites[prio 3]
- Favorites[prio 3]
    - Shall be able to save, delete and use favorites.
- Notes
    - The user shall be able to put out notes that other users can see on the map and with the camera. [prio 2]
- Sms ticket [prio 3]
- Travel help [prio 1]
    - Give detailed results that guide the user through the whole journey with text, map and augmented reality in the camera.

3.4 TOOLS FOR DESIGN

To make my design and sketches I used a program from Microsoft called Microsoft Expression Blend 4[12].

3.0 DESIGN

Microsoft Expression blend is a tool to make user interface for web and desktop applications. It also got a special feature called SketchFlow that are used to create prototypes. It let you present your work with a consistent “prototype”-look to keep focus on the concepts presented.

One of the big reason I used this tool was because it could put out all the controls and component that I could use when creating my real phone application. That made it easier to know what I could and not could do later on.

3.5 BASIC DESIGN PHILOSOPHY AND MICROSOFT GUIDELINES FOR USER EXPERIENCE

When I design my application I will follow Microsoft's guidelines for user experience in Windows phone [14]. This is to make sure that my application feel and look like the rest of the system and to keep the design consistent. Consistency is an important part and is the first of Schneiderman’s eight golden rules for interface design [13].

Microsoft guidelines contain information for most of the controls in an application, but also information as application structure, user and platform interaction and the design process. The guidelines for controls contained information as:

- Appearance and action
- Detailed and description
- Standard use
- Design guidelines

For example, buttons should never contain more than two words and it should be concise and typically a verb.

The guidelines also come with a lot of examples how you should design your graphic for your application.

3.0 DESIGN

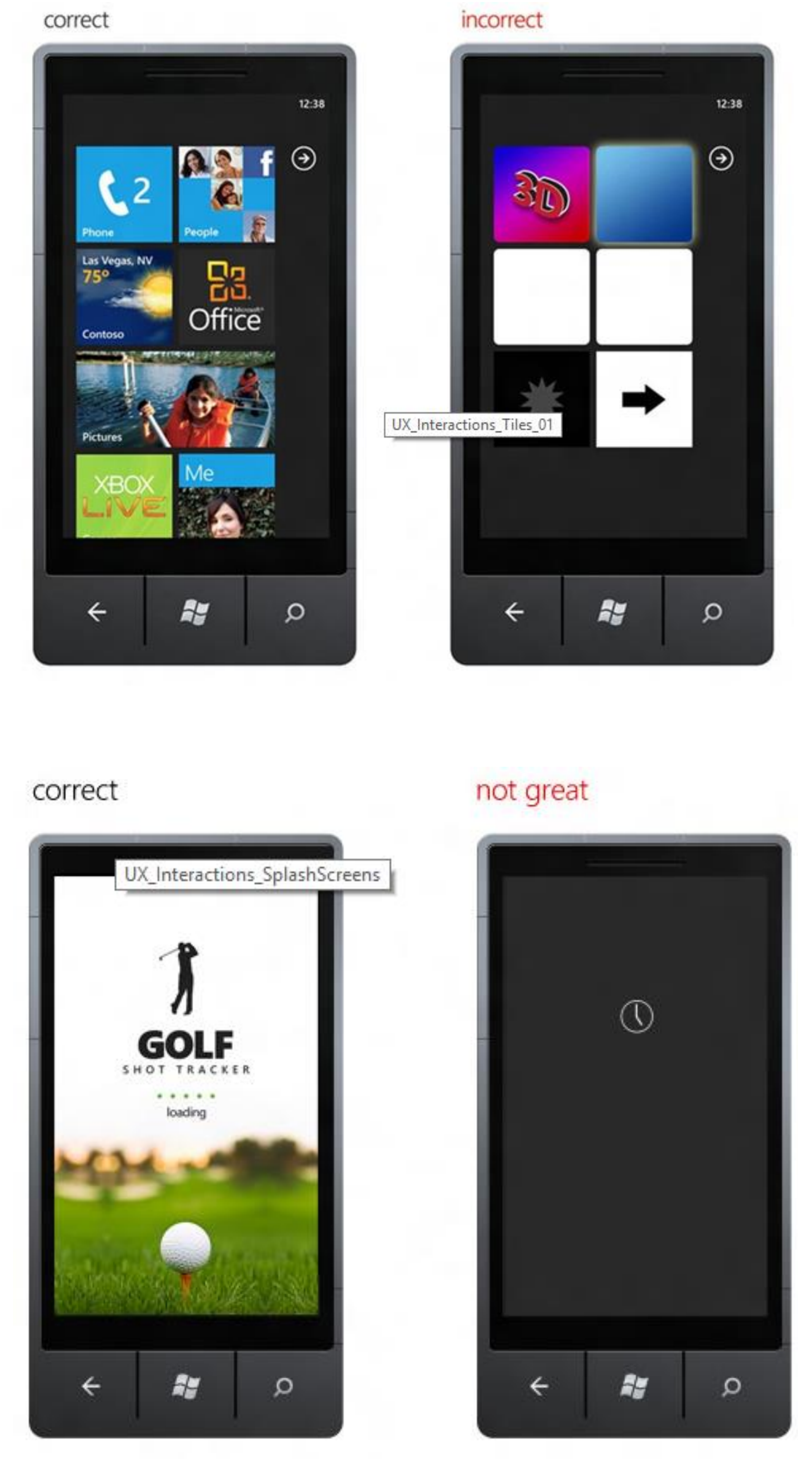


Figure 5 - Correct and incorrect splash screen examples [14]

3.6 FLOWCHART

I created a flowchart to easily demonstrate how a user navigate through the application. It show the all paths through the applications and all the screen it contains.

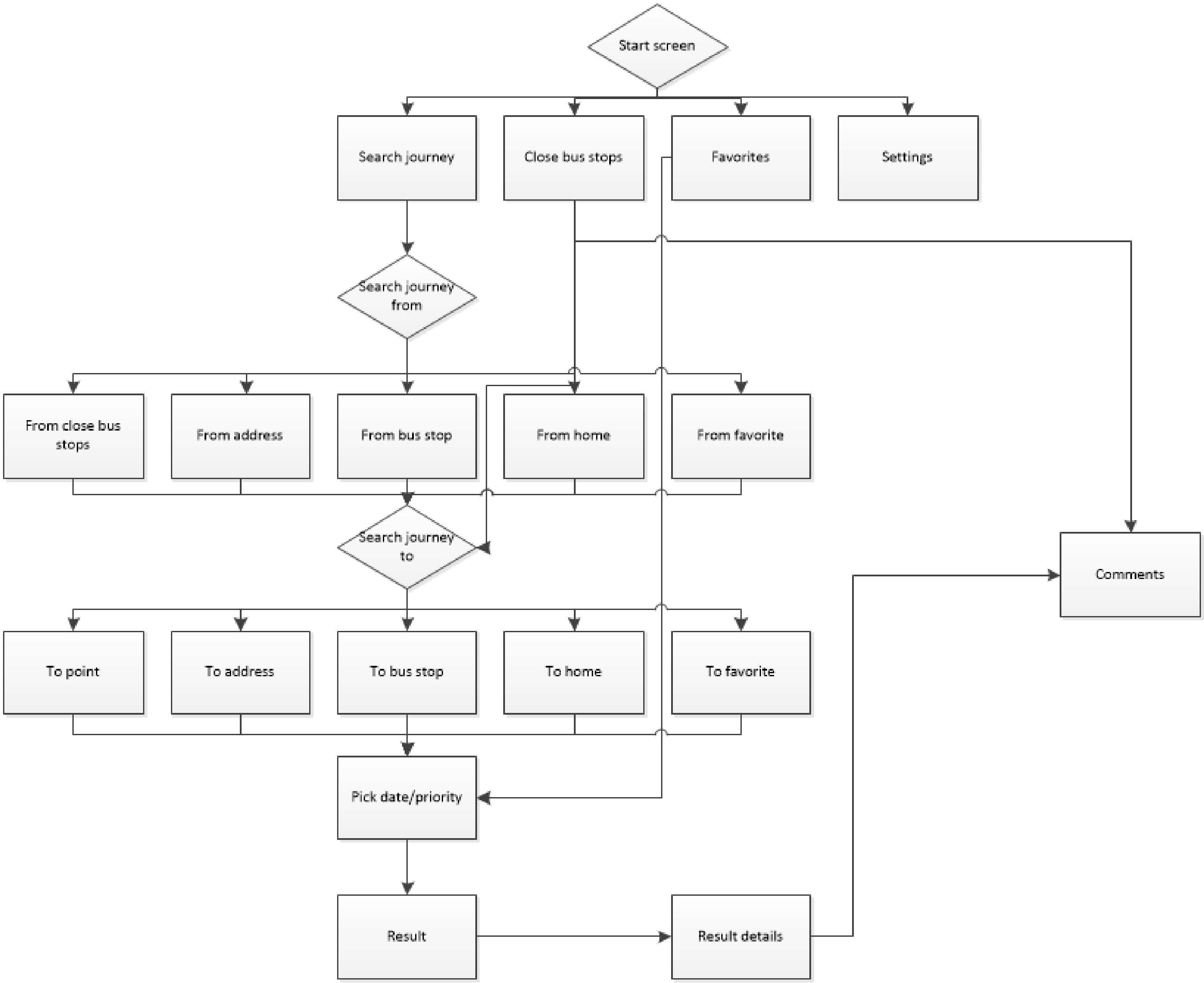


Figure 6 - Flowchart

Start screen - The application start here and just contain a simple menu to guide the user further in the application.

