

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Szczurková Jana

Studijní program: Otevřená informatika  
Obor: Softwarové systémy

Název tématu: Platforma pro podporu práce operátů tele-asistenčních center pro navigaci zrakově postižených

Pokyny pro vypracování:

Analyzujte práci operátorů telefonické linky Navigačního centra SONS. Použijte metodu pozorování a rozhovor, vypracujte scénáře a pro vybrané scénáře vytvořte storyboardy. Vypracujte rešerši vhodných vizualizačních technik. Vytvořte papírové mockupy a získejte zpětnou vazbu od operátorů metodou design probe. Zpětnou vazbu operátorů reflektujte do papírového prototypu. S papírovým prototypem proveďte a vyhodnoťte test použitelnosti alespoň se třemi účastníky. Na základě výsledků testu použitelnosti papírového prototypu, ve vhodném nástroji vytvořte lo-fi prototyp, proveďte a vyhodnoťte test použitelnosti alespoň se třemi účastníky. Na základě výsledků testu použitelnosti lo-fi prototypu, vytvořte hi-fi prototyp (použijte techniku 'throw-away' prototyping, implementujte v HTML a JavaScript). Popište interakční, vizuální a user experience design, definujte funkční požadavky. Proveďte a vyhodnoťte test použitelnosti hi-fi prototypu alespoň s pěti účastníky.

Seznam odborné literatury:

- [1] Spence, Robert. Information visualization. Vol. 1. New York: Addison-Wesley, 2001.
- [2] Bertini, Enrico, Maurizio Rigamonti, and Denis Lalanne. "Extended excentric labeling." Computer Graphics Forum. Vol. 28. No. 3. Blackwell Publishing Ltd, 2009.
- [3] Balata, Jan, et al. "Automatically generated landmark-enhanced navigation instructions for blind pedestrians." Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on. IEEE, 2016.
- [4] Eick, S. C., Joseph L. Steffen, and Eric E. Sumner. "Seesoft-a tool for visualizing line oriented software statistics." IEEE Transactions on Software Engineering 18.11 (1992): 957-968.
- [5] Agrawala, Maneesh, and Chris Stolte. "Rendering effective route maps: improving usability through generalization." Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001.
- [6] Liu, He, et al. "Visual analysis of route diversity." Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on. IEEE, 2011.

Vedoucí: Ing. Jan Balata

Platnost zadání do konce letního semestru 2017/2018



prof. Dr. Michal Pěchouček, MSc.

vedoucí katedry

prof. Ing. Pavel Ripka, CSc.

děkan

V Praze dne 16.1.2017



bachelor's thesis

**A platform supporting the work of operators  
of tele-assistance centers for navigation of  
visually impaired people**

*Jana Szczurková*



May 2017

Ing. Jan Balata

Czech Technical University in Prague  
Faculty of Electrical Engineering, Department of Computer  
Graphics and Interaction



## **Acknowledgement**

I would like to thank supervisor of this project Mr. Ing. Jan Balata for providing many valuable advice and reminders, without which this work could not have arisen. I also would like to thank all of participants who helped me get useful feedback on prototypes and big thanks also belong to my family for their support and patience.

## **Declaration**

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

## **Abstract**

Cílem této práce je navrhnout, vytvořit a otestovat prototypy platformy pro podporu práce operátorů tele-asistenčních center pro zrakově postižené, která jim má usnadnit práci a pomoci vyřešit problémy objevené při pozorování v Navigačním centru SONS. Zaměřili jsem se především na optimalizaci pracovního postupu operátora při vytváření detailních popisů tras a umístění většiny zdrojů informací do jednoho přístupového bodu.

V průběhu práce na projektu jsme definovali případy užití, podle nichž jsem postupně vytvořili storyboardy, papírové návrhy a low-fidelity prototypy, které byly průběžně testovány a konzultovány s operátorem a dalšími odborníky.

V závěrečné fázi projektu byl naimplementován high-fidelity prototyp v podobě webové aplikace demonstrující většinu funkcionalit platformy. Tato aplikace a výsledky z jejího testování i testování jejích předchůdců mohou posloužit jako zdroj dat v dalších fázích projektu.

## **Klíčová slova**

itinerář; plánování tras; zrakově postižení; tele-asistence; vizualizace

## **Abstract**

The aim of this thesis is to design, create and test a prototype of a platform to support the work of operators of tele-assistance centers for visually impaired people, which will simplify their work and help to solve the problems discovered during the observation in the SONS Navigation Center. We primarily focused on optimizing the operator's workflow when creating detailed route descriptions and aggregating most of the information sources to one access point.

During the project, we have defined use cases, according to which we gradually created storyboards, paper designs and low-fidelity prototypes, which were continuously tested and consulted with the operator and other experts.

In the final phase of the project, a high-fidelity prototype has been implemented in the form of a web application that demonstrates most of the functionalities of the platform. This application, its test result and test results of its predecessors can serve as a data source in the next phase of the project.

## **Keywords**

itinerary; route planning; visually impaired; tele-assistance; visualisation

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Problem specification</b>	<b>2</b>
2.1	Motivation . . . . .	2
2.2	Problem specification . . . . .	2
2.2.1	Navigational Centre for the Blind . . . . .	2
	Provided services . . . . .	2
2.2.2	Observation . . . . .	2
	First phone call . . . . .	3
	Second phone call . . . . .	3
	Itinerary creation . . . . .	3
2.2.3	Interview summary . . . . .	4
2.2.4	Problems . . . . .	5
	Itineraries . . . . .	5
	Remote Assistant . . . . .	5
2.3	Target group and its requirements . . . . .	5
2.4	Method . . . . .	5
2.4.1	Extended excentric labeling . . . . .	6
	Reason of use . . . . .	6
2.4.2	Automatically generated landmark-enhanced navigation instructions for blind pedestrians . . . . .	6
	Reason of use . . . . .	6
2.4.3	Route Visualization Using Detail Lenses . . . . .	6
	Reason of use . . . . .	6
2.4.4	Automated Landmark Identification . . . . .	7
	Reason of use . . . . .	7
<b>3</b>	<b>Design</b>	<b>8</b>
3.1	Sketch . . . . .	8
3.2	Use cases . . . . .	9
3.2.1	Existing itineraries . . . . .	9
3.2.2	New itineraries . . . . .	9
3.2.3	Remote Assistant . . . . .	11
3.3	Storyboard . . . . .	11
3.4	Paper mockup . . . . .	11
3.4.1	Description . . . . .	11
	Interface description . . . . .	11
3.4.2	Design probe . . . . .	14
	Design probe definition . . . . .	14
	Evaluation . . . . .	14
3.5	Low-fidelity prototypes . . . . .	14
3.5.1	Paper prototype . . . . .	15
	Prototype description . . . . .	15
	Interface description . . . . .	15
	Evaluation . . . . .	16
3.5.2	Electronic prototype . . . . .	16
	Description . . . . .	16



	Balsamiq Mockups . . . . .	16
	Changes in the interface . . . . .	17
	Evaluation . . . . .	19
3.6	High-fidelity prototype . . . . .	19
3.6.1	Changes in the interface . . . . .	20
3.6.2	Functional requirements . . . . .	20
3.6.3	Interaction design . . . . .	21
	Words . . . . .	21
	Visual representations . . . . .	21
	Physical objects/space . . . . .	23
	Behavior . . . . .	23
3.6.4	Visual design . . . . .	23
3.6.5	User experience design . . . . .	24
3.6.6	Evaluation . . . . .	24
<b>4</b>	<b>Implementation</b>	<b>25</b>
4.1	HTML5 . . . . .	25
4.2	JavaScript . . . . .	25
4.2.1	Classes . . . . .	25
	Request class . . . . .	25
	Itinerary class . . . . .	26
	Stage class . . . . .	26
4.2.2	Program logic . . . . .	26
4.2.3	Used APIs . . . . .	26
	Mapy.cz API . . . . .	26
	Google Custom Search API . . . . .	27
	Other APIs . . . . .	27
4.3	CSS . . . . .	27
<b>5</b>	<b>Evaluation</b>	<b>28</b>
5.1	Evaluation of paper low-fidelity prototype . . . . .	28
5.1.1	Experiment description . . . . .	28
	Pre-test questionnaire . . . . .	28
	Post-test questionnaire . . . . .	28
5.1.2	Experiment procedure . . . . .	28
	Participant 1 . . . . .	31
	Participant 2 . . . . .	31
	Participant 3 . . . . .	32
5.1.3	Experiment findings . . . . .	32
	Suggestions for new features . . . . .	32
	Changes in design . . . . .	33
	Suggestions for changes to existing features . . . . .	33
5.2	Evaluation of electronic low-fidelity prototype . . . . .	33
5.2.1	Experiment description . . . . .	33
	Pre-test questionnaire . . . . .	34
	Post-test questionnaire . . . . .	34
5.2.2	Experiment procedure . . . . .	36
	Participant 1 . . . . .	36
	Participant 2 . . . . .	37
	Participant 3 . . . . .	37

5.2.3	Experiment findings and discussion . . . . .	37
	Suggestions for new features . . . . .	37
	Changes in design . . . . .	38
	Suggestions for changes to existing features . . . . .	38
5.3	Evaluation of high-fidelity prototype . . . . .	39
5.3.1	Experiment description . . . . .	39
	Pre-test questionnaire . . . . .	39
	Post-test questionnaire . . . . .	39
5.3.2	Experiment procedure . . . . .	39
	Participant 1 . . . . .	40
	Participant 2 . . . . .	40
	Participant 3 . . . . .	40
	Participant 4 . . . . .	42
	Participant 5 . . . . .	42
5.3.3	Experiment findings and discussion . . . . .	42
	Changes in design . . . . .	42
	Suggestions for changes to existing features . . . . .	43
5.3.4	Recommendations for the next phase . . . . .	43
<b>6</b>	<b>Conclusion</b>	<b>44</b>
6.1	Evaluating the achievement of the project objectives . . . . .	44
6.2	Recommendation to other phases of the project . . . . .	44
	<b>Appendices</b>	
<b>A</b>	<b>Paper mockup</b>	<b>45</b>
A.1	Layer of noise . . . . .	45
A.2	Layer of slope . . . . .	45
A.3	Layer of existing itineraries . . . . .	47
A.4	Coloring the route according to slope of its segments . . . . .	47
<b>B</b>	<b>Paper low-fidelity prototype</b>	<b>48</b>
	Fig. 19 content . . . . .	48
	Fig. 20 content . . . . .	49
<b>C</b>	<b>Test scenarios</b>	<b>50</b>
C.1	Test scenario of paper low-fidelity prototype . . . . .	50
C.2	Test scenario of electronic low-fidelity prototype . . . . .	51
C.3	Test scenario of high-fidelity prototype . . . . .	52
	<b>Bibliography</b>	<b>53</b>

# List of Figures

1	Picture of operator writing new itinerary taken during observation. . . .	4
2	Sketch of possible functions of the platform. . . . .	8
3	Storyboard showing an example of using several use cases. . . . .	12
4	Mockup with opened street view . . . . .	13
5	Mockup, which was used as a pattern for a paper prototype . . . . .	15
6	Electronic low-fidelity prototype . . . . .	17
7	Electronic low-fidelity prototype: Incoming call . . . . .	17
8	Electronic low-fidelity prototype: Searching result . . . . .	18
9	Electronic low-fidelity prototype: Segment buttons . . . . .	18
10	High-fidelity prototype . . . . .	22
11	Visual representation of route modification . . . . .	22
12	An example of a modal window in high-fidelity prototype . . . . .	23
13	APIs in high-fidelity prototype . . . . .	26
14	Picture taken during a test of paper prototype. . . . .	31
15	Mockup: Layer of noise . . . . .	45
16	Mockup: Layer of slope . . . . .	46
17	Mockup: Layer of existing itineraries . . . . .	46
18	Mockup: Coloring the route according to slope of its segments . . . . .	47
19	Paper prototype interface . . . . .	48
20	Paper prototype layers . . . . .	49

# List of Tables

1	Paper prototype pre-test questionnaire . . . . .	29
2	Paper prototype post-test questionnaire . . . . .	30
3	Electronic prototype pre-test questionnaire . . . . .	34
4	Electronic prototype post-test questionnaire . . . . .	35
5	High-fidelity prototype pre-test questionnaire . . . . .	40
6	High-fidelity prototype post-test questionnaire . . . . .	41

## Abbreviations

API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
MS	Microsoft
pdf	Portable Document Format
QA	Quality Assurance
RA	Remote Assistant
SONS	Sjednocená organizace nevidomých a slabozrakých ČR (Czech Blind United)
SW	Software
UI	User Interface



# 1 Introduction

The work of the operator of a tele-assistance center for visually impaired people includes searching information on the internet or public transport connections and preparation of textual itineraries of planned routes. Usually, the operator opens several browser windows with different services and map providers at the same time to quickly find desired information. This task demands high cognitive load and increases a stress level. Our goal is to ease these stressful situations through aggregating several information sources in one access point and optimizing, operator's workflow. We aim to achieve this goal by introducing a sophisticated set of visualization tools laid over a map, and several techniques which utilize previously created descriptions of routes and provide partial automatic generation of the new routes.