Search journey- The user search journeys here and I chose to priorities flexibility instead of lowering amount of click. I wanted the user to be able to use all the different combinations of searches.

Result - Here the user can see the results. This page is created to show basic information as time, from, to and amount of changes. If the user want more information he must click pass this and go to result details.

Result details - Give the user more information and also guide through the journey with text, map and augmented reality in the camera.

Comments - Show notes, comments and pictures from other user.

Close bus stops - Show all close bus stops on the map or through the camera.

Favorites - Here the user can travel with his favorites or edit

Settings - Change settings.

3.7 LOW-FI PROTOTYPING

Here I will show all my low-fi prototyping and design.

Start screen

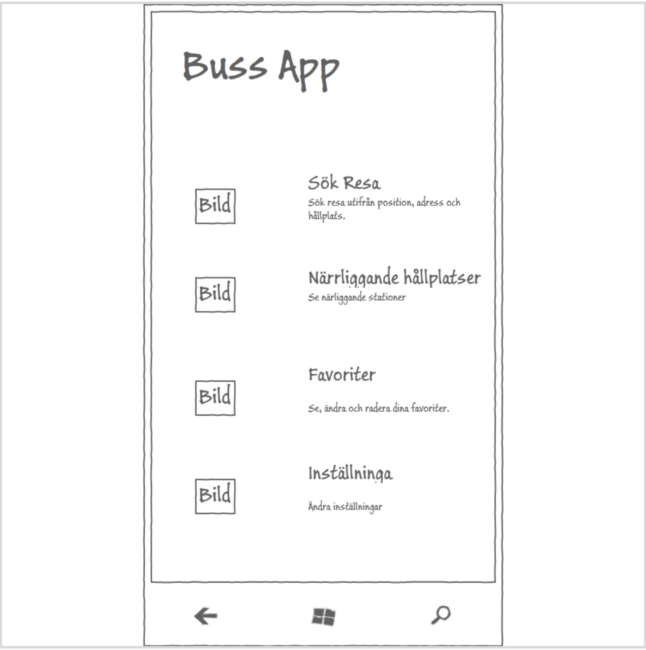


Figure 7 - Start screen

3.0 DESIGN

The start screen are supposed to guide the user further and show the applications functionality. I have therefore decide to use both icon, title and explaining text. I don't want the user to feel insecure about which functions that exist and what's happening when they click to next step.

I will also follow general guidelines for icons: [13]:

- The icon represent the action.
- Each icon is distinctive from every other icon.
- They are created in harmony and as a member of a family of icons.

To make sure I follow these I use Microsofts own icons later in the application.

Search journey from bus stop screen

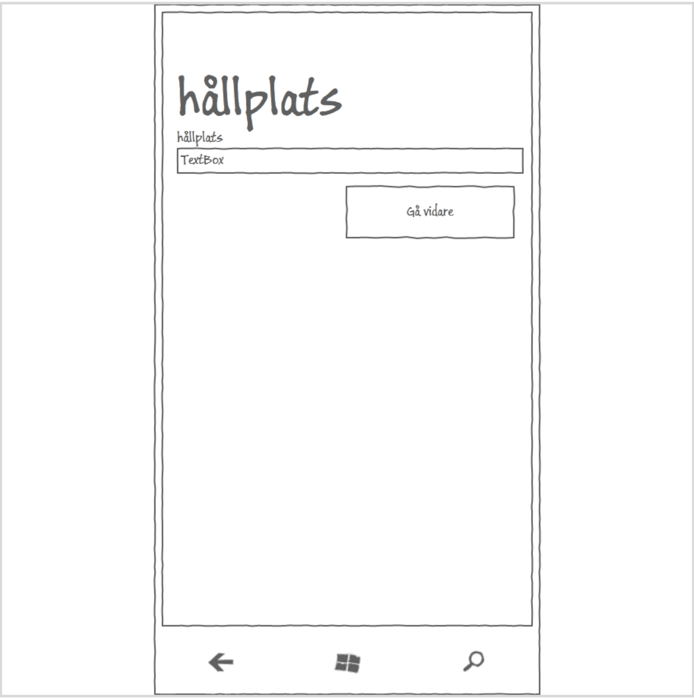


Figure 8 - Search journey from bus stop version 1

3.0 DESIGN

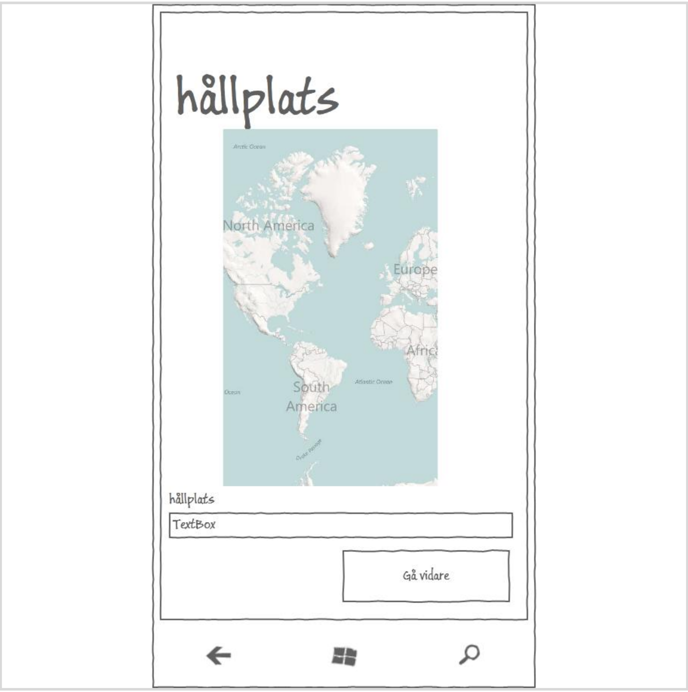


Figure 9 - Search journey from bus stop version 2

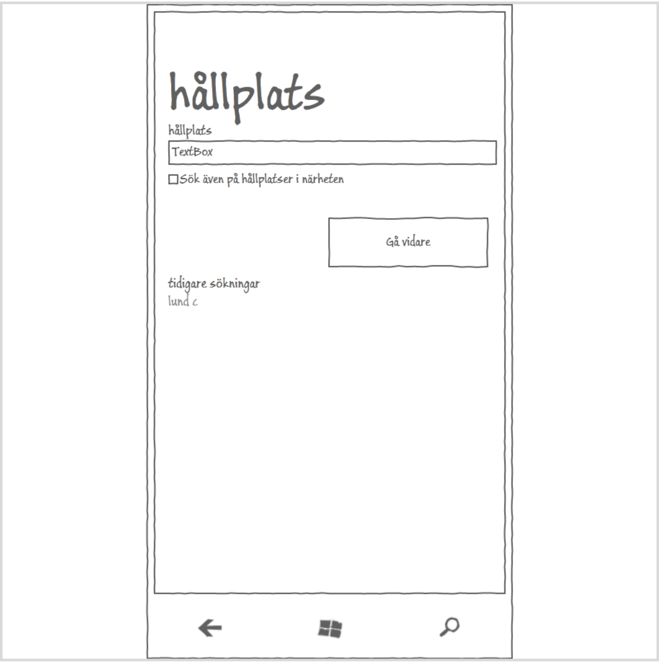


Figure10 - Search journey from bus stop version 3

Search journey from bus stop went through a couple of changes before I got a final version that felt good. I felt that the first version was very plain and not exciting enough for the user. It also forced the user to write the bus stop name every time as it did not save history.

In version two I got a screen that was more exciting. Every time the user wrote a new bus stop the map zoomed into its location that let the user know where the bus stop is located. But it felt like the map could make the user confused, as it was so big and did not have any other purpose than to be "interesting". It would also add a couple of seconds of loading and delay and made the experience less smooth.

In my third and final version I removed the map, added history and a new checkbox to search bus stop that are close to the bus stop you picked. I decided to keep the screen a bit less interesting but instead faster and more functional.

Search journey from address screen

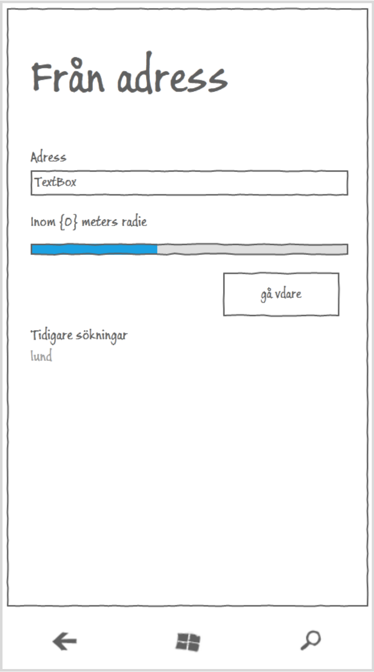


Figure 11 - search journey from address

The "from address"-screen contains a textbox to write the address and a slider to decide search radius. As I already tested a couple of things with my "search from bus stop"-screen I felt that I didn't have to test as much on this one. The only thing that was new was the slider that let the user decide the search radius and I think it's pretty simple to understand.