## 2 Problem specification

### 2.1 Motivation

The aim of this work is to facilitate the work of operators of tele-assistance centers for navigation of visually impaired people, who are mainly here to help their clients to move independently of the seer and become their eyes.

There are days when no one calls, but then there may be situations when clients requests are accumulating, and the operator does not know what to do sooner, which demands high cognitive load and increases a stress level. Our goal is to ease these stressful situations through aggregating several information sources in one access point and optimizing, operator's workflow. We aim to achieve this goal by introducing a sophisticated set of visualization tools laid over a map, and several techniques which utilize previously created descriptions of routes and provide partial automatic generation of the new routes.

### 2.2 Problem specification

#### 2.2.1 Navigational Centre for the Blind

Navigational Centre for the Blind[1] is a section of the Digitisation and Technical Support Department run by Czech Blind United[2]. The center is available for all visually handicapped people in the Czech Republic since 2007, and its main purpose is to help visually impaired people to travel independently on a sighted guide only by using their cane or their guide dog even in places they are not familiar with. From the start, it carried out about 79 thousand requests, but since 2013 their services are not for free, so the number of clients decreased.

#### Provided services

List of provided services of Navigational Centre according to their website[1].

1. Searching public transport connections, telephone numbers or information about the searched object or the area.
2. Route planning and a creation of itineraries.
3. Satellite navigation.
4. Help and navigation via the Remote Assistant application.
5. Emergency help.

#### 2.2.2 Observation

Thanks to our co-operation with Navigational Centre[1] we were allowed to observe the work of the operator and to do an interview with him twice. The first session was



on 19th October 2016. It lasted about 3 hours and we saw two phone calls and one itinerary writing, so we got a lot of valuable information.

### **First phone call**

The client wanted to go in June to a summer cottage in Chrást nad Sázavou and needed to describe the route.

The operator opened the Seznam map and found the destination. He panned and zoomed up the map to look for a neighboring village, probably trying to find the most suitable way to the destination. The client told him which way he wants to go and the operator then described what is on the road and also mentioned the point of interest (shop). He described the path, materials of the entrances to the family houses along it and a metal gate at the destination using the Seznam street view.

### **Second phone call**

The client wanted to go to the Cultural Sokol House at Radlická.

The operator looked for the address of the Sokol House and how to get to it from Na Knížecí station. He opened a document with a description of an Anděl metro station, which is very detailed (about two A4 pages), mentioning bus numbers. Then he searched for the entrance to the Sokol House through the street view, describing the fence with a concrete curb and the columns of illumination in the way. There was a gate that seemed closed on the street view, so he didn't know if it was the entrance to the building. The opened street view occupied about 2/3 of the screen, but map only 1/3.

The client also needed an address for the Monastery of Agnes. The operator tried to find it through the Google search, then he set the address on the map, but the client didn't agree (probably he didn't know the correct name of the monastery). The operator mentioned points of interest in the neighborhood and then tried to find the monastery on Wikipedia.

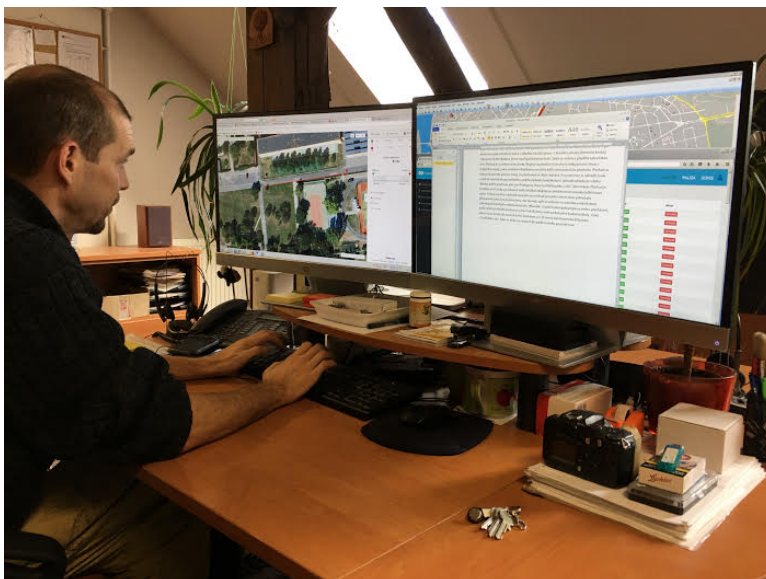
### **Itinerary creation**

The client wanted to go from the Olšanské Náměstí tram station to Chelčického 43.

The operator performed a quick survey. He typed the destination address and then examined the street view of the station. He measured the distance of pedestrian crossing and checked the embossed tiles. The client will pass four entrances with stairs. Then he will have to go to the right where the railing and the public lighting column are. The pedestrian crossing consists of 3 parts. The client will have to go by the railing. Streetview ended so the operator switched to Google Maps.

He started writing in the MS Office Word and formatted a title. He described how the client should orientate himself on the stop. After going out of the tram, he must go a bit to the right in the opposite direction of the tram's arrival. Then the operator indicated the number of road lanes and measured the distance from the crossing to the corner of the street. He checked through the street view stairs that hit the pavement. Counts total of 4. On the last stairs, the embossed tiles of the pedestrian crossing appear on the left. The crossing is long but has a guide groove and is divided into two parts with an isle. Again he mentioned the number of road lanes. "I'm going to get back to it, now I'm creating a skeleton," said the operator.

He described how to evade a lamp and found embossed tiles after the crossing. The pedestrian crossing has three sections: 1 lane, slightly left the isle, four lanes (2 on the



**Figure 1** Picture of operator writing new itinerary taken during observation.

right, two on the left) and the second isle. The route goes on through the park because it's straight. He didn't know whether to lead the client to the right (problem finding the turnoff) or to the left (the pavement is expanding). Finally, he was looking for entrance to the building (school), but it was not visible on the street view.

He worked with a Seznam map, which was very slight because the biggest part of the screen covered the street view. On the other monitor, he had MS Office Word (as can be seen on Fig. 1). Nomenclature of the itinerary is not strict (path vs. pavement).

### 2.2.3 Interview summary

There is only one operator in the Navigational Center[1]. He has worked here about five years. To be able to hold this position he must know how to work with computer and also he had to try, how it is to be blind. Each day, he solves around three requests that come mainly from men and younger people from the outskirts of larger cities. In the first place is Prague. There are no complaints, but once he mistook the left and right sides.

Route description must be ordered at least two days in advance. At first, he looks if the itinerary does not exist. If so, check the correction of data at least through street view. The time spent working on the itinerary depends on its complexity. The most difficult routes include transfers to public transport with several exits. Such work may take a day. To create itinerary he uses street view where he tries to find a route without unnecessary jinking, but when the operator is not sure he goes to the field. Sometimes he creates different route variants. He writes everything in MS Office Word.

Approximately a half the people who downloaded Remote Assistant[3] application uses it. These are mainly men and younger people. People tell him about their needs directly on the phone. They most often need to read a door sign, name tags, business cards, use it for computer problems, or everyday things such as switching on a washing machine, consulting clothes, etc. The only disadvantage is when using Remote Assistant[3] while navigating because it only shows one current client position.

### 2.2.4 Problems

According to the results of interview and observation, we discovered some problems in operator's workflow that can be solved.

#### Itineraries

- When the operator wants to find existing itinerary, he has to remember where they are located in the computer and navigate through many of folders to found them.
- When he finds the wanted itinerary, he must check, if it is still actual.
- Writing new itinerary includes a lot of parameters:
  - Names of streets and public transport stops
  - Location of public transport stops.
  - Landmarks that will help the visually impaired person to navigate (material of pavement, embossed tiles, railing, ...)
  - Approximate distance of route segments
  - Cross the street information (direction of cars, the number of streams, ...)
  - Slope of street
  - Noise
  - Where is the entrance to building
- All of this information he must search on maps, street views and other sources on multiple tabs in web browser.

#### Remote Assistant

- The application shows only 1 actual position of the client. We don't know from which direction he came and we can't save the route.
- The operator must find the position of the client in map provider manually.
- Detecting the direction of view (somebody has a mobile phone to his ear, somebody ahead).

## 2.3 Target group and its requirements

The target group is operators and staff of tele-assistance centers and other institutions involved in navigating visually impaired people. They navigate the clients via phone or provide them textual descriptions of routes.

Operators, unlike their clients, are not limited, so the resulting interface may not meet any special requirements.

The main requirement is that the application must be executable on a regular computer and must include most of the sources of information with which the operators are accustomed working.

## 2.4 Method

This section introduces some methods that could be used in the platform.

### **2.4.1 Extended exentric labeling**

The article[4] presents an extension to the labeling technique, which dynamically shows labels around a movable lens. Each label is connected with the object through a line. Dynamic labeling treats the labels as dynamic objects that can appear and disappear according to some specific interactions, e. g. Mouse cursor position.

#### **Reason of use**

This technique can be used in the map layer of existing itineraries. It will show a map with highlighted routes which will correspond to each existing itinerary, so the operator can only look at this map if there is some itinerary which can be reused in a new one. Each route will include a label with a link to its text form, which will allow the operator to open it directly in the program, without unnecessary search.

### **2.4.2 Automatically generated landmark-enhanced navigation instructions for blind pedestrians**

In this article[5] is presented a way how to help visually impaired people to travel independently without sighted guide. This research group introduced an algorithm that can generate itineraries which are composed of navigation instructions for each segment of the route in natural language. It includes a system of landmarks, that can help visually impaired people to navigate because in existing navigation systems appropriate details and landmarks are missing.

#### **Reason of use**

This method, respectively the application based on it will be used to generate itinerary templates.

These templates will be gradually improved by segments of existing itineraries written by the operator.

### **2.4.3 Route Visualization Using Detail Lenses**

The article[6] describes a method which generates a printable version of a route map with detail views. It is primarily for drivers, but the operator can use it too. Their system highlights the route on the map and other parts are gray not to attract attention. Then they focus on lens placement strategy to let the map be still well-arranged. Each point of interest has its special marker and lens, that is placed around the map. The lens is also highlighted, and small arrow along the route shows the direction of travel.

#### **Reason of use**

We want to use this technique when the itinerary is generated. This itinerary will be composed of many numbered segments, and we want to display the number of the section on the route. It will help the operator to navigate between them and to join text with the particular part of the route on the map, so he can always see which part is modifying. Also when itinerary must be improved, the detailed lens may show magnified objects or shapes of crossing around the particular segment.

#### 2.4.4 Automated Landmark Identification

This method is described in the paper[7]. These researchers analyzed route descriptions written by blind individuals to find out some patterns thanks to them the landmarks in the text could be automatically identified.

Identifying landmark starts with analyzing phrases, where some landmark occurs. We must focus on words that accompany them. They are in most cases nouns like stairs, corner, pedestrian crossing, etc. Words that can reveal landmarks are verbs, terms of cardinal directions and terms of distance measurements and also prepositions. In the paper they defined nine types of phrases, that can reveal landmark.

These types are: (*landmark*, **keyword**)

1. Spatial Simple Transitive Phrase - e. g. "**next** to the *stairs*"
2. Spatial Compound Transitive Phrase - e. g. "**in front of** the *water fountain*"
3. Spatial Intransitive Phrase - e. g. "The *stairs* are **afterward**"
4. Spatial Angle Phrase - e. g. "The *door* at **angle** to you"
5. Spatial Distance Phrase - e. g. "It is about **10 meters** to the *pedestrian crossing*"
6. Biased Part Phrase - e. g. "Go to the **top** of the *stairs*"
7. Egocentric Reference Phrase - e. g. "Tell **your guide dog** to find *stairs*"
8. Verb Phrase - e. g. "**Enter** the *metro station*"
9. Phrase Secondary Landmark Phrase - e. g. "You will come right to the **doors of** the *Front Office*"

#### Reason of use

This method can be used to invent an algorithm to detect mostly crossings in existing itineraries to split them into reusable segments, which can be shown on a map.

We won't use all of the defined phrases types for our purposes because the Czech language has shorter phrases. We will focus mostly on verbs, prepositions, and distances, which occurs in itineraries a lot.

# 3 Design

According to the results of the observation in the Navigational Center[1], we suggested some possible solutions optimizing the operator's workflow. In this chapter, we are gradually introducing them from the first sketch to the high-fidelity prototype.

## 3.1 Sketch

In the beginning, we have brainstormed some features that could help the operator with his work and which the platform should offer. We transferred these thoughts to paper in the form of the first sketch shown on Fig. 2.

The sketch is divided into three main parts. In the upper half of the left side, you can see ideas about existing routes, such as a list of itineraries with a search, a map of itineraries on which are highlighted already created routes with labels with reference to their text form, marking of itineraries according to their age, etc.

The top half of the right side is related to the Remote Assistant application. These functions are to move the client location from the Remote Assistant to the map provider and to set a direction of view of the client on the map.

Other features are related to creating, editing, and further working with routes in map provider, street view, and in their text formats.

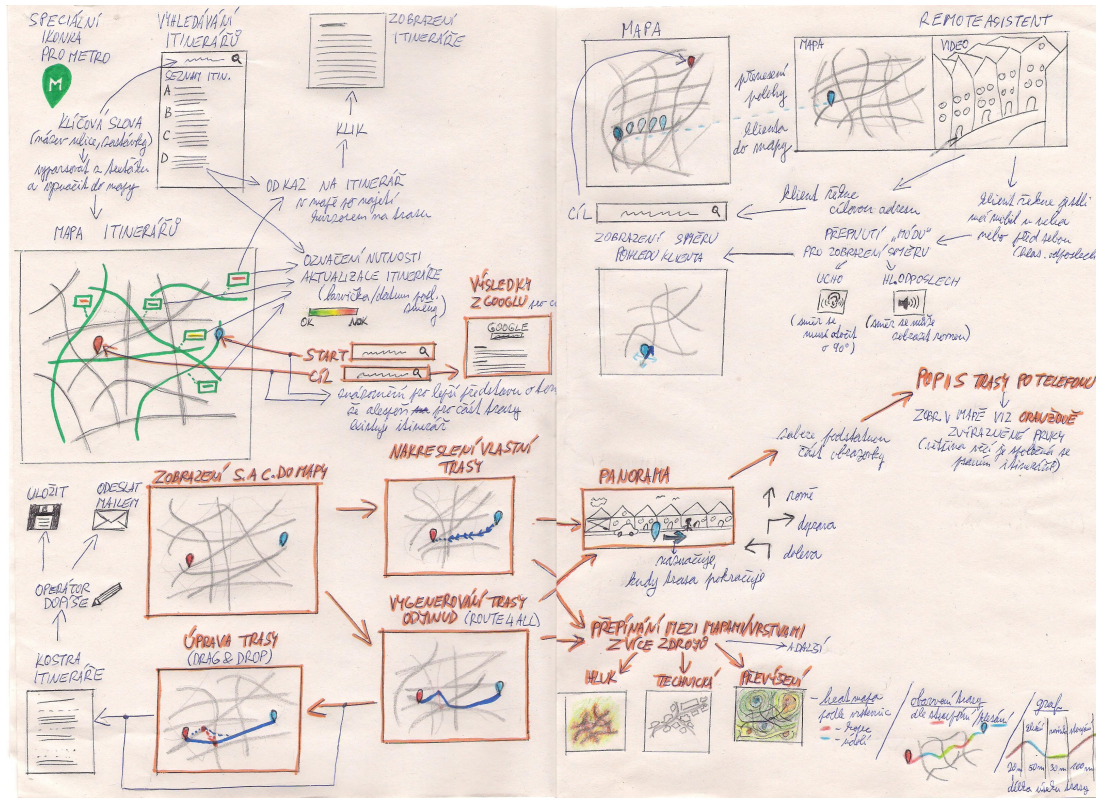


Figure 2 Sketch of possible functions of the platform.

## 3.2 Use cases

After the sketch had been created, the writing and extension of the use cases which it presented, followed. For better orientation, we divided them into three groups depending on what they are related to:

### 3.2.1 Existing itineraries

1. Map layer of existing itineraries.

*It will show a map with highlighted routes which will correspond each existing itinerary, so the operator can look at this map if there is some itinerary which can be used in the new one.*

2. Itineraries will be marked according to the date of their last modification.

*It can be colored by a color of its last modification. E. g. Green for a new one (last modification date is recent), red for ancient one.*

3. Itineraries of metro stations will have their label on a map.

*Metro stations can not be displayed as routes, so they must have their marker.*

4. When the mouse cursor is over some itinerary route, the label of the itinerary will show.

*On the label will be a name of the itinerary and date of last modification.*

5. Clicking on the label will open the itinerary directly in the software.

*With this feature, it won't be necessary to search the itinerary manually and to open it directly in the interface will reduce the number of opened windows with information sources.*

6. List of existing itineraries (Fig. 3b).

*Already written itineraries will be in one easily accessible place, in a list, so there will be no need of knowing their exact location in database/folder.*

7. Full-text search in list of existing itineraries (Fig. 3b).

*In the future, the number of existing itineraries will increase and therefore their manual search in the list will become more and more complicated. However, this problem will solve full-text search, which will make it possible to search for routes that only cross the searched section.*

### 3.2.2 New itineraries

1. Searching start and destination location (Fig. 3c).

*This feature is necessary when a new itinerary is created because you need to know both the start and end position of the client. These locations will be marked on the map and will also serve as parameters for the generated route.*

2. Generating route via Naviterier[8] (Fig. 3d).

*Naviterier Routeplanner[8] will generate a route, that will be showed on a map.*

3. Generating itinerary template via Naviterier[8] (Fig. 3d).

*Naviterier Routeplanner[8] will generate a template of an itinerary, that will be gradually improved by existing more detailed itineraries written by an operator. It will help with reuse of existing itineraries, so operator won't have to rewrite some segment of route multiple times.*

4. Drawing route in a map manually.

*It may happen that the Naviterier[8] will not have covered the required route and the operator will have to create it. This will be used to draw your route on the map.*

5. Route modifying.

*If the operator do not like the generated route, he will be able to change it on the map.*

6. The route will be shown in map provider.

*It's necessary for work with the route and its street view.*

7. Street view will be set to start position.

*The operator won't need to set up street view manually and then will be able to start going through the route immediately.*

8. Arrow in street view will show the direction of the route.

*The arrow will help the operator navigate the route directly in the street view, he is working with the most, and he did not need to look too much on the map.*

9. Generated itinerary will be in text processor directly in the software (Fig. 3a).

*Opening it directly in the interface will reduce the number of opened windows with information sources.*

10. Switching between map layers (existing itineraries, noise, technical map, etc.).

*Map of the slope will be visualized as a map of contour lines colored by the color of its slope e.g. red for ascending, blue for descending. Another possible and maybe better solution of visualizing slope can be coloring only a particular route with a graph of individual segments of the route with the category of its slope and length of it.*

11. Saving not completed work (itinerary, position on a map, ...).

*There may be situations where the operator will have an unfinished itinerary, and a higher priority request will come. In order not to lose his not completed work, he will be able to save it and leave it for later.*

12. Saving completed itinerary into database/folder.

*The itinerary will appear in the list and the layer of existing itineraries and will be able to be re-used.*

13. Sending itinerary via email.

*To avoid unnecessary copying of the text to the email, a button for sending it would be available.*



### 3.2.3 Remote Assistant

1. Saving positions from Remote Assistant[3] (Fig. 3g).

*Client's positions will be transferred to a map provider automatically. He did it by now manually, which waes quite tedious and slow.*

2. Saving created route.

*While navigating the client over the phone, it is possible to encounter a problem on a route that would be good to record. So the operator will be able to mark the problem and save the created itinerary.*

3. An arrow shows the direction of view of the client (Fig. 3g - Fig. 3i).

*Some people have a phone near their ear, some ahead, so it's sometimes hard to adapt a look at the map and street view according to the client's video. Arrow to show the direction of view of the client would solve the problem with orientation in a map.*

## 3.3 Storyboard

After writing use cases followed a presentation of some of them in the form of storyboard (Fig. 3), which has outlined their use in the operation and showed their advantages. It was created using Pixton Comics[9].

The storyboard demonstrates how some of the operator's steps will look like when creating a new itinerary, and reveals some of the functionality of the Remote Assistant[3] call handling.

## 3.4 Paper mockup

Use cases were defined, so we had to start with the design of the user interface itself. The first phase was the creation of a paper mockup (Fig. 4), which we then consulted with the operator using the design probe method.

### 3.4.1 Description

We created the design from the main feature of the platform, which is the aggregation of most information sources at one access point. This information includes existing itineraries, a map, street view, etc.

The mockup was an A4-sized sturdy paper on which an interface was drawn, and the layers could be embedded. These layers were a map of Prague 2, a street view, map layer of existing itineraries, noise and slope of streets, and a map showing the route that was dyed according to its slope. We also created markers for the start, destination, metro station itineraries, and street view direction arrow. In the Fig. 4 you can see the map and street view layers and the markers for the start, destination, and street view arrow. Other mockup variants can be found in the appendix A.

### Interface description

In this section we will describe individual parts of the interface.

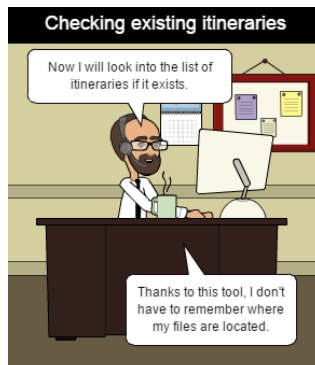
Features are described from top to bottom.

1. The mockup is created as a program window. The buttons on the top are:



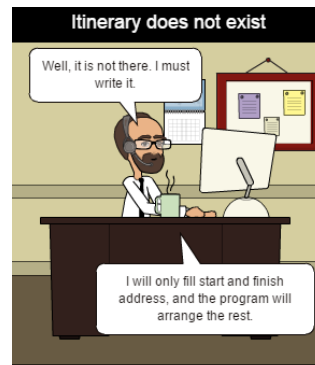
When there is a new request, the operator only clicks on one button, and all of the needed tools will be displayed on the screen.

a) All tools in one place



At first, the operator has to check if itinerary does not exist. He will only look into list of existing itineraries

b) List of existing itineraries



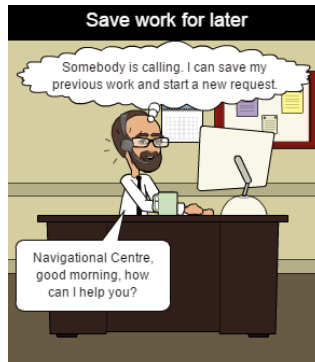
If it does not exist, the operator has to write it. He only has to fill start and finish address, and the program will generate its route and a template of the itinerary.

c) Creating new route



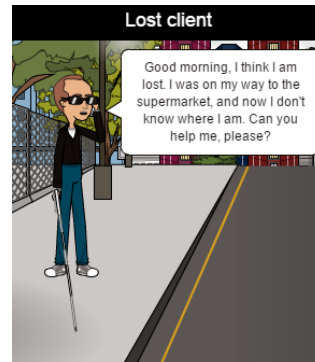
Naviterier Routeplanner will generate a template of the itinerary. The operator only has to check if it is correct and enrich it with some details.

d) Generating template of itinerary



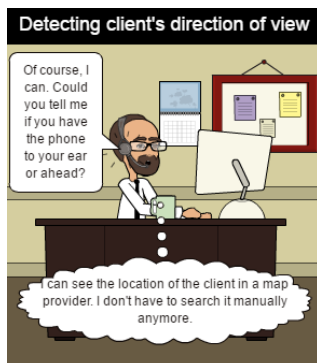
The operator can save his not completed work for later and do something with higher priority, e. g. Handle a call.

e) Saving not completed work



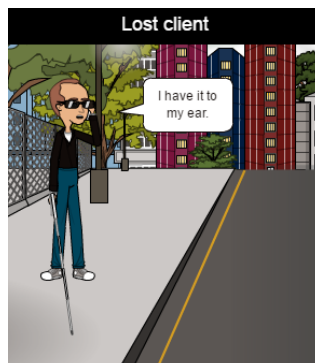
Client is lost and is calling the operator via Remote Assistant application.

f) Lost client is calling via Remote Assistant



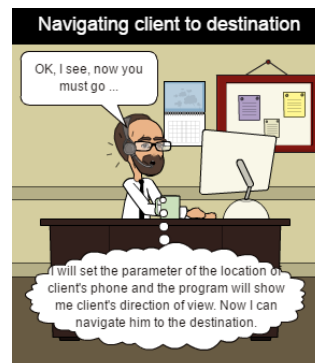
To detect client's direction of view, the operator must ask him about the position of his phone.

g) Detecting client's direction of view



The client has his phone to his ear.

h) Lost client answers the question



When the phone is to client's ear, the interface must rotate his direction of view of 90 degrees.

i) Navigating client to destination

Figure 3 Storyboard showing an example of using several use cases.