Search journey to point Screen

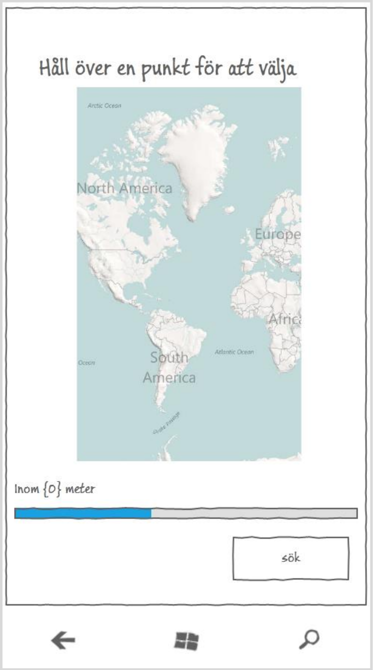


Figure12 - Search journey to point screen

On the “search journey to point”-screen the user click on a spot on the map and the application will search for bus stops that are close to the user. As with the address screen this screen use a slider to decide the radius of the search.

When screen

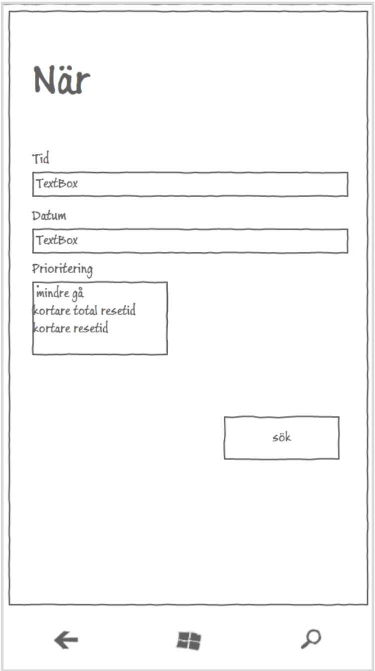
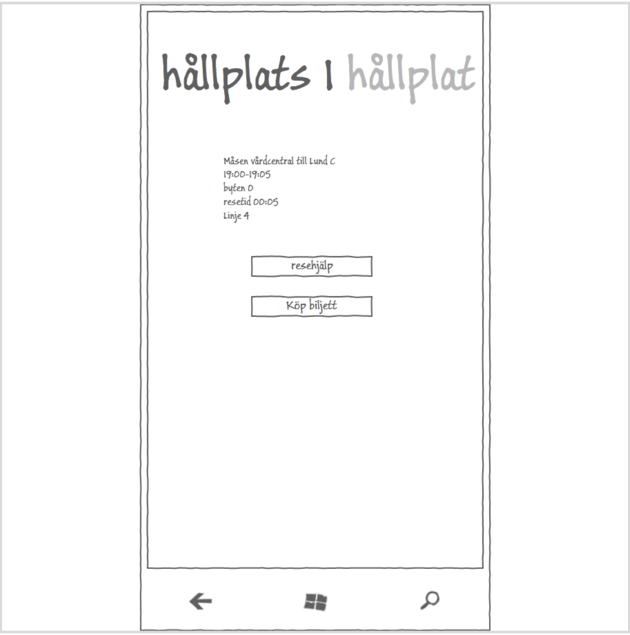


Figure 13 - When screen

The when screen contains a date and menu control. It let the use decide date and time to travel and search priority. The different search priorities are:

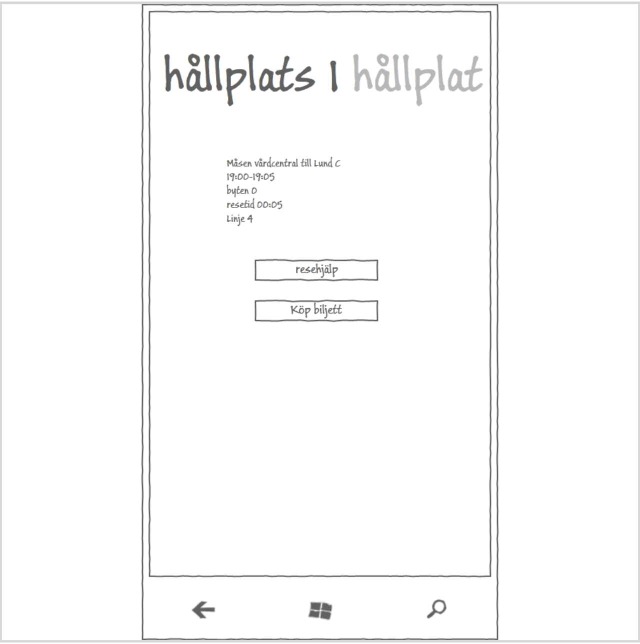
- Walk less – Show journeys with close bus stops if there is bus changes show the ones with the least distance between the bus stops.
- Shortest travel time – Show journeys with the shortest travel time.
- Shortest total time – Show journeys with the shortest total time (time to bus departure and travel time).
- Least changes - Show journeys with the least number of bus changes.

Result screen



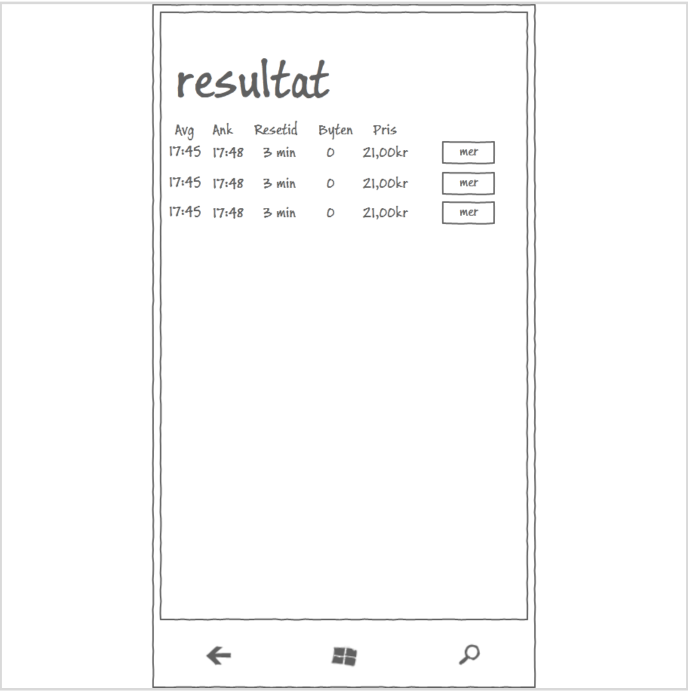
Måsen vårdcentral till Lund C
19:00-19:05
byten 0
resetid 00:05
Linje 4

Figure 14 - Result version 1



Måsen vårdcentral till Lund C
19:00-19:05
byten 0
resetid 00:05
Linje 4

Figure 15 - Result version 2



Avg	Ank	Resetid	Byten	Pris	
17:45	17:48	3 min	0	21,00kr	mer
17:45	17:48	3 min	0	21,00kr	mer
17:45	17:48	3 min	0	21,00kr	mer

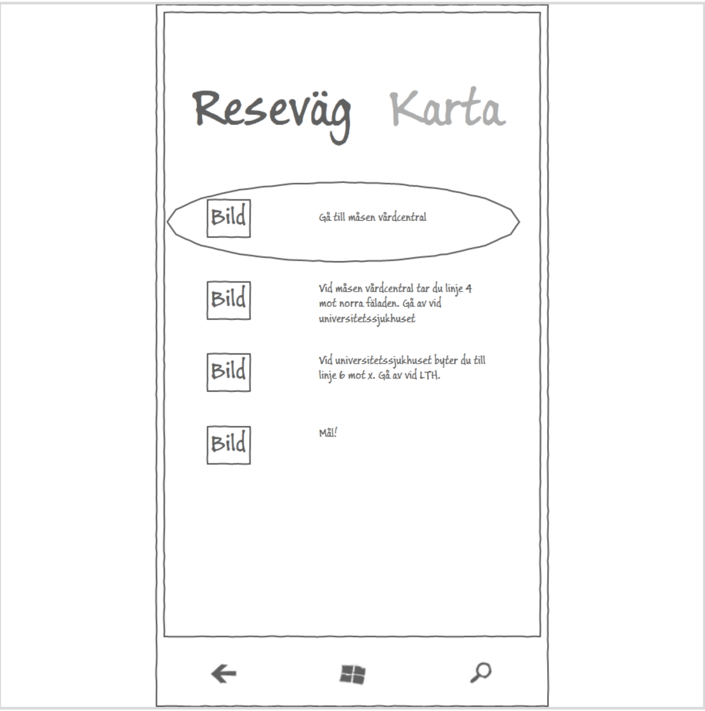
Figure 16 - Result version 3

Also the result page went through a couple of changes before I got to the final version. In my first version I used a simple list, but it was a bit messy and hard to read as the screen are so small. The image to the left did also use up a lot of unnecessary space.