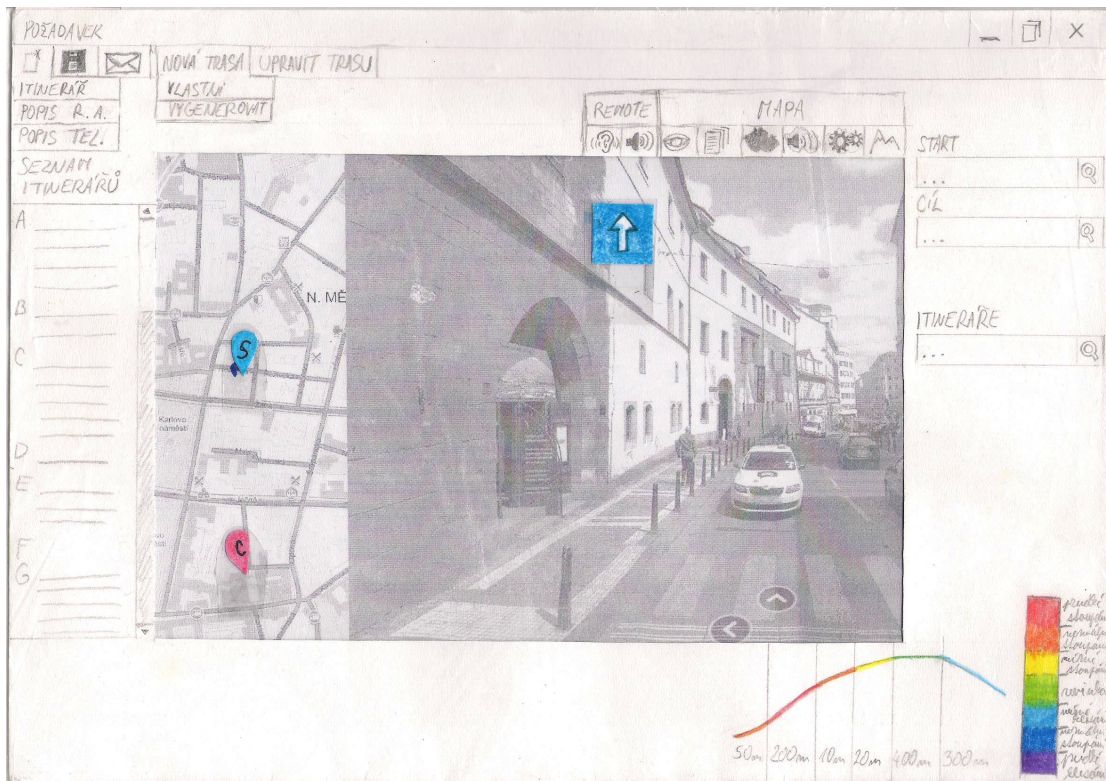


Figure 4 Mockup with opened street view

- minimize
  - maximize
  - close
2. Bar with buttons for:
- creating new:
    - itinerary
    - description via Remote Assistant[3] application
    - description via telephone
  - request saving
  - request sending
  - new route creation:
    - own (drawing)
    - generated via Naviterier[8]
  - route modification
3. Remote Assistant[3] buttons for setting the user's phone location:
- ear icon for client who has phone to his ear
  - loud-speaker icon for client who has phone ahead on loud eavesdropping
4. Map tools:
- street view (see on Fig. 4)

- existing itineraries layer (more in appendix A.3)
- map without layers
- noise layer (more in appendix A.1)
- technical layer
- slope layer (more in appendix A.2)

5. Main parts of platform:

- list of existing itineraries
- map
- searching:
  - start address
  - destination address
  - existing itineraries

6. Graph of slope of individual segments of route with its scale.

#### 3.4.2 Design probe

We consulted the created mockup with the operator of Navigational Centre[1] on 21st November 2016 using the design probe method.

##### Design probe definition

Design Probe is an interactive qualitative method of collecting data. It is aimed to capture the self reported user experience and it is mainly used in the field of applied arts and more precisely in industrial design.[10]

##### Evaluation

The operator liked the interface, and basically, he had no negative comment.

He appreciated the itinerary link along with the map. Generation would certainly help, but he could not imagine how the text itineraries would appear on the map. However, he liked this idea because the written segments would gradually fill up the map and he would not have to write much.

He would like to switch sources (Seznam/Google) of street view, even though he has been working mainly with Seznam maps lately.

The layer and other tools for viewing slope of pavement he said to be useful, but not so necessary for him, because it will be mentioned in the itineraries so that these tools will lose their meaning once.

About Remote Assistant[3], he most appreciated the location of the client's position on the map, because he would not have to look at the map in the app, and because he sometimes finds it difficult to find the client's location on the map. The direction of view of the client seemed a bit unnecessary to him because in most cases, he could orientate in the street view thanks to client's video instantly.

### 3.5 Low-fidelity prototypes

According to the results of design probe of the paper mockup, we created interactive low-fidelity prototypes presented in this section.

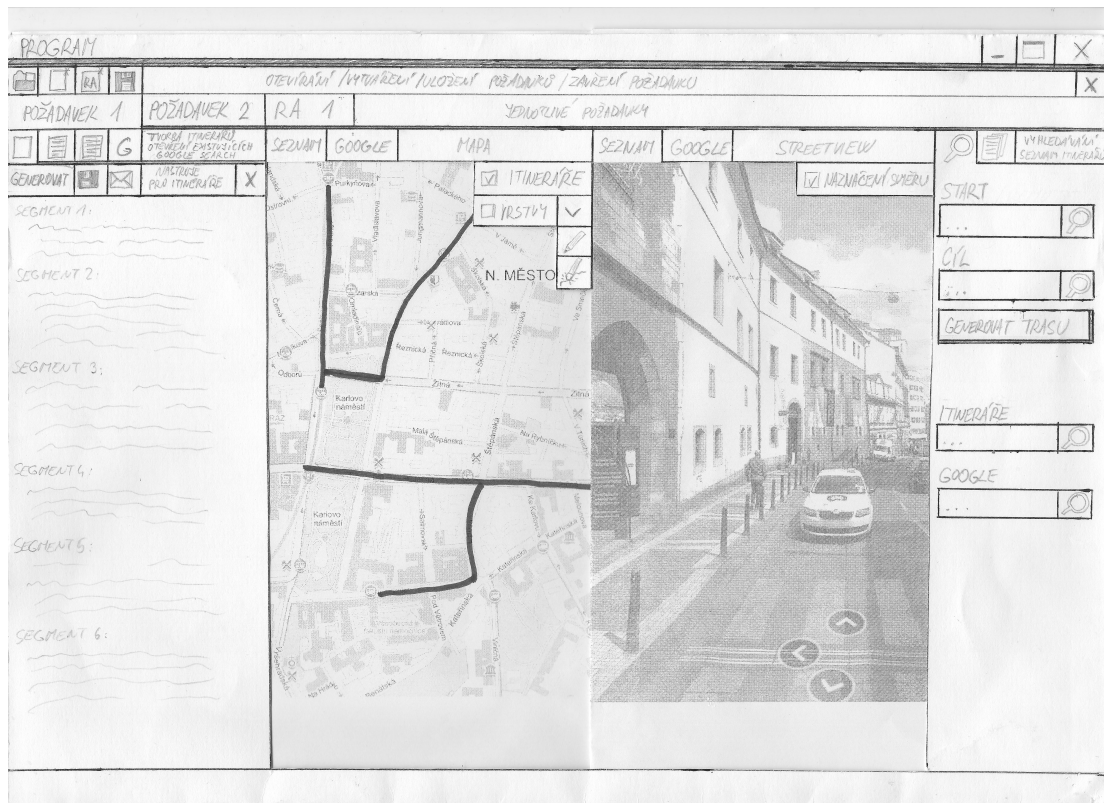


Figure 5 Mockup, which was used as a pattern for a paper prototype

### 3.5.1 Paper prototype

The first low-fidelity prototype was created in paper form, which was subsequently tested with three participants.

#### Prototype description

The prototype interface was drawn on a sturdy paper of A3 size and expanded with other parts drawn on pieces of paper depending on the interaction. More information about all parts of the prototype can be seen in the appendix B.

The Fig. 5 shows a mockup that served as a pattern of a paper prototype, introducing changes and new design elements of the platform and now we will use it for the description of the interface.

#### Interface description

The paper prototype does not differ much from a paper mockup 3.4. A button bar is located at the top of the program window to open, close, save, and create a new request or special request for Remote Assistant[3] (in the interface marked as RA). Then there are tabs for individual already opened requests. The opened tab consists of 4 parts (left to right):

1. The itinerary and other text sources section:

At the top is a bar with all open sources of information. A new itinerary, an open itinerary (twice), and a Google search result. Below is a toolbar for text with: Gen-

### 3 Design

erate, save, send, and close the itinerary buttons. Under the tabs is a generated itinerary divided into individual segments.

#### 2. Map section:

Tabs for map source selection (in paper prototype this is solved using radio buttons in map tools). Then there is a map with map tools. Those include checkboxes to enable individual layers (existing itineraries, noise, technical, slope, and a graph of slope) and buttons to draw and edit a route on the map.

#### 3. Street view section:

As with the map, there are also tabs for street view source selection (in paper prototype this is solved using radio buttons in street view tools). Street view with its tools follows. Street view tools include a checkbox to turn on route direction suggestion.

#### 4. Search and itinerary list section:

There are tabs for search and a list of itineraries. Search is currently enabled. The search contains a field for the start and destination of a route and a button for its generation. There are also fields for search in the list of itineraries and on the Google. The list of itineraries consists of a field of their search and a list where the items are sorted in alphabetical order.

## Evaluation

Usability testing of the paper prototype took place in two days with three men aged between 30 and 40 years. They were university staff with essential experience working with the blind and Navigational Center[1] operator. Everyone was able to complete the assigned tasks and described the prototype as synoptic. Most of them appreciated its sophistication, availability of the most required information and the use of fundamental components (map, street view, layers, etc.). Several suggestions for change have emerged from testing. It's about improving existing features, design changes, and a whole new set of features. Please refer to chapter 5.1 for details of the experiment, including the list of findings.

### 3.5.2 Electronic prototype

Results of testing the paper prototype came with other platform improvements that we incorporated into the low-fidelity prototype in electronic form (Fig. 6) and tested with three participants.

#### Description

In the program Balsamiq Mockups[11], we created the prototype and then exported it to the pdf format that we used for testing.

The interface looks and is exactly the same as the paper prototype (3.5.1) and differs only in the location of some functions. These differences will be presented in the following subchapter.

#### Balsamiq Mockups

Balsamiq Mockups[11] is a tool for creation of wireframes to better understand all the features of the project and its complex architecture. The software includes a variety of

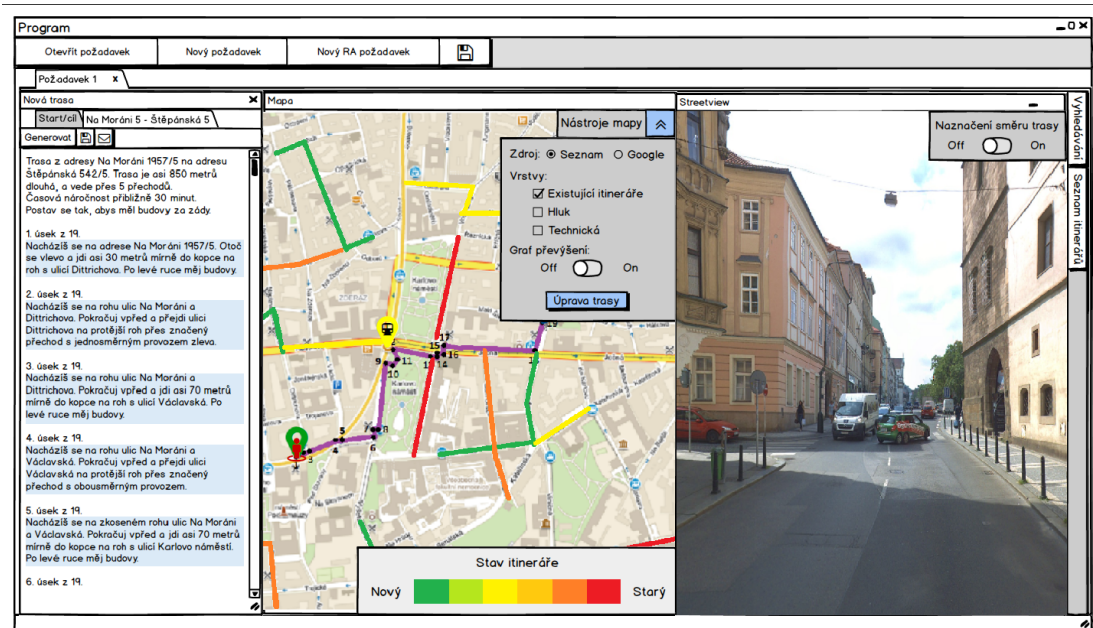


Figure 6 Electronic low-fidelity prototype

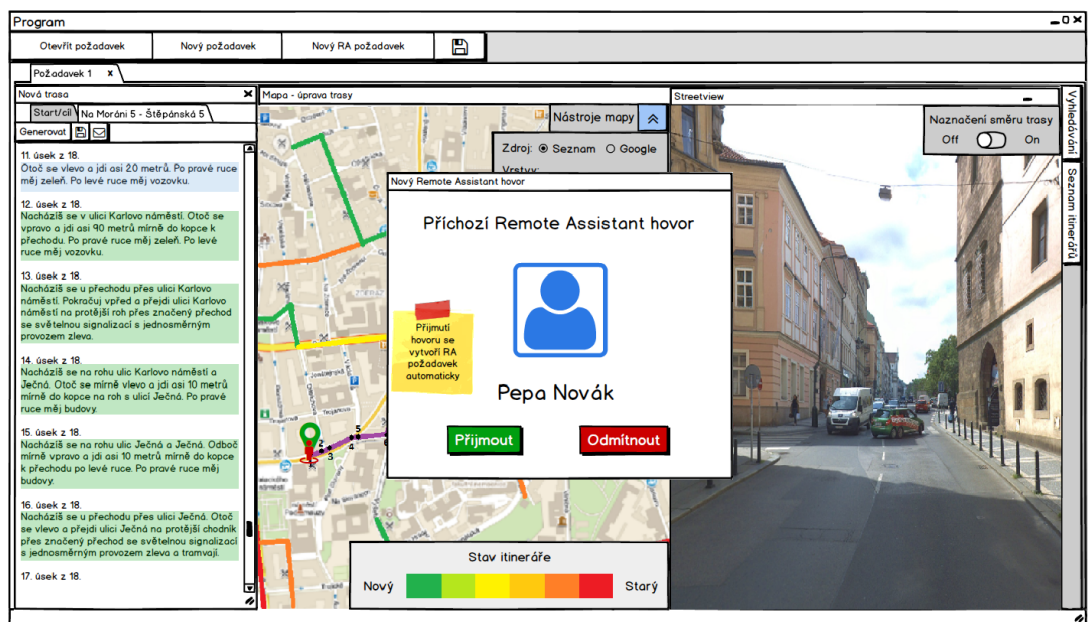


Figure 7 Electronic low-fidelity prototype: Incoming call

pre-arranged layouts, graphic elements, buttons, symbols, control bars, pictures, and icons.