In my second version I used a pivot control. A pivot control is like a list with screens and the user swipe the screen to the left or right to change item. It gave me a lot of space but also made it hard to get a good overview of the times. Microsoft user guidelines also say that a pivot page should never have more than six items as that will confuse the user [15].

In my final version I used a table. It was the best way to get a good overview of the data and it was not as confusing as the first version. It is also how Skånetrafiken show their data on the homepage.

Result details screen



Gå till måsen vårdcentral

Vid måsen vårdcentral tar du linje 4 mot norra fäladen. Gå av vid universitetssjukhuset

Vid universitetssjukhuset byter du till linje 6 mot x. Gå av vid LTH.

Mål!

Figure 17 - Detailed result

On the detailed results page I wanted to show all the information about a journey, but also guide the user through it. I wanted to guide the user with both text, map and augmented reality to make the experience new and fresh. So a pivot control was perfect here. With the pivot control I could show the user three different views - travel way, map and camera.

Travel way - Detailed information in text that guide the user through the journey. It shows where the user have to go, which distance he has to walk, and how much time there is left to the bus depart. I decided to show this as a list where the current step was highlighted in a strong color to let the user know which step he or she are on.

The user would be able to go to the next step in two different ways. Either automatically where the application feel when the user is near the goal position or done manually. The user can change this on the settings page.

Map - The map will show the users current position and his goal position. The goal position depends on which step of the journey he are on. For example, if the first step is to walk toward the bus stop the map will just show the user and that bus stop.

Camera - The camera will show the direction and where the bus stop are.

Close bus stops screen

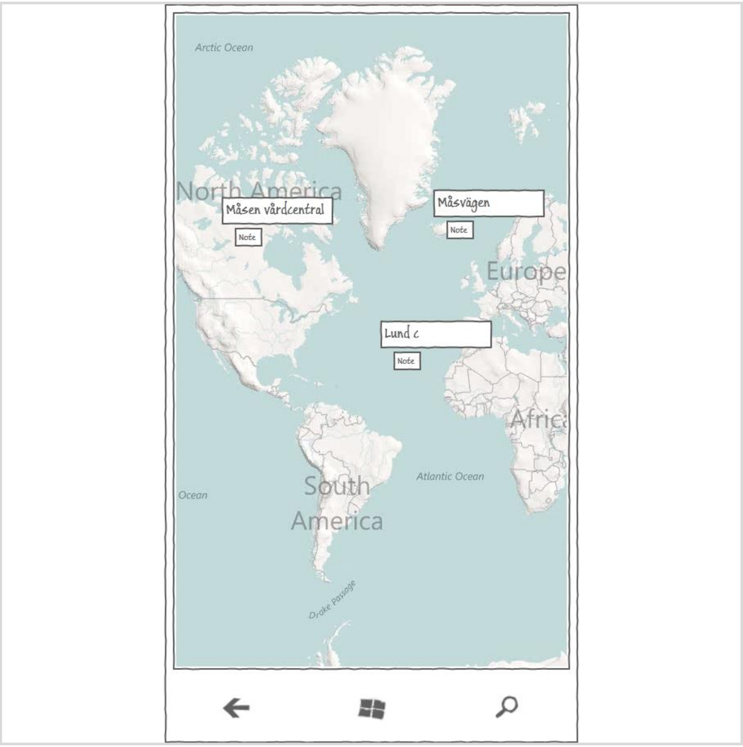


Figure 18 - Close bus stops screen

The “close bus stops”-screen show a map with all bus stops that are close to the user. It also show an icon for comments related to the bus stop. The user will be able to change how the bus stops are represented in a couple of different ways.

- As icons
- As Text
- As Icon and text

They will also be able to decide if they want to show the distance to the bus stop from their current location and if they want it to be color coded. Color coding help the user recognize the bus stops that are closest faster [13].

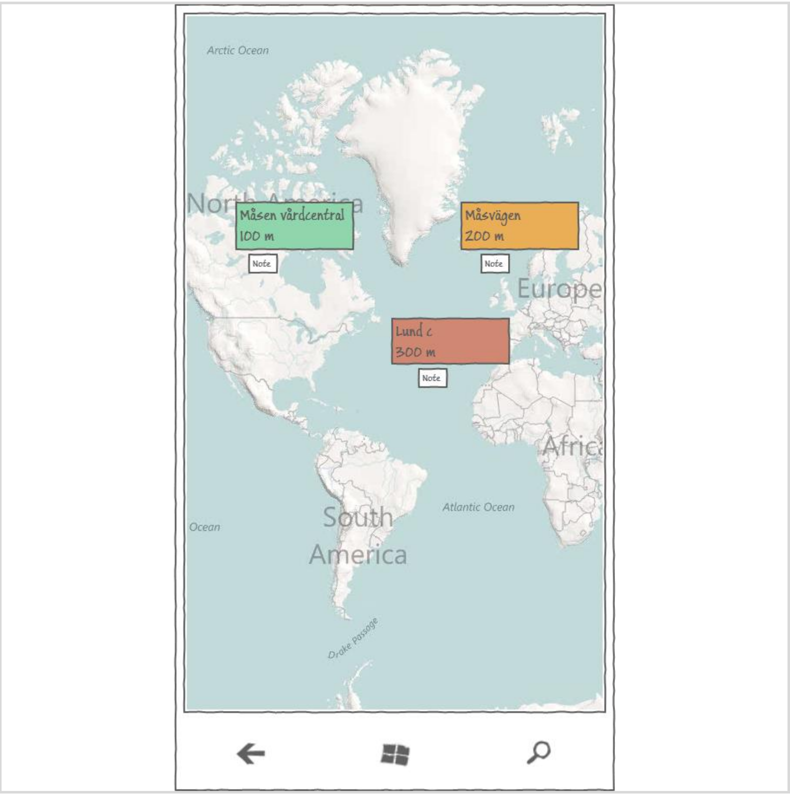


Figure 19 - Close bus stop with color coded icons

Favorite screen

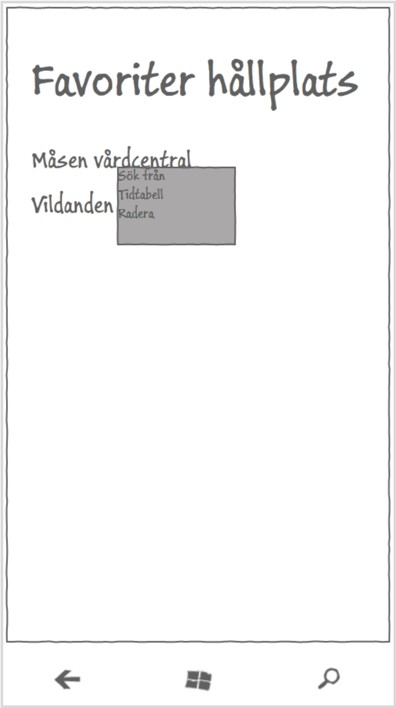


Figure 20 - Favorite bus stop screen with menu pop up

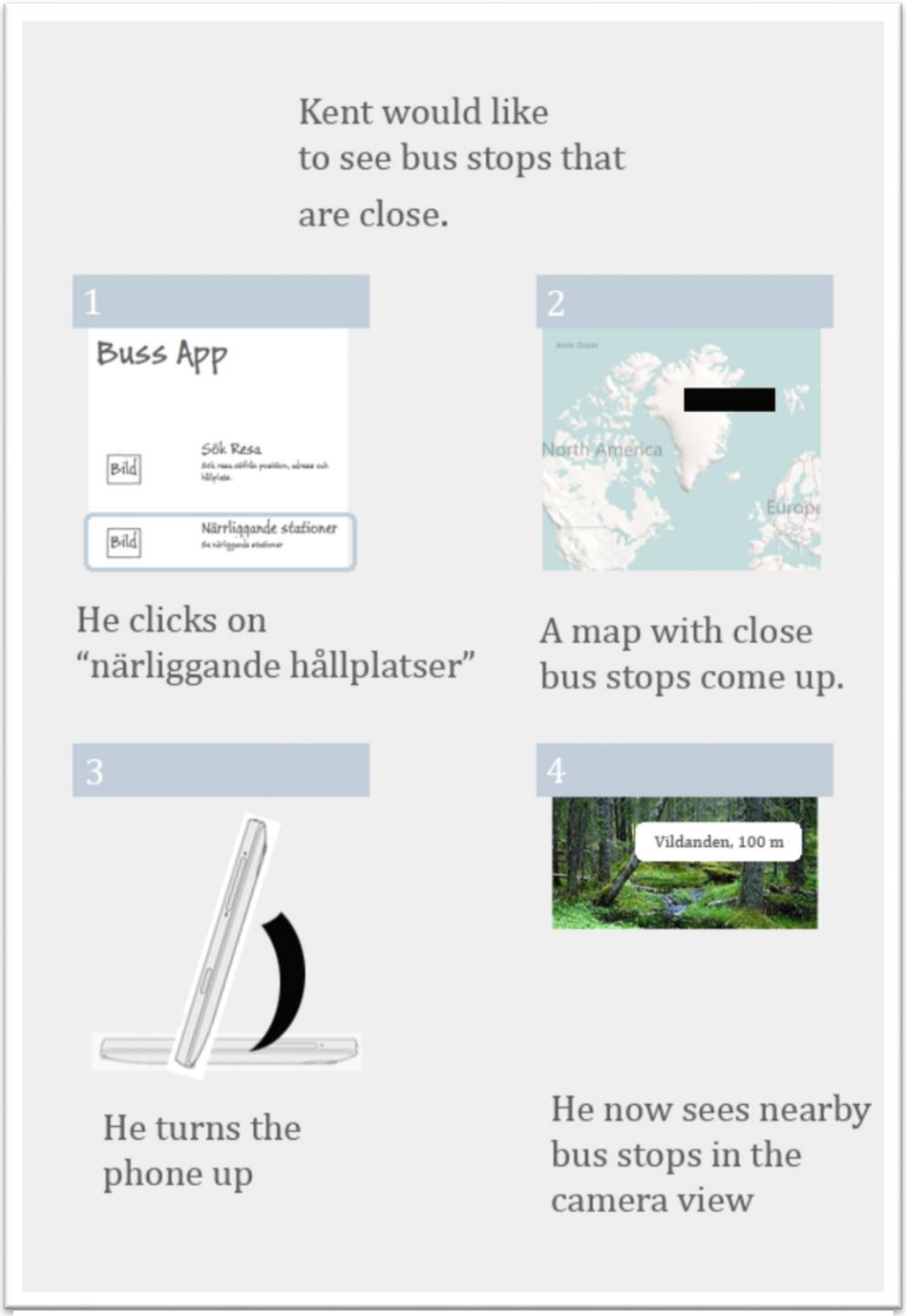
There is a couple of different things that the user can do on the favorite screen. For example, he can search with a favorite, see bus schedule and remove a favorite. To keep it simple I split favorites into three screens. The favorite start screen where the use can decide if he want his favorite bus stops or journeys the favorite bus stop screen with a list of all bus stops and a favorite journey screen with a list of all journeys.

To save space I used a pop up menu that gets visible if a user click on an item. It contains the items – search from, time schedule and erase.

3.7 STORYBOARD

As the augmented reality functions would work a bit different I created a storyboard. The storyboard would work as an easy way to illustrate how it could work but also as a description during the final evaluation session.

The storyboard I created show how a user may use the application to show close bus stops with the camera.



Kent would like to see bus stops that are close.

He clicks on "närliggande hållplatser"

A map with close bus stops come up.

He turns the phone up

He now sees nearby bus stops in the camera view

Figure 21 – Storyboard how the user may use the augmented reality part

4.0 IMPLEMENTATION AND TESTING

4.0 Implementation and testing

This chapter deals with the implementation and testing of the application.

4.1 SOFTWARE

To create applications for Windows phone you need to use Visual studio. It is an integrated development environment (IDE) made by Microsoft. In addition to just developing it also have a "what you see is what you get"-editor and performance analysis tools. When developing applications for the phone, it also have an emulator that can be used for testing.

The only problem with visual studio is that it have high system requirements and to run the phone emulator you need the latest version of visual studio and Windows 8 64-bit.

4.2 LANGUAGE

You can separate a Windows phone application in to two parts, the user interface [UI] and code-behind.

UI

The UI is built with XAML that is a declarative XML-based language made by Microsoft [29]. The language is built to make it easier to create UIs for .NET framework applications and are very similar to XML.

Example how create a button:

```
<StackPanel>
  <Button Content="Click Me"/>
</StackPanel>
```

It's very flexible and easy to use when you worked with it for a while.

Code-behind

The code-behind can be written in a couple of different languages that depends on if you create your application for Windows phone 7 or Windows phone 8. For Windows phone 7 you can only use C# or visual basic bug for Windows phone 8 you use:

- C#
- C++
- Visual basic
- JavaScript

As I started to build the application for Windows phone 7 and later wanted to port it to Windows phone 8, I chose C#. It was also the language I knew the most so it made the choice easy.

4.3 DATA FROM SKÅNETRAFIKEN

As the application are supposed to show bus schedules and bus stops, I needed some way to get the data from Skånetrafiken. Thankfully, Skånetrafiken have their own API and allow everyone to use it. [16].

Skånetrafiken send their data as SOAP [17] and that is a protocol specification for exchanging structured information. SOAP is an XML based protocol that consists of three parts: an envelope that describe what's in the message and how to process it, a set of encoding rules for how expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP can also be used over any transport protocol, such as HTTP, SMTP and TCP.

SOAP can be handles in a couple of different ways with C# and .NET. You could either use web services or parse the message yourself. After a couple of tries I decided to go with parsing. It is not the best way to do it but I felt that it gave me full control of what I want to do with the data.

I could send seven different requests too Skånetrafiken:

[Querypage] – Search for depart and goal points.

[Resulpage] – Search for a journey

[Querystation] – Search for a station/bus stop.

[Neareststation] – Search for nearest station/bus stop from a point.

[Stationresults] – Departure time from a station/bus stop.

[Trafficmeans] – Current way of travel means

[Journeypath] – Travel path coordinates

4.4 CODE DESIGN AND DEPENDENCIES

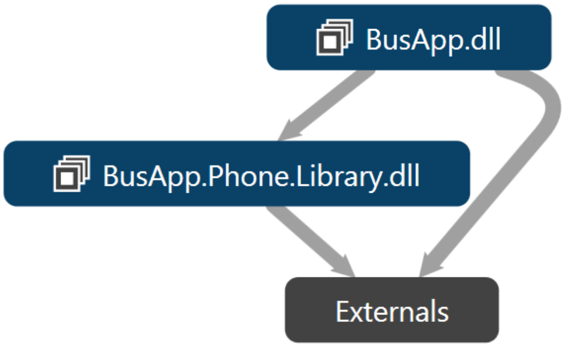


Figure 22 - Dependency diagram

To make my code reusable in other applications (for example a bus application for table/computer) I had to design and separate it in to two projects.

BussApp.dll

In this project I have all my UI-classes and everything that is just for Windows phone 7 or 8.

BussApp.Phone.Library.dll

In this project I have all my models and data classes.

My design is close to the MVC (model-view-controller) pattern where BussApp.dll is my view and controller and BussApp.Phone.Library.dll is my model. The controller send commands to the view to change its presentation of the model. Then the view sends a command to update the models state and when the model is done it notify the view about the update.

4.5 SCREEENS

Here I will go through my screens and implement choices for each one of them.

4.0 IMPLEMENTATION AND TESTING

Start screen

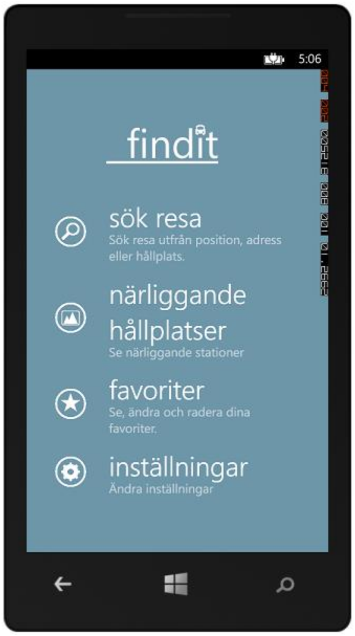
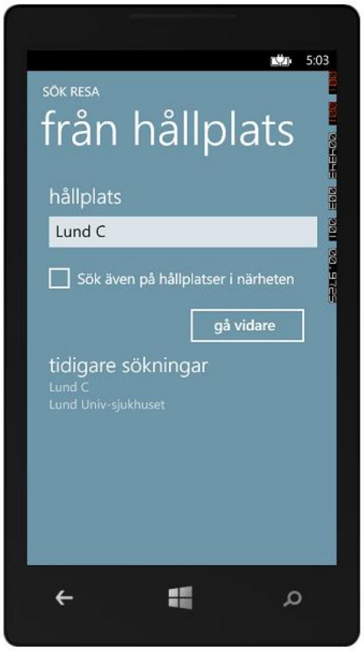


Figure 23 - Start screen

The start screen contains a listbox with an image and two textboxes. This was the easiest way to do it as there was no real "menu control". For the images I use the Windows phone standard icons as they have the same look as the rest of the application.

Search journey from bus stop screen



SÖK RESA

från hållplats

hållplats

Lund C

□ Sök även på hållplatser i närheten

tidigare sökningar

Lund C

Lund Univ-sjukhuset

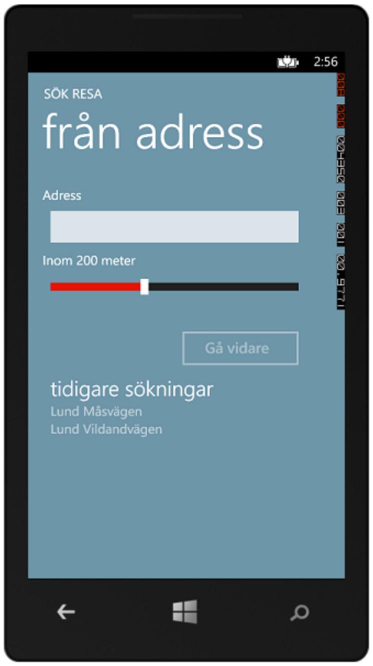
Figure 24 - Search from station

The biggest implementation with the ¨search journey from bus stop˝-screen was the auto-complete textbox. There was a control for this but I had to populate it with information from Skånetrafiken. So I had to do a request with the **[Stationresults]-**method to get all bus stops that contain the current text input from the user. Then I parse out the bus stop names and populate the text box list.