### Changes in the interface

These suggestions that emerged from the paper prototype testing were granted:

- The request closing button has been moved to its tab's label because many of the participants have overlooked its original location or proposed to move it.

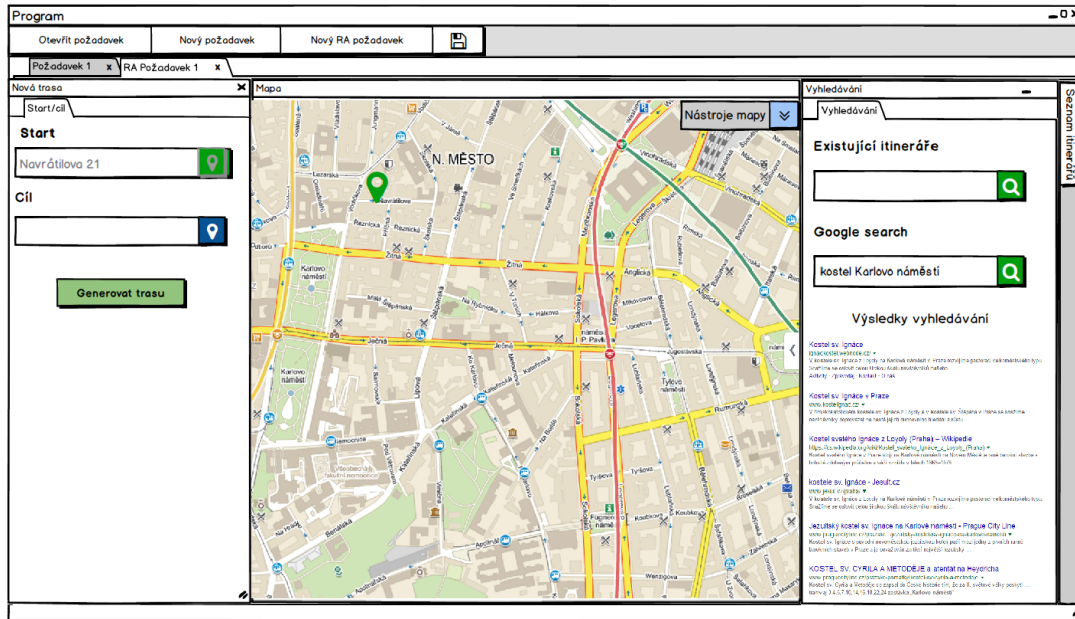


Figure 8 Electronic low-fidelity prototype: Searching result

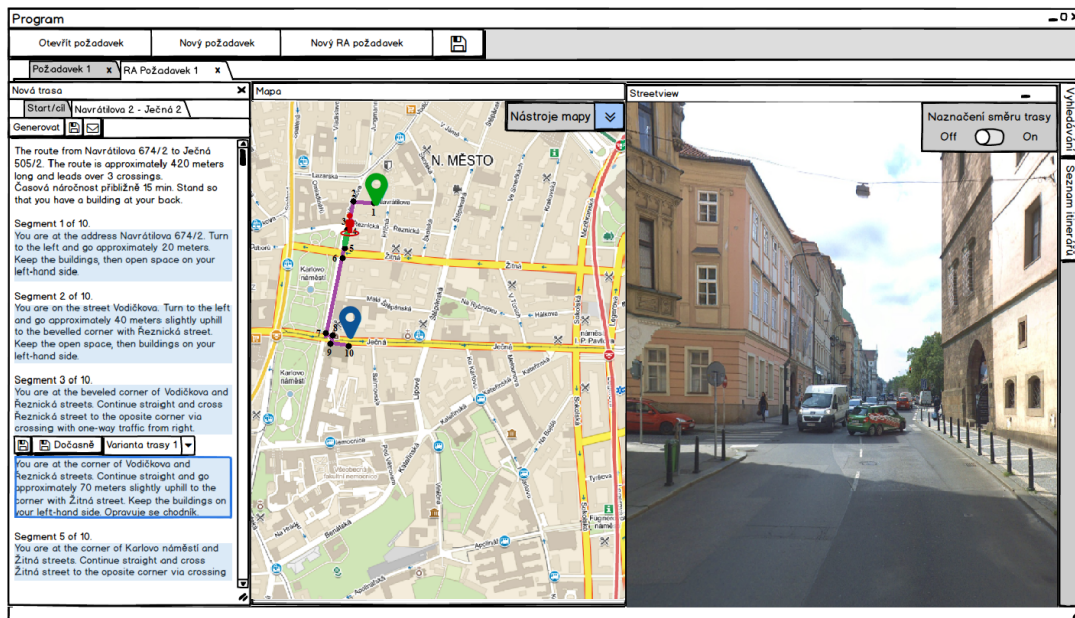


Figure 9 Electronic low-fidelity prototype: Segment buttons



- The start and destination text fields were moved to the itinerary panel, and the search icon on the button near their fields was replaced with the marker because it seemed confusing to some of the participants (Fig. 8).
- Search results have been moved to the search engine (Fig. 8).
- Map tools can be minimized because they take up plenty of space in their open state.
- Each segment now has its textarea, and after a click, it is highlighted on the map. Also, the generated and handwritten segments are color differentiated. Textarea will help with editing, because only a particular part will be edited, not the entire itinerary, to prevent unwanted editing or to delete the whole route. Modified section can then be saved to the database. When editing, buttons for storing it permanently or temporarily (for example, if there is a temporary obstacle that disappears after some time) and a combobox with the choice of the route variant will show (Fig. 9).
- When you enable route modification in map tools, the same button confirms the change. Once the change is confirmed, an information dialog box appears with the information that the itinerary does not match the route and if we want to re-generate it.
- The itinerary was enriched by a rough estimate of the time spent on the route so that the client could at least know if he will catch the bus, how much in advance to go to a meeting.
- The search and list of itineraries remained in place but were hidden under the buttons of the same name because their use is not required, but they can be opened if needed.
- In the layer of existing itineraries, individual routes were colored by a color according to their topicality.
- A notification that someone is calling with Remote Assistant[3] will appear and accepting a call opens a new RA request automatically (Fig. 7).
- The request can be saved as fulfilled/unfulfilled.

#### **Evaluation**

Usability testing of electronic low-fidelity prototype took place on two days with three men aged between 30 and 45 years. They were college staff with essential and long-term experience mainly in the navigation of the blind. They all managed to accomplish their assigned tasks and independently agreed on some prototype insufficiencies. The biggest problem for them was to find a button to edit the route. Otherwise, they liked the prototype and labeled it as sophisticated and easy to use, and gave us further suggestions for improving it. Please refer to chapter 5.2 for details of the experiment, including the list of findings.

### **3.6 High-fidelity prototype**

After testing the electronic low-fidelity prototype, we created a high-fidelity prototype in the form of a client web application to demonstrate how the platform could look and work in the future. For details of its implementation, see 4. Five participants then tested the prototype.

### 3.6.1 Changes in the interface

According to the test results of the electronic prototype, the following changes were incorporated into the high-fidelity prototype:

- You do not have to click on any confirmation button when entering a start/destination address. The marker is set immediately after selecting an item from whisperer.
- Existing requirements, like the itineraries, have been placed in a searchable list, so the user no longer has to worry about where they are stored. At the same time, the request will automatically set the name that is composed of the request start and destination addresses.
- To know that the segments can be edited, no attributes have been deleted from their text areas plus each of them has a button to save or to record the problem.
- The search and its results were placed outside of the request card.
- The route editing button has been moved to the bar above the map.
- The button for saving the temporary segment change has been renamed to "Record a problem."
- The request storing button has been removed. Instead, when the user tries to close it, he will see a modal window asking him if he wants to save it.

### 3.6.2 Functional requirements

This section lists the functional requirements that the resultant platform should fulfill.

1. A user interface containing all the necessary information sources  
The interface should include a map and street view, or possibly a search on the internet (e.g., Google search).
2. Route creation using parameters  
The user enters an initial and destination address, or other parameters and the system should be able to generate the route according to them.
3. Route view  
The system should be able to display the generated route on the map.
4. Route modification  
The user should be able to edit the generated route on the map by moving, adding and deleting its points.
5. Generating itineraries  
The system should be able to generate an itinerary for the created route using the Navitierier[8].
6. Itinerary view  
The system should be able to display the generated itinerary in the interface.

## 7. Itinerary modification

The user should be able to modify the generated itinerary.

## 8. Saving itineraries

The user should be able to deposit the itinerary.

## 9. Access to existing itineraries

The user should be able to view a list of existing itineraries and search for it.

## 10. Transferring client's location from RA

The system should be able to transfer the client's position from the Remote Assistant[3] application to the map.

## 11. Saving of not completed work

The user should be able to save his not completed work for later.

## 12. Recording obstacles on the route

The user should be able to record permanent/transient obstacles along the route.

### 3.6.3 Interaction design

In this section, we will introduce the prototype interaction design using the five dimensions, except for the time dimension, because in our prototype is completely missing.

The five dimensions, created by Professor and Head of Department of Computer-related Design Gillian Crampton Smith and senior interaction designer Kevin Silver, represent the ways in which a person can interact with a product or service.[12]

#### Words

We tried to make the interface contained as little text as possible, so all the labels and button names consist of a maximum of two words that summarize their purpose (Fig. 10).

The exception is the button to create a new RA request because it is necessary to indicate that it is a special request and buttons and labels of existing itineraries/requests where their name suggests the start and destination of the route.

#### Visual representations

In the prototype, we decided not to use well-known button icons but text, to help users to identify what the buttons mean. Some of the buttons are only highlighted in green or red, so the user does not overlook them. We use it for generating a route, search, and modal window buttons.

The use of visibility to interact is mainly seen on a generated route on a map that has a distinctly visible light blue color and on a layer of existing itineraries where the itineraries are marked with color according to their age. Green means that the route is new and red indicates a problem (in our case an ancient route).

Another visual element, which also applies to the route, appears when the modification is turned on (Fig. 11). Specifically, you'll see individual route points that suggest they are here to be edited by a user. The point can then be marked (changing its color) and moved elsewhere.

### 3 Design

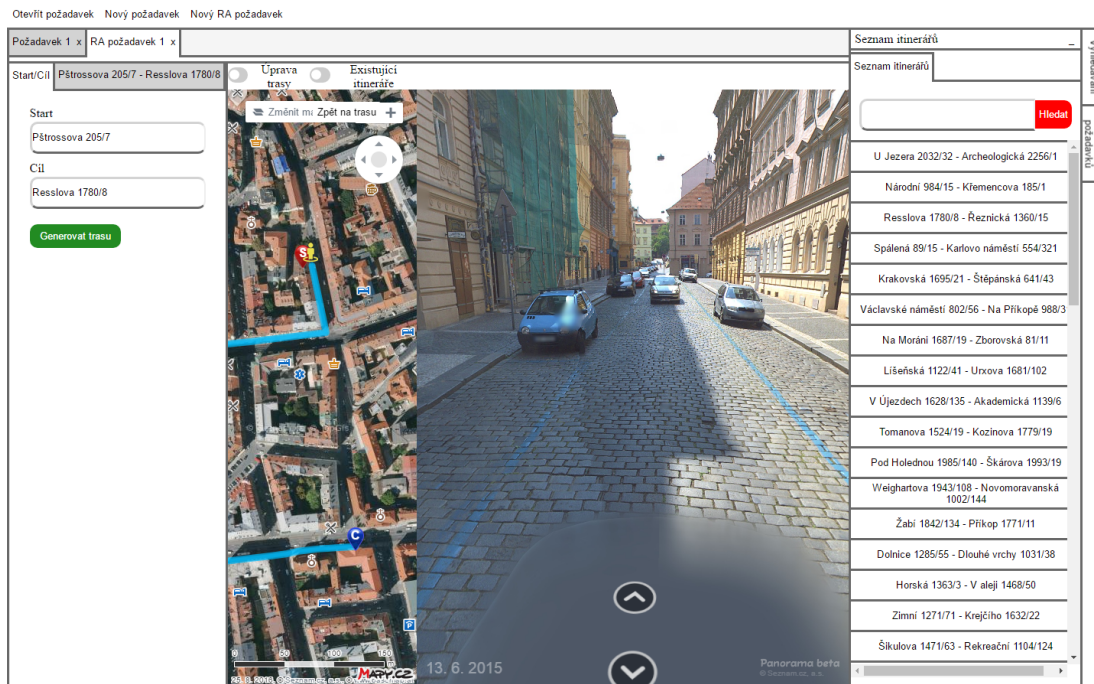


Figure 10 High-fidelity prototype

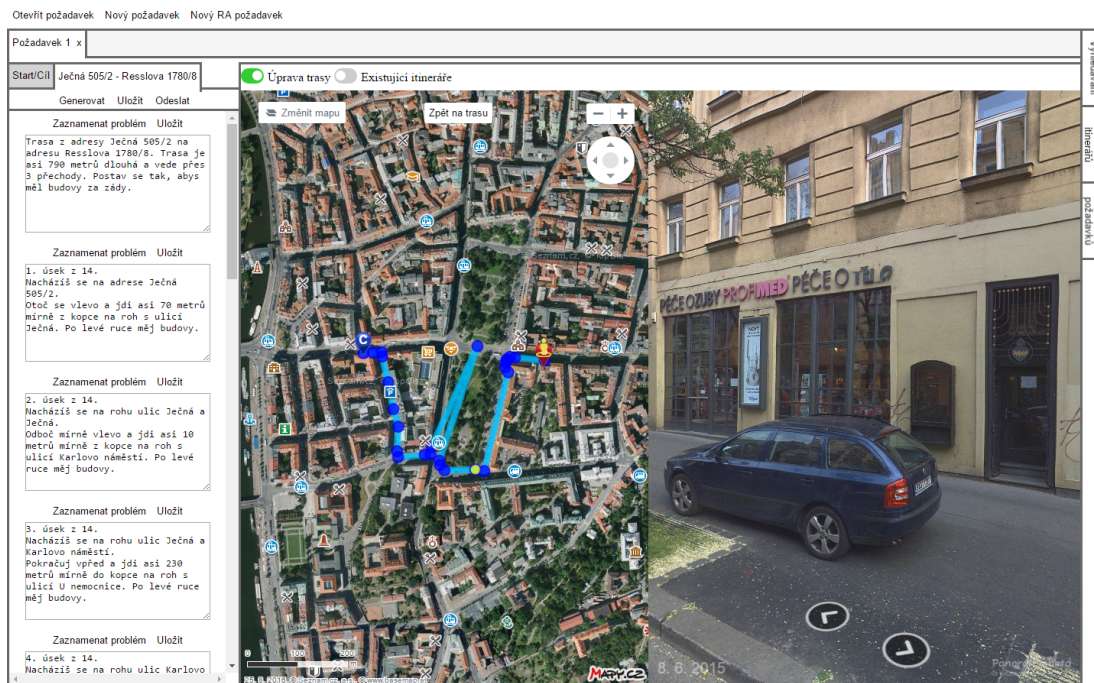


Figure 11 Visual representation of route modification

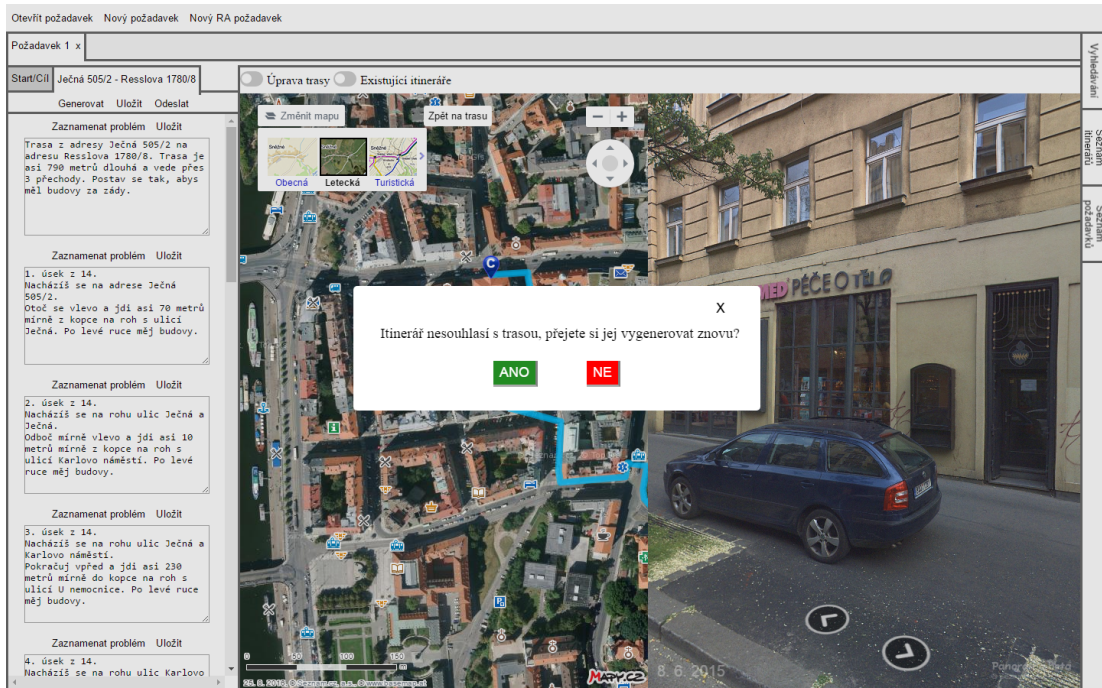


Figure 12 An example of a modal window in high-fidelity prototype

### Physical objects/space

Operators of the tele-assistance centers for navigation of visually impaired people for which we design this platform have a computer with a large monitor (or two), so the interface is adapted especially for displays with a large diagonal and for mouse control. But it is also executable on mobile devices and can be controlled by touch, but we do not recommend this because on small displays the interface for meaningful work is almost unusable.

### Behavior

The user controls the interface using the buttons provided, or fills and modifies the text fields or works with a map or street view. The system responds to his interaction by changing the look of a particular element or the entire application, or by displaying a modal window (Fig. 12) in which the user can affect the behavior of the application (whether to save the request, re-generate the route, etc.).

#### 3.6.4 Visual design

Since it is only a prototype, we did not try to create any sophisticated elements of the visual design and focused on simplicity.

Lines separate individual parts of the prototype. The top buttons to open and create a new request are separated from the workspace itself. It consists of a part for opened requests and to the right is the toolbar with buttons for lists and search, which are minimized so that they do not waste too much space when they are not used.

The request section consists of tabs of individual requests, where the inactive requirements are gray and separated by a line from the current requirement workplace. Active request tab is white, and its bottom edge is linked to its workplace.

There is a section on the workplace to enter itinerary parameters and to edit it, which is separated from the section for map and street view because they take up the most space and the user will work with them the most.

Background colors are white and shades of gray only, also for buttons, except for search buttons, route generation, and modal windows buttons which are red or green. The white buttons after mouse hovering over it will change the color to gray.

The color of the route on the map is light blue to make it easy to see. For layers of existing itineraries, colors ranging from green, yellow to red were selected to indicate the age of the route. Green = new, red = very old.

Start and destination markers are color-coded and labeled S = start, C = destination ("Cíl" in Czech).

#### 3.6.5 User experience design

The prototype has been designed to be as simple as possible to make the user learn how to work with it very quickly and liked to come back to it.

It incorporates essentially all of the tools that an operator, who is creating new routes for the visually impaired, needs, to be able to do just with this interface and with nothing more.

The map and street view were created using the Mapy.cz API[13] from Seznam that the operator knows, so he is not forced to learn to work with it again. And overall, the design and location of the individual buttons and other content we have been tried to make them not too different from what the operator has ever been forced to work with until now.

#### 3.6.6 Evaluation

Usability testing of high-fidelity prototype took place on five days with five participants aged between 20 and 35 years.

There were people with many years of experience but also people with almost no experience with work with visually impaired people. Everyone was able to complete the tasks either individually or with a little help, and they all agreed on some prototype deficiencies. Everyone would like to drag and drop to edit the route and enter the name of the point of interest as the start and destination. Otherwise, the interface, despite all the bugs, liked and came to them as a powerful tool in experienced hands.

Please refer to chapter 5.2 for details of the experiment, including the list of findings.

## 4 Implementation

The prototype was created as a throw-away prototype. That means that only a small part of the system was created, which was then tested with users to see if their requirements were properly defined and understood or to detect some discrepancies and get feedback, which can be incorporated into the main system. The development of the prototype is then not continued (therefore throw-away).

The prototype was developed in the form of a client web application using HTML5, JavaScript, and CSS.

You can view the resulted application and try it on [leyfi.felk.cvut.cz/naviteriersons](http://leyfi.felk.cvut.cz/naviteriersons). The application should work in Google Chrome web browser.

### 4.1 HTML5

All HTML is located in the index.html file, except for the parts generated by JS (e.g., segment's text areas).

To save space on the screen, we decided not to display the header and footer of the page. This makes the interface more similar to a desktop application. It consists of a large number of sections, navs, mains, divs, and other parts that are not even worth describing.

Svg and canvas, none are defined here, but they are generated by some of the APIs we use.

### 4.2 JavaScript

#### 4.2.1 Classes

The application is partially object-programmed. There are two classes specifically for the request and an itinerary for storing their data.

##### **Request class**

The class is used to store and retrieve data for individual requests and you can find it in the Request.js file.

It consists of a constructor specifying the id, starts position and type of request, and getters and setters for:

- name
- start location
- destination location
- last street view position
- itinerary

## 4 Implementation

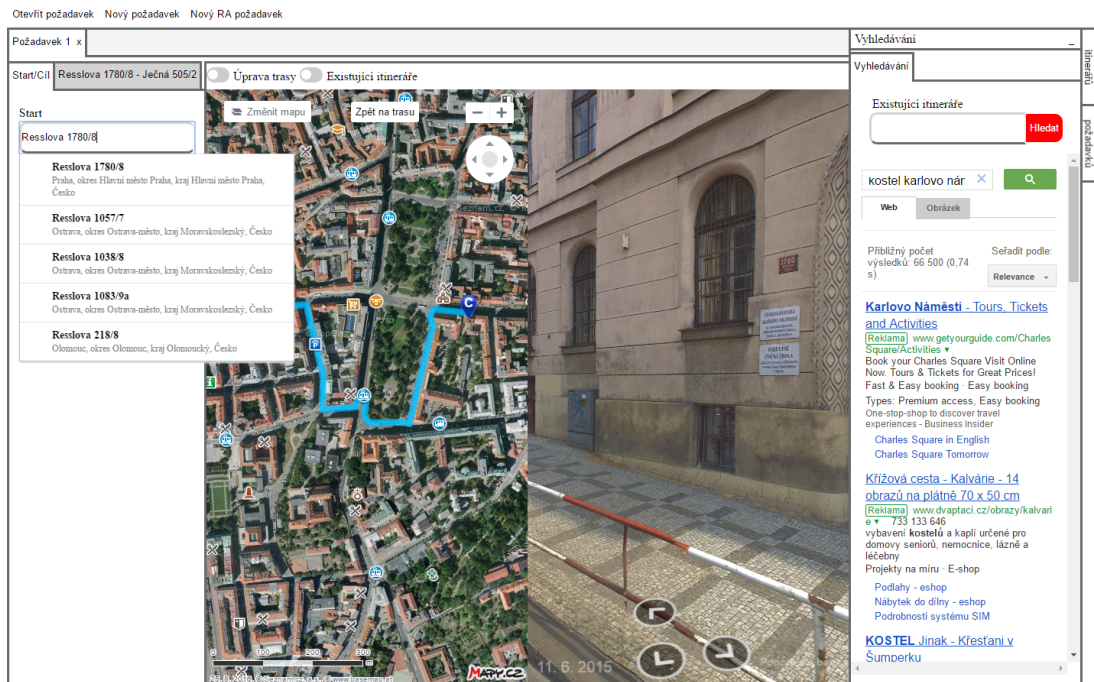


Figure 13 APIs in high-fidelity prototype

### Itinerary class

The class is used to store and retrieve data for individual itineraries generated by Naviterier[8] and you can find it in the Itinerary.js file.

It consists of a constructor specifying its description, itinerary segments and coordinates and getters and setters of these parameters.

### Stage class

The class is used to store individual segments of itineraries. You can find it in the Itinerary.js file.

It consists of a constructor specifying its text and type (if the segment is written or generated) and getters and setters of these parameters.

### 4.2.2 Program logic

All logic of the program is located in the utils.js file. It is a large implementation of features for generating content, working with maps and with street view and controlling the entire interface, including sending requests to the Naviterier's[8] server, which sends back the response with the generated itinerary in JSON format.

### 4.2.3 Used APIs

#### Mapy.cz API

Mapy.cz API[13] is an API for viewing and working with a map and street view from the Czech company Seznam.cz. It offers a wide range of features and is completely free also for commercial purposes.