I had two different ideas how to handle the search history. My first idea was to save it as a list in application settings. Application settings is a dictionary that keep settings between different runs. But unfortunately the application settings could only store built-in data types (int, bool, string, etc) so I had to go with my other idea to binary serialize and save to the isolated storage.

4.0 IMPLEMENTATION AND TESTING

Search journey with address screen



SÖK RESA

från adress

Adress

Inom 200 meter

tidigare sökningar

Lund Måsvägen

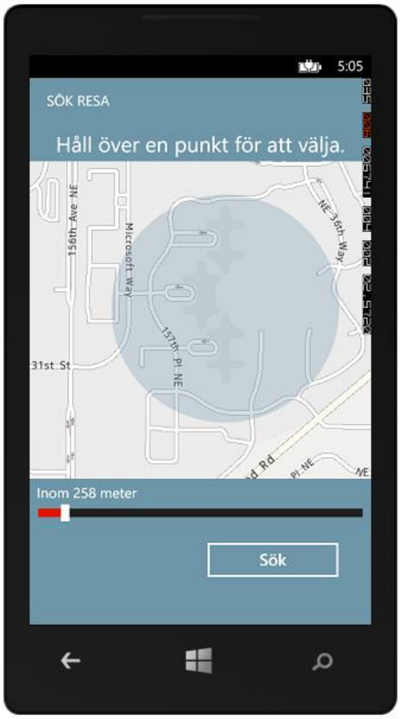
Lund Vildandvägen

Figure 25 - Search with address screen

The “search journey to address”-screen contains an autocomplete textbox and a slider. To fill the autocomplete textbox data I needed to find a provider with addresses. With Windows phone 8 it is easy to use Nokia maps for this, but with Windows phone 7 I would had to use some kind of API from a page like eniro [21].

With Nokia maps I send a map request and get a list back with map location objects. The map location object contain information as location (latitude and longitude), address and name. Then I could use this information to populate my textbox.

Search journey to point screen



SÖK RESA

Håll över en punkt för att välja.

Inom 258 meter

Figure 26 - Search to point screen

The “search journey to point”-screen contains a new control - the map control. The map control is one of the big difference between Windows phone 7 and Windows phone 8. With Windows phone 7 you use Bing maps and with Windows phone 8 you use Nokia maps. They are similar in many ways but Nokia maps have richer functionality, especially for routes.

I had to figure out a good way for the user to set out search points and as the map control catch most of the user inputs I did not have a lot to choose between. The best and easiest way was to let the user hold the finger down in 2 seconds.

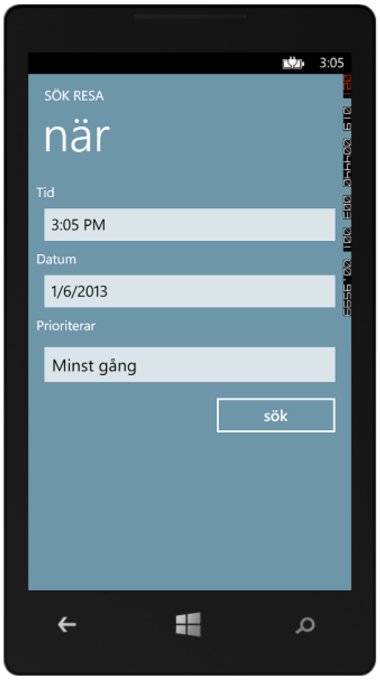
When the user hold their finger down I will create a circle on that position that show the search radius. To draw the circle I needed to know how many meters there is for one pixel. The formula I use for this is [23]:

$$meters\ per\ pixel = \frac{\cos\left(latitud * \frac{\pi}{180}°\right) * 2 * \pi * 6378137}{256 * 2^{zoom\ level}}$$

This formula will give me how many meters there is for one pixel and to decide the draw radius I just have to divide with this value.

$$draw\ radius = \frac{real\ world\ radius}{meters\ per\ pixel}$$

When screen



3:05

SÖK RESA

när

Tid

3:05 PM

Datum

1/6/2013

Prioriterar

Minst gång

sök

Figure 27 - Date screen

The “when”-screen consist of a timepicker, datepicker and a listbox controller. The time and date picker work the same and the user navigates to a new page where he can pick the date or time.

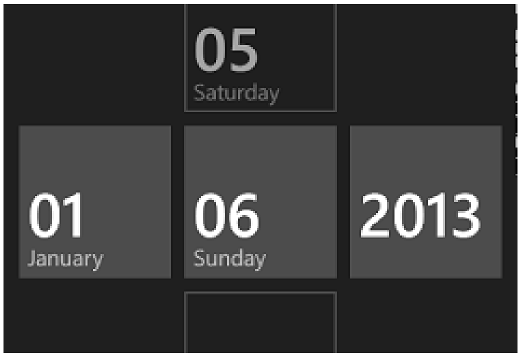
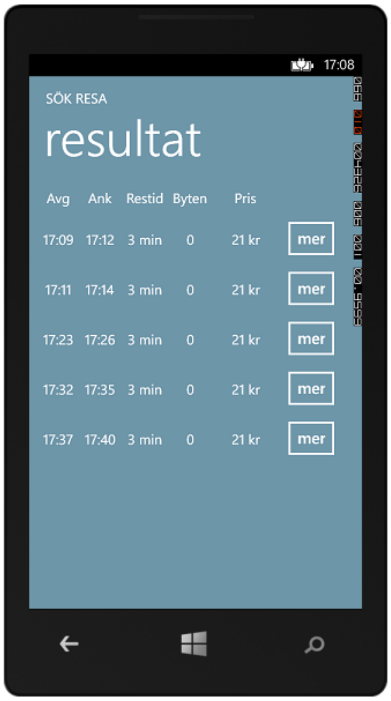


Figure 28 - datepicker control

The listbox controller contains all the search priorities that the user can pick (see design for more information). I decided to let the user save which priority he wanted to standard in settings, so he don’t have to re-pick it every time he come to this page.

Result screen



17:08

SÖK RESA

resultat

Avg	Ank	Restid	Byten	Pris	
17:09	17:12	3 min	0	21 kr	mer
17:11	17:14	3 min	0	21 kr	mer
17:23	17:26	3 min	0	21 kr	mer
17:32	17:35	3 min	0	21 kr	mer
17:37	17:40	3 min	0	21 kr	mer

Figure 29 - Result page

The result page contains a table with 7 columns. There was no table control so I had to create my own table by using list box and stack panels.

It's also the page that call send most requests to Skånetrafiken. The amount of requests I need to do depends on the search. A simple search from one bus stop to another is just one request but If I made a bit more advanced search like from current point to another point it could go up to 25-30 requests.

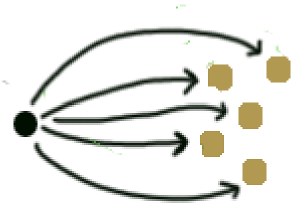
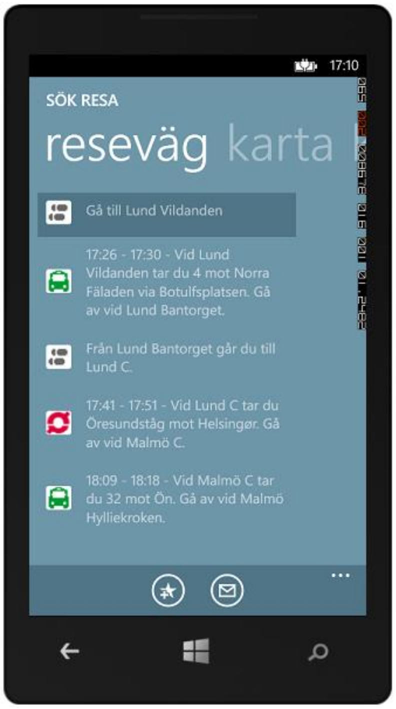


Figure 30 - Amount of requests when I search from one bus stop (black) to five other bus stops (brown)

When I did more than 20 requests I received a delay and that suggests that skånetrafiken have a limitation on the numbers of request you can do in a short time. As my search technique was pretty brutal I tried to figure out some other way to do it and lower the amount of requests I needed to make. To my disappointment I could not figure out any other way.

Result details screen



Gå till Lund Vildanden

17:26 - 17:30 - Vid Lund Vildanden tar du 4 mot Norra Fäladen via Botulfsplatsen. Gå av vid Lund Bantorget.

Från Lund Bantorget går du till Lund C.

17:41 - 17:51 - Vid Lund C tar du Öresundståg mot Helsingør. Gå av vid Malmö C.

18:09 - 18:18 - Vid Malmö C tar du 32 mot Ön. Gå av vid Malmö Hylliekroken.