In the interface, we used it for whisperer, map, and street view. Further to view the route and map layers, work with markers and bi-directional geocoding (Fig. 13).



### **Google Custom Search API**

The Google Custom Search API[14] as the name itself says is an API to create a custom search engine whose script is then inserted into the parent element. We used it in the search section to have all the sources together (Fig. 13).

### **Other APIs**

Additional APIs have been used to create, modify, or delete interface content and work with listeners. E.g., Java API for KML[15] (JAK), Document Object Model API[16] (DOM API) and jQuery API[17].

## **4.3 CSS**

The styling of the entire interface is located in the style.css file. For most elements, the Flexible Boxes[18] layout mode was used.

# 5 Evaluation

## 5.1 Evaluation of paper low-fidelity prototype

A lot of valuable suggestions and comments were received during testing of the paper prototype, which concerned, in particular, the modifications of the existing functionalities, suggestions for new features and a few changes in design. Most of them were then incorporated into the next, electronic low-fidelity prototype.

At Fig. 14 you can see what the prototype looked like during the test. Individual parts of the prototype can be found in Appendix B.

### 5.1.1 Experiment description

The test took place with three participants to meet the conditions and get all the necessary information. For each participant, an hour was reserved which seemed too long in the final because it took about half an hour for the preparation and testing itself, but the exception was the 3rd participant who had quite a talk, and the testing with him took almost an hour.

Besides to the paper prototype, writing accessories and a laptop with a prepared post-test questionnaire were used for testing. The whole test was recorded on the GoPro Hero3 camera, so we were able to process it later.

The pre- and post-test forms were prepared for the participants. In the pre-test questionnaire, we asked them about their experience of working with the visually impaired people and the software, and also their age. The output survey concerned their feelings about testing and the prototype itself.

Below you can see the questions of individual questionnaires with the answers of each participant.

#### Pre-test questionnaire

The questions asked in the pre-test questionnaire are presented in Tab. 1.

#### Post-test questionnaire

The questions asked in the post-test questionnaire are presented in Tab. 2.

### 5.1.2 Experiment procedure

Testing took place with 3 participants in two days in hourly blocks, but most participants managed to complete all the tasks much earlier.

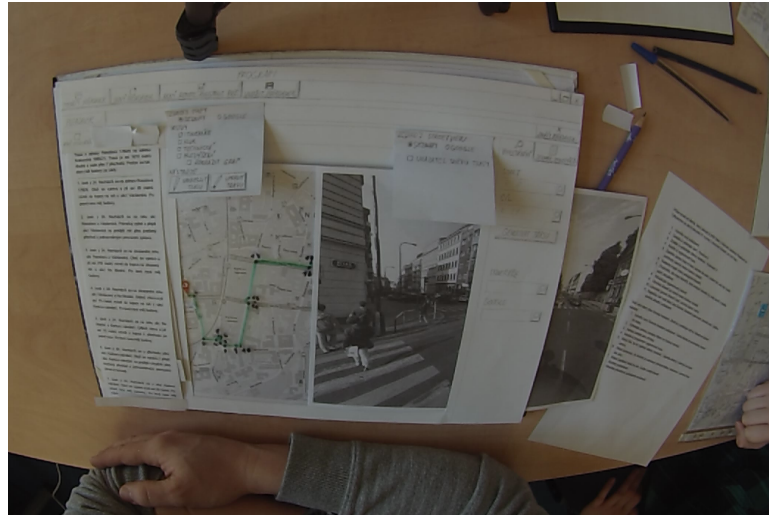
At the beginning of each session, we introduced the subject of the problem and explained what the usability testing is if he didn't already know it. Then the interface was presented to them using a paper mockup (Fig. 5). After this brief introduction, a test scenario (Appendix C.1) was passed to the participant with individual tasks and testing started. The participants progressively performed the tasks and the moderator, according to their interaction, changed the prototype under their hands.

	<b>Participant 1</b>	<b>Participant 2</b>	<b>Participant 3</b>
<b>Age:</b>	30 - 40	30 - 40	30 - 40
<b>Your experience with software: (For example, whether you are working with it on a user level or you are developing it, experience with testing, etc.)</b>	User only.	With some SW I work only at the user level (e.g., school information systems). Otherwise, I have fairly good experience with .NET development. Including the development of a custom client-server solution that integrates various target UIs (PC, Smartphone, TV ...). Including automatic generation of context-sensitive UIs. I have fairly good experience with qualitative and quantitative UI testing. I have experience with SW testing in general (integration tests, unit tests, stress tests).	With the software, I have experience with all phases of the process, i.e., design, implementation, and testing. Currently, I design the most and implement and test a bit.
<b>Your experience with working with the visually impaired people:</b>	As a Navigational Center for the Blind operator, I have been working for five years.	My experience with development for the blind is based on the i2home project (intelligent household), where we primarily dealt with a group of seniors. I also contributed to the development of some solutions for the blind, mostly as part of student project management. Developing for blind users, I now deal mainly with the InHospitalNavigation project - navigation in the interior of large buildings.	Experience is essential, I have been the moderator of several tests with the blind. In the last ten years I have been part of the research of navigation for the blind in the form of preparation of experiments and analysis of data, and about 15 years ago I was doing about a year a conductor of blind skier.

**Table 1** Paper prototype pre-test questionnaire

	<b>Participant 1</b>	<b>Participant 2</b>	<b>Participant 3</b>
<b>How did you like testing?</b>	I liked it.	The tasks seemed to me quite detailed, making it easier for me to work with the interface (maybe too much).	I liked it. But I had several suggestions for change.
<b>How would you evaluate the tested prototype?</b>	Great. Organized - intuitive.	Very complex on the low-fidelity prototype. Overall quite organized. Allowed me to perform assigned tasks.	Good, I have a few suggestions for changes.
<b>What did you like on the prototype?</b>	Working with itineraries. Possibility to use previous routes.	Sophistication, availability of most of the necessary information.	Good integration and use of standard components as a map, street view, layers.
<b>What did you not like on the prototype?</b>	-	Work with segments of itineraries.	Grouping information, such as search boxes and search results. I was confused about which itineraries are generated and which are created by hand.
<b>Is there some functionality missing in the prototype? If so, what?</b>	- The possibility of interconnection with timetables and public transport. If part of an itinerary is available via public transport - show the departure times in the results. - A rough estimate of the time spent on the route.	Identifying the up-to-date of an itinerary segment in the map. Identification that the segment is generated automatically. Notification of RA requirement.	Better connection of itineraries.

**Table 2** Paper prototype post-test questionnaire



**Figure 14** Picture taken during a test of paper prototype.

Below you will find detailed inventories of testing with individual participants.

After the participant had completed the test, he received a post-test questionnaire to complete. After filling the survey he received a small sweet reward as a thank, and he could go home.

### **Participant 1**

The first participant, as operator of the navigational center, never tested, and therefore probably the paper prototype seemed too abstract to him so did not know what to say.

He was able to complete all the tasks until the last one because he did not find a button to close the request. He only saved the request and opened a new one. But what we would highlight was his independence when he could find out that some of the prototype's part was missing because the moderator had forgotten to place it on the prototype.

As far as the search for a supermarket address is concerned, he would find it in practice somewhere else because of the address and put into the interface.

Otherwise, he liked the prototype, he found no negatives and devised only a few new features that could be embedded in the system. He liked the most the work with itineraries and use of existing routes.

### **Participant 2**

Participant number two was an experienced tester and developer. Tasks seemed too suggestive to him, and he said that the prototype looks too sophisticated for low-fidelity.

When working with the itinerary, he imagined that when he clicks the itinerary it will display on the map and he would edit the route by dragging the points with a mouse. After a few minutes, he found the button to turn on the editing. Consequently, he would be happy if the itinerary will update immediately after the route was changed.

With a layer of existing itineraries, he figured out how to find out the itinerary's age after some time, so he would color the route straight by its age not to have to click on every route.

Halfway through testing, he was somewhat confused by the nomenclature, because he had found that everything was an itinerary. If he found a non-existent section, he

would like to mark it and provide its information somehow, but he is not sure what is generated by the Naviterier and what is written manually.

When there is incoming RA call, something would happen with the interface.

Otherwise, the prototype seemed very complex and synoptic. He only reproached the work with itinerary segments and did not like the location of some buttons.

### Participant 3

The third participant also had experience with designing and testing the software, during the test he had a long talk and a lot of comments.

First, he looked at what was on the interface. He did not like the button to close the request because it was far, and the placement of the start and the destination below the search section because they are the items that need to be entered. At the same time, when he filled these parameters he expected the markers to be set straight on the map not to have to click on some button to confirm it.

By generating a route, he would consider his job to be over, but he would intuitively use drag and drop or add a point on the route. On the other hand, the points seem fixed to him, so he was not sure about the result. After editing the route, he would like to confirm the change, but this feature was not there. The update term evokes change from the system, not the map in him. Itinerary regeneration would be solved by either a modal window or it would be set immediately after modification, but he would like to see the information that the itinerary does not match the route.

For one complicated section, he would create more different variants that would then be offered near the affected segment (e.g. combo box).

In the itinerary, for each segment, he would create a text area not to change something he did not want.

Like a second participant, he would expect the incoming call information to be displayed in the interface and, once accepted, would open an RA request.

Search results would be placed just below the search engine.

In dealing with the obstacle on the route (front garden), he did not know how to solve it, but it came to him that some of the barriers could be called temporary.

In spite of all the comments, he liked the interface and highlighted the good integration and use of map, street view and layer components.

### 5.1.3 Experiment findings

In this section, we will list all improvements to the interface which have been mentioned. These include new features, design changes, and changes of existing features.

#### Suggestions for new features

1. A rough estimate of the time spent on the route.
2. The possibility of interconnection with timetables and public transport if the part of the route is realizable via public transport.
3. Different variants of one segment (e.g., a combo box for a segment).
4. Drawing the exits from the metro.
5. Temporary obstacles: Exposure setting for temporary obstacles (front garden only over the summer).

### **Changes in design**

1. The button to close the request move on the request card label.
2. Move search results to the search engine.
3. Move the start and destination field to the itinerary panel
4. To replace the magnifying glass at the start and destination buttons with another mark (e. g. map marker).

### **Suggestions for changes to existing features**

1. To differentiate generated and manually created itineraries (segments).
2. Identify the itinerary age directly on the map (color the highlighted section directly).
3. Opening the RA request immediately after receiving a client call.
4. Saving request as fulfilled/unfulfilled.
5. When you click on an itinerary, the edited section is highlighted on the map.
6. When you click on a particular segment of a map, that section is highlighted in the slope graph.
7. Drag and drop to edit a route.
8. Information that the itinerary does not match the route - "Would you like to regenerate it?"
9. After editing the route, confirm the modification.
10. If there is no itinerary for a segment, you can mark it and write an itinerary directly for it.
11. Minimize map/street view tools.
12. Text area for every itinerary segment.

## **5.2 Evaluation of electronic low-fidelity prototype**

After incorporating the changes that arose from testing the previous prototype, it was time to test a new prototype, this time in electronic form. From this testing, we also took a lot of rare suggestions and comments for an even better interface. Similar to the paper prototype, it was mainly about the modifications of the existing functionalities, suggestions for new features and a few changes in design.

### **5.2.1 Experiment description**

As with a paper prototype the test took place with three participants to meet the conditions and get all the necessary information. For each participant, an hour was reserved which seemed too long in the final because it took about half an hour for the preparation and testing itself.

A laptop with an interface in the form of a pdf file was prepared for testing. The test was recorded using a screen capture tool, so we were able to process it later.

	<b>Participant 1</b>	<b>Participant 2</b>	<b>Participant 3</b>
<b>Age:</b>	40+	30 - 40	30 - 40
<b>Your experience with software: (For example, whether you are working with it on a user level or you are developing it, experience with testing, etc.)</b>	I design, develop, test.	With the software, I have experience with all phases of the process, i.e., design, implementation, and testing. Currently, I design the most and implement and test a bit.	I test, use, and implement the software
<b>Your experience with working with the visually impaired people:</b>	I have been working with the blind for more than 15 years. Especially in the field of navigation and application accessibility.	Experience is essential, I have been the moderator of several tests with the blind. In the last ten years I have been part of the research of navigation for the blind in the form of preparation of experiments and analysis of data, and about 15 years ago I was doing about a year a conductor of blind skier.	Long-time with orientation and navigation, the creation of navigation software.

**Table 3** Electronic prototype pre-test questionnaire

The pre- and post-test forms were prepared for the participants. In the pre-test questionnaire, we asked them about their experience of working with the visually impaired people and the software, and also their age. The output survey concerned their feelings about testing and the prototype itself.

Below you can see the questions of individual questionnaires with the answers of each participant.

### **Pre-test questionnaire**

The questions asked in the pre-test questionnaire are presented in Tab. 3.

### **Post-test questionnaire**

The questions asked in the post-test questionnaire are presented in Tab. 4.