Figure 31 - Result detail screen - reseväg

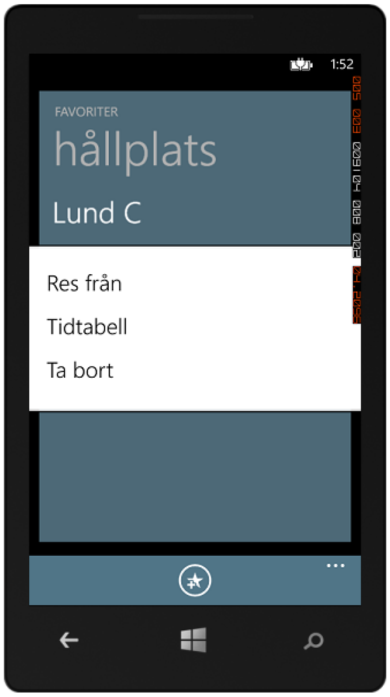
The “result details”-screen contains a pivot control. With the pivot control you can separate the screen into sub screens and let the user swipe to the right or left to change sub screen.

The first sub screen is “reseväg” that contains a list with directions for the user. I made the current direction a bit darker to let the user show where he is and what he are supposed to do.

The second sub screen is “karta”, it show the users current position and the goal on their current step. It will also show the walk route if the current step is walking.

The third sub screen is a camera screen and it show the goal on the current step and which direction the user have to take. I write more how this work on the augmented reality screen.

Favorite screen



FAVORITER

hållplats

Lund C

Figure 32 - Favorite screen with drop down menu

The favorite screen contains a listbox and a context (popup menu) control. To show the context control the user has to press and hold on an item in the listbox for two seconds and then the menu popup. At first I wanted to use this menu for everything and show all the alternatives (Search from, delete etc.) but as the delay was so long between input and feedback it felt a bit sluggish. I then changed it to so a fast push made the user search and if they wanted more options they had to hold.

The hardest part with these screens was how I handled the saving and loading. I decided to create a new class called FavoriteData in my library project that I used for loading and saving my favorites. I then load all favorites every time I started the application. This may be a bad idea sometimes but as I load asynchronous and the files usually are every small it won't slow down the application in any significant way.

I saved binary as it was the easiest way as I already had a working binary saving class (that I used earlier to save history).

Close bus stops screen

The close bus stop screen just contain the map control. As I wrote before the big difference between Windows phone 7 and 8 is the map control and how you work with them. With Windows phone 7 you use Bing maps and pushpins and with Windows phone 8 you use Nokia maps and your own custom controls.

This make Nokia more flexible as you can use you can put anything you want on your map. So I used a couple of different techniques to create my icons and labels.

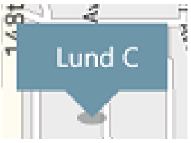


Figure 33 - map text icon

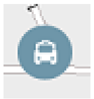


Figure 34 - icon

The label I created with polygons and a text block. Polygons is a very flexible way to work but it's also time consuming. The icon is created with an ellipse and an image box.

Augmented reality Screens

The "result details"- and "close bus stop"- screen use augmented reality through the camera. This is done by using the GART toolkit that I mention earlier in chapter 2.

GART is built around four different layers:

- **HeadingIndicator -** Draws an arrow to show the users heading.
- **OverheadMap** – Displays a bing map.
- **VideoPreview** – A rectangle that displays video from the camera.
- **WorldView** – This is the augmented reality layer and the layer that display a virtual world in 3d space. It then applies matrix math to keep the virtual world align with what's seen through the camera.

In my project I used two of the layers – VideoPreview and WorldView. I did not find any good reason for the HeadingIndicator layer and the OverheadMap had some performance problem. The OverheadMap map did also not have any functionality that separated it from a regular map control so I had no problems to not use it (especially with Windows phone 8 that used the new Nokia map).

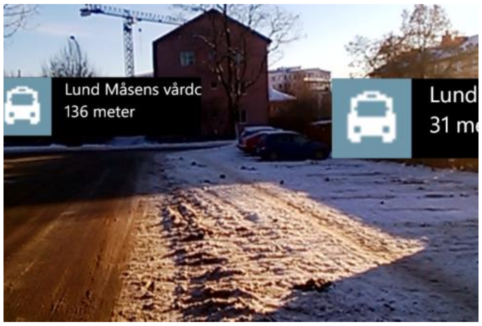


Figure 35 - Augmented reality with GART

The WorldView layer was easy to start work with and the only thing it needed was a list with AR-item. AR-items contains the location (in latitude/longitudes) and content. The content can be anything from a text to an image. In my application the content is a rectangle, imagebox and two textboxes (see figure 8). The two textboxes contains the name and the distance to the bus stop.

GART was originally made for Windows phone 7 and therefor had to be converted to work with Windows phone 8. The conversion did not work flawlessly as there was a couple of frameworks that GART used that was changed or didn't exists anymore. It resulted in a long delay and sometimes it didn't work at all.

4.6 TESTING AND PERFORMANCE

The last step was to test and check performance. To upload an application to the app store it need to meet a couple of requirements. For example it needed to start in less than x seconds and have an average response time for y seconds. It also have a limit for how much memory the application can use.

Testing

To test my application I use the built-in tools in visual studio together with exploration testing. With the tools I can try different resolutions, change from landscape to portrait and test with low or no internet connection.

The first test I did was with low or no internet connection. It worked fine with low connection but I had some problems with no connection as I catched connection errors but didn't give any feedback to the user. So I had to make sure that every time I got a timeout error I send it back to the interface that show a pop up with the error message.

The resolution did not make any big difference and I did not have to change anything for it to work fine. But the landscape view made most of the pages unusable so I disable that kind of view.

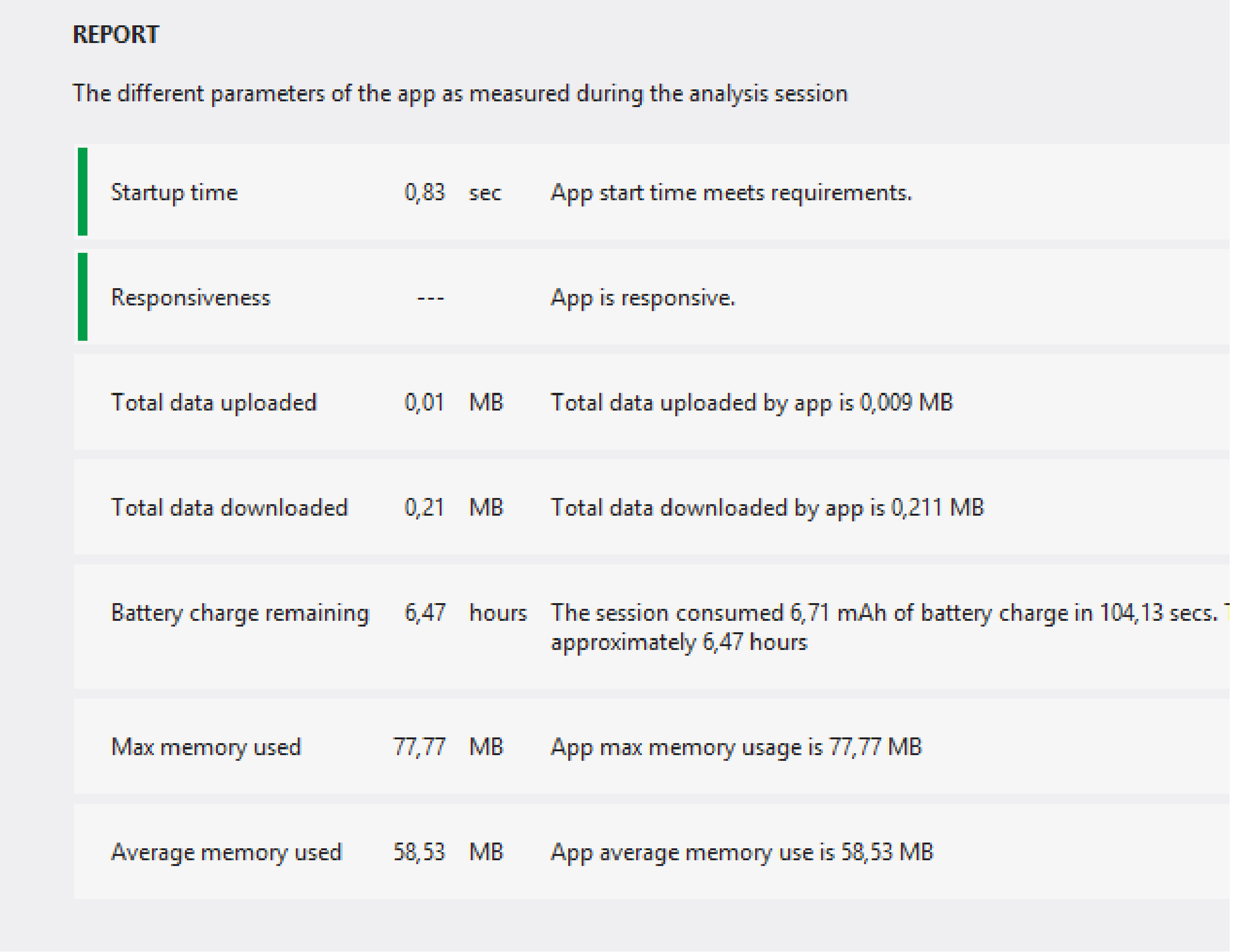
Other test I did was for example to click “clos bus stops” when there were no close bus stops or pick a point in the middle of the ocean. This gave back a result with zero bus stops and I had to handle it by showing the user a message box that there was no results.

The last test I did was with the code analysis. What is does is that It check though the code and if I break any of the coding and design standards. Most of the result was design problems, but it also gave some hints on where I could expect bugs. One of them where on all pages that used my data loading class (that get all data from Skånetrafiken) and told me that I forgot to call the disposable method. This method will remove all listeners and let the garbage collector later delete these items. If I would forget to call this method they would still be listening when I closed the page and later maybe crash the application.

Performance

To test the performance of a Windows phone application you can use visual studios performance analyzer. Performance analyzer is easy to start work with and you just start your application, use it for a while and then stop it. Performance analyzer will then collect all the necessary data - as startup time, response time, memory management and errors.

My application managed to pass all tests and I got these results:



REPORT

The different parameters of the app as measured during the analysis session

Startup time	0,83	sec	App start time meets requirements.
Responsiveness	---		App is responsive.
Total data uploaded	0,01	MB	Total data uploaded by app is 0,009 MB
Total data downloaded	0,21	MB	Total data downloaded by app is 0,211 MB
Battery charge remaining	6,47	hours	The session consumed 6,71 mAh of battery charge in 104,13 secs. T approximately 6,47 hours
Max memory used	77,77	MB	App max memory usage is 77,77 MB
Average memory used	58,53	MB	App average memory use is 58,53 MB

Figure 36 - Performance report

DISCUSSION

Discussion

In this chapter I will describe my thoughts about the result, what I think could be better and feature work that could be done.

THE APPLICATION

In chapter 3.3 I wrote all the functionality I wanted in my application and I manage to implement everything from prio 1 and 2. The only thing I missed from prio 3 was sms -ticket.

The application ended up as expected, but it may not have been as groundbreaking as I wanted it to be. The augmented reality functionality ended up being smooth with Windows phone 7, but had some problems with Windows phone 8. The problems is with GART and can't be fixed till they decide to update the framework.

But the standard functionality was good and I did them a bit different than the other applications I tested. The application did also pass all performance test and can be uploaded to the marketplace. Something that may be done in the future.

METHODS

If I were to given the opportunity to work with this thesis again I would change a couple of things in my process.

- I would try to work harder with low-fi prototyping and work more with ordinary pen and paper. Microsoft Expression blend was a good software but it felt that it didn't give the freedom and speed as I wanted and if I could re-do it I would start with pen and paper, then go over to Microsoft Expression blend.
- I would take more inputs from others and have it better organized. This time I talked with others about the application and checked for feedback but it was mostly spontaneous. It would have been better with a focus group or a survey.
- I would start with a requirement document to know exactly which functions I wanted and stay with those. It would help me later with testing but also my developing process. I would then be able to work more test-driven. My functionality list in chapter 3.2 was good but it would have been better with detailed requirements.

FUTURE WORK

There is still a lot that can be done to make it a better application:

- More social interaction with comments, notes about delays.
- Implement the sms-ticket functionality.

- Work more with notifications. There could be notifications for a lot of things, for example bus delays and bus getting close.
- Work with Windows phone 8 start screen. With Windows 8 you can put real-time information on the start screen and I think there is a lot of things that may fit for that. One of them is for example the bus schedule. The user could mark a bus stop as their main stop and the bus schedule would then be updated on their start screen.
- More work with augmented reality, for example work more with marker-based technologies.
- Put it on Microsoft marketplace for more feedback. As it passed the performance test it's not that much that need to be done for it to happen.
- Make sure that GART work correctly with Windows phone 8.

IN RELATION TO STUDIES

This project allowed me to use a lot of knowledge that I gained from previous courses. I could use the knowledge from my "Software Engineering Process – Methodology" course to plan my process. I used a lot of information from my design courses to design and think about the user interaction and experience. My programming courses made it easy for me to design and implement the application and the Software Testing course helped me structure my testing.

It was also the course "Virtual Reality in Theory and Practice" who taught me about augmented reality and gave me the idea for this master's thesis.

CONCLUSION

Conclusion

My original goal was to use augmented reality in a new and fun way and even if I did I don't think I made it good enough to be used as an everyday thing. It's still very situational and more fun than useful. But the technology is there to use.

In the beginning of this thesis I set up some goals I wanted to meet and here are a summary and answer of them:

To investigate what's possible to do with augmented reality and current technology.

My background and state-of-the-art research was good and I think I found most of the technology that could be used for Windows phone at the moment. Some of them are still not ported to Windows phone 8 but can be used with some modifications.

To investigate current applications for public transport.

At the start I felt that my check was good enough but as I progress I think I could do a better job at this. I found more functionality and applications when I got further in to the project and that made some of my ideas not as new as I thought (some applications did already use address/GPS in fun ways). If I could do it again I would borrow an iPhone/Android phone from a friend and download all the applications I could find to do more research.

To investigate how augmented reality can be used to improve the user experience in public transport.

I had a couple of ideas for this and I ended up with my sensor based solutions where I tracked the bus stops through the GPS and camera.

As I said earlier, I think I found this as a new way to use public transport but it was still not in a very useful way. I think I could have explored the possibilities with marked-based (SLART and Goblin XNA) technologies more but I still don't have any good ideas for it.

To design and implement an application for Windows phone.

The design and implementation of my application went mostly good. I think I could have changed a couple of things in my process (as I mention in the discussion) but the application is done and can be used on the market. I also met all the requirements and performance test to upload it on an app store if I wanted.

References

[1] SLARToolkit, http://slartoolkit.codeplex.com/ [Accessed 2012-09-11]

[2] NyARToolkit, http://nyatla.jp/nyartoolkit/wp/ [Accessed 2012-09-11]

[3] ARToolkit, http://www.hitl.washington.edu/artoolkit/ [Accessed 2012-09-11]

[4] GART, http://gart.codeplex.com/ [Accessed 2012-09-11]

[5] Goblin XNA, http://goblinxna.codeplex.com/ [Accessed 2012-09-11]

[6] XNA, http://xbox.create.msdn.com/en-US [Accessed 2012-09-11]

[7] ALVAR, http://virtual.vtt.fi/virtual/proj2/multimedia/alvar.html [Accessed 2012-09-11]

[8] MobiTime, http://itunes.apple.com/se/app/mobitime/id372372299?mt=8 [Accessed 2012-09-28]

[9] London Tube, http://www.londontubeiphone.com/ [Accessed 2012-09-28]

[10] Metaio, http://www.metaio.com/ [Accessed 2012-09-28]

[11] Catch the bus http://catchthebusapp.com/ [Accessed 2012-09-28]

[12] Microsoft expression blend http://www.microsoft.com/expression/products/Blend_Overview.aspx [Accessed 2012-10-29]

[13] Designing the user interface, 4th edition, Pearson Education, Ben Schneiderman, Catherine Plaisant,

[14] User experience design guidelines for windows phone, http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202915(v=vs.92).aspx [Accessed 2012-11-05]

[15] Pivot page, http://msdn.microsoft.com/en-us/library/hh202919(v=vs.92).aspx [Accessed 2012-11-05]

[16] Skånetrafiken API, http://www.labs.skanetrafiken.se/ [Accessed 2012-11-15]

[17] SOAP Specifikation, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ [Accessed 2012-11-16]

[18] WG84 to RT90 Converterter, http://blog.sallarp.com [Accessed 2012-11-18]

[19] Emerging Technologies of Augmented Reality: Interfaces and Design, 2007, Michael Haller, Mark Billinghurst, Bruce H. Thomas

[20] SONY – Augmented reality, http://www.eat.lth.se/fileadmin/certec/Augmented_Reality.pdf , [Accessed 2012-12-28]

REFERENCES

[21] Eniro API, http://api.eniro.com/ [Accessed 2012-12-28]

[22] About Silverlight, http://www.microsoft.com/silverlight/what-is-silverlight/ [Accessed 2013-01-03]

[23] Noka map API, http://www.developer.nokia.com/Resources/Library/Lumia/#!guide-to-the-maps.html [Accessed 2013-01-05]

[24] Årsrapport 2011, Kollektivtrafikbaroometern http://www.svenskkollektivtrafik.se/Global/Fakta%20och%20publikationer/publikationer/KOLBAR%20Ipsos_Rapport_Hel%C3%A5r%202011_rev%20120207.pdf [Accessed 2012-10-10]

[25] Junaio, http://www.junaio.com/ [Accessed 2013-01-10]

[26] GNU public license, http://www.gnu.org/licenses/gpl.html [Accessed 2013-01-15]

[27] Open source Initiative, http://opensource.org/osd-annotated [Accessed 2013-01-15]

[28] Skånetrafiken, http://www.skanetrafiken.se [Accessed 2013-01-15]

[29] XAML, http://msdn.microsoft.com/en-us/library/ms752059.aspx [Accessed 2012-11-10]