	<b>Participant 1</b>	<b>Participant 2</b>	<b>Participant 3</b>
<b>How did you like testing?</b>	All right.	-	-
<b>How would you evaluate the tested prototype?</b>	The design is successful.	It solves complex work on the creation of itineraries leading to a complex UI. It is necessary to separate and combine functionalities well.	Relatively easy to operate on such a complex system.
<b>What did you like on the prototype?</b>	Searching for existing routes/itineraries. Working with layers (route visualization and editing option).	Use of standard components. Route search, work with layers above the map. Age of itinerary routes.	I like the ability to modify the generated route and street view connection with it.
<b>What did you not like on the prototype?</b>	Editing itinerary when you find a problem on the route. Confusing "Temporary save." Storing itineraries into the directory structure - I would like to be spared from it (using smart route/itinerary search).	Google searching is interfering with the map when I search for places. The button for editing the route is too hidden. Button for storing the request is too far from the window, and it is not clear to which window is linked. Additional features could be added to itinerary entries, at least signaling the possibility of making changes.	See record - Need to click the marker to see the address on the map. Need to enable drag and drop when editing a route. Otherwise, I liked it.

**Table 4** Electronic prototype post-test questionnaire

## 5.2.2 Experiment procedure

Testing took place with 3 participants in two days in hourly blocks. However, all participants were able to complete the tasks within half an hour.

At the beginning of each session, we introduced the subject of the problem and explained what the usability testing is if he didn't already know it. Then the interface was presented to them. After this brief introduction, a test scenario (Appendix C.2) was passed to the participant with individual tasks and testing started.

Below you will find detailed inventories of testing with individual participants.

After the participant had completed the test, he received a post-test questionnaire to complete. After filling the survey he received a small sweet reward as a thank, and he could go home.

### Participant 1

The first participant was very experienced in design and testing, and he described all his activities nicely, that it seemed like he was telling a fairy tale.

Right from the beginning of the search for an existing itinerary, he was confused that there was a preset street with a house number that he would not mention there.

After filling start and destination address, he skipped the marker buttons. He would welcome multiple route parameters. After generating the route, he would see if it meets his requirements, he would zoom somewhere, then look at the itinerary and check. If he encountered a problem on the route, he would have labeled a section that he did not like and edited it immediately. Then he would send or save the itinerary and get his job done.

He liked the layer of existing itineraries very much. He would always turn it on to see what he could use. Finally, he mentioned that he did not understand the 4th task at first, yet he did it "the interface was more intuitive than the description." said.

When editing the route, he overlooked the button to turn it on. He would rather modify the route using the context menu. He would move points or create a point and move it to rebuild the route, etc. The button to confirm the route change has come to be non-intuitive for him, he would save it by saving the itinerary.

The church would either enter the destination or the search. He would like if the click on a link in the search results will show the location of the searched object on the map.

Since the participant was one of the creators of the Naviterier, it came to him that the created itinerary could be sent to the client directly to his phone if he had the Naviterier application installed.

When encountering an obstacle, he would like to mark a problem on the route, and the system would solve it. He meant something like an invalidation of the segment. He does not know what temporary save means. For his surprise, it was what he imagined - setting the duration of the obstacle. With a big obstacle, he would invalidate the segment.

Since the RA request was opened by receiving a call, he would like to close it by putting down. After the program asks for saving the request, he does not like saving files. He would not save the RA request.

He would like to edit the old itineraries directly in the program.

Otherwise, he described the design as successful, and he liked the work with existing routes and itineraries the most.

## Participant 2

Participant 2 tested the prototype already in the previous stage, so he already knew it and completed the test very quickly, but still had a couple of comments.

Again he skipped the buttons to set start and destination markers on the map. He would generate a route immediately, on the other hand, he notes that it might be confirmed.

When the layer for existing itineraries is turned on, he wanted to hide the toolbar for the map, but instead, he clicked the button to edit the route because he considered it to be hiding the menu button.

He did not like the fact that he did not know if there was an existing route under the generated route. Otherwise, he would not change it.

He did not know how to edit the route, so the moderator had to advise him. He thought it was the button to save the layer selection. He said that the edit button is invisible, and he would move it somewhere else, but he didn't know where.

He also lacked information that he could change the segment. Then he does not know what the difference is between save and temporarily save buttons. He would rename them.

He called the interface as comprehensive and highly appreciated route tracking, work with layers, and displaying the age of existing itineraries.

## Participant 3

The third participant was also quite experienced. Just like others, he would not click on buttons when he was filling the start and destination and let the system generate the itinerary immediately

He wanted to modify the route immediately, without any turning on because he also could not find the edit button.

When encountering an obstacle, he would mark that there is no way to go there and allow the system to regenerate the itinerary.

He also did not like the fact that Google is far away from the route and that he needs to copy the address from it.

Otherwise, he fulfilled all the tasks without bigger problems, and he said that the prototype had a relatively easy control. He liked the most editing the generated route and its connection with street view.

### 5.2.3 Experiment findings and discussion

In this section, we will list all improvements of the interface which have been mentioned. These include new features, design changes, and changes of existing features.

#### Suggestions for new features

1. More parameters for route generation (e.g., noise).

One of the participants would like to be able to set more route parameters to generate it according to the client's special needs.

2. "Mode" to edit old itineraries and create new ones.

For example, when the operator has a free time between request, he could turn on the mode to modify existing itineraries or create new ones and prepare it for future requests.

3. Incorporate call termination. It would close the RA request.

If the interface is already enabled to accept the call, there should also be a termination button that would close the RA request.

4. Disable segment (major obstacle, permanent obstacle).

The interface should be enabled to disable the segment. E.g. There was a new house that blocked the pavement. The system would then resolve this problem by re-generating the route.

5. Send an itinerary to the client to the Naviterier application if he has it (The idea of one of the creators of the Naviterier[8]).

6. Link the Google results with the map.

Some participants did not like to copy the address from the results of the Google to the field at the other end of the interface. They would, therefore, welcome the linking of the results with the map.

### Changes in design

1. Save request button move to the request.

The participants didn't like that a button to save the request was too far from the request, and they did not know to which one it belongs to.

2. Rename the buttons for the segment change. E.g., "Save new option," "Save existing" or something like that.

Some of the participants were unclear what the buttons meant, or what was the difference between them.

3. Move the route modification button somewhere where it will be visible.

All participants overlooked the edit button. Some of them misplaced it with a different button. So it would be appropriate to move it somewhere to prevent this situation.

4. Indicate that segments can be edited (icon, comment, etc.).

Text areas' borders were removed, so the participants did not know they could change the segment. This mistake would be fixed for the next time.

### Suggestions for changes to existing features

1. Set the markers immediately after filling the start and destination fields.

All participants skipped the buttons to set the marker for the start and destination, so these buttons are redundant and can be removed.

2. Opening the existing requests like the list of itineraries - list and search (not files).

One of the participants did not like the requests stored in the files. He would like to be spared of it. Therefore, the requirements can be available in a list as well as the existing itineraries.

3. Route editing via adding and deleting points (context menu), drag and drop.

All participants said they would move the points using drag and drop or that they would take or remove a point, for example, using the context menu.

4. Edit the route immediately without button clicked.

All participants overlooked the edit button. Some of them misplaced it with a different button. So it would be appropriate to move it somewhere to prevent this situation or remove it.

## 5.3 Evaluation of high-fidelity prototype

At the final stage of the project, we created a high-fidelity prototype web application that we tested with 5 participants. Testing was successful, and there were only a few suggestions for design changes and changes to existing features.

### 5.3.1 Experiment description

The test took place with three participants to meet the conditions and get all the necessary information. For each participant, an hour was reserved which seemed too long in the final because it took about half an hour for the preparation and testing itself, but the exception was the 3rd participant who had quite a talk, and the testing with him took almost an hour.

Besides to the paper prototype, writing accessories and a laptop with a prepared post-test questionnaire were used for testing. The whole test was recorded on the GoPro Hero3 camera, so we were able to process it later.

The pre- and post-test forms were prepared for the participants. In the pre-test questionnaire, we asked them about their experience of working with the visually impaired people and the software, and also their age. The output survey concerned their feelings about testing and the prototype itself.

Below you can see the questions of individual questionnaires with the answers of each participant.

#### Pre-test questionnaire

The questions asked in the pre-test questionnaire are presented in Tab. 5.

#### Post-test questionnaire

The questions asked in the post-test questionnaire are presented in Tab. 6.

### 5.3.2 Experiment procedure

Testing took place with 5 participants in 5 days in half an hour blocks.

At the beginning of each session, we introduced the subject of the problem and explained what the usability testing is if the participant didn't already know it. Then the interface was presented to them. After this brief introduction, a test scenario (Appendix C.3) was passed to the participant with individual tasks and testing started.

Below you will find detailed inventories of testing with individual participants.

After the participant had completed the test, he received a post-test questionnaire to complete. After filling the survey he received a small sweet reward as a thank, and he could go home.

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
<b>Age:</b>	30 - 40	20 - 30	30 - 40	20 - 30 (woman)	20 - 30
<b>Your experience with software: (For example, whether you are working with it on a user level or you are developing it, experience with testing, etc.)</b>	Design and development.	I have experience with the development of user interfaces, and I have already done several similar tests or participated in them.	User only.	Experience with application development - frontend, experience with testing - both organizer and participant.	Testing, using, occasionally developing
<b>Your experience with working with the visually impaired people:</b>	Long-time with navigation design.	I am doing my bachelor thesis which is centered on blind users of mobile phones, so I would say that I have advanced experience.	I've been working with the blind since the little boy so rich.	Application development for crowdsourcing data for navigation of the blind.	none

**Table 5** High-fidelity prototype pre-test questionnaire

### Participant 1

The first participant had long-time experience with designing applications for navigating the visually impaired people.

With a few tips, he was able to accomplish all the tasks, despite the fact that some things in the prototype did not work. He didn't see the difference between the "Record a problem" and "Save" buttons for the segments, and the names of points of interest he would enter directly into the destination field.

On the prototype, he liked the connection of the street view, which makes it possible to complete the route description.

### Participant 2

Participant number two was a student who worked on a bachelor's thesis aimed at blind cell phone users. He described his experience with the visually impaired people as advanced, and he also had experience with usability testing.

Despite all the troubles and help refusal, he was able to accomplish all the tasks. He liked testing and the prototype and said that after removing implementation errors (he has experience with UI development), the prototype would be usable in normal operation.

He would welcome the street view direction of view marked on the map and the placeholder in the search because he was not sure what to enter in the field and just as everyone would expect drag and drop to modify the route.

### Participant 3

The third participant was an experienced navigational center operator who has been working with the blind since a little boy. Even though he never tested and had poor eyesight, he was very inquisitive, and the test found interesting.

He could do most of the tasks, some with a little help, but he always automatically opened a new bookmark in the browser and searched in the Google. However, we

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
<b>How did you like testing?</b>	Some tasks would deserve a better description (15.).	I liked it.	Testing was successful and interesting.	All right, it was not difficult.	The test case was very unclear.
<b>How would you evaluate the tested prototype?</b>	The prototype allows you to find a route and control the map, edit the existing route, and enrich the description. The advantage is street view integration for the possibility of completing a more detailed description.	Except for small mammoth-sized flies, I liked it all. After removing these minor bugs, I think the prototype is usable in regular service.	This project is beneficial to work and navigate people in the terrain.	It's easy to use when you orientated in it.	Useful and handy, but not very intuitive.
<b>What did you like on the prototype?</b>	Connection with street view and editing the description.	I liked a listing of existing itineraries and requests. The possibility to look on the map with the street view on the side, it comes to me as a very powerful tool in experienced hands.	Detailed processing of individual routes, including the recording of temporary obstacles.	Map and previews - street view, possibility to view existing itineraries and requirements.	Suggestions, the ability to save routes and historical requirements, a Google search engine.
<b>What did you not like on the prototype?</b>	Sometimes something does not work.	If we neglected the tiny toddler's mistakes of the implementation itself, I missed things for orientation. For example, the direction of view of street view on the map for easier orientation or labels how to perform certain actions, such as modifying the route or search format of the list of requests/itineraries.	While navigating a person by the terrain, it would be good to allow you to close each of the segments you've already checked.	Route editing - drag and drop would be more acceptable, and the route should only be done through pavements, not through houses.	Editing a route is useless.

**Table 6** High-fidelity prototype post-test questionnaire

understand that this is his habit.

Like the others, he tried to drag and drop points and would like to re-generate the route on the pavements. Temporary obstacles he would not write anywhere, but after opening the feature in the interface, he liked the idea. Furthermore, he would appreciate the option of hiding or mark segments, which he already checked to avoid changing them.

He called the prototype beneficial for navigating people in the terrain. He liked detailed route processing and recording of temporary obstacles the most.

### **Participant 4**

The fourth participant was a student (and the only woman) who developed an application for crowdsourcing data for the navigation of the blind, so she had a little awareness of what is involved in creating itineraries for the visually impaired. She also had experience with usability testing both as a participant and as a moderator.

She was able to accomplish all her tasks with just a few tips. She would like to fill the points of interest names into the start and destination field and to modify the route by dragging and dropping. She would also appreciate automatical modification of the route so that it does not lead through the houses.

She said that the prototype was easy to use once she orientated in it and most liked the map view and street view and the possibility to view already created itineraries/requests.

### **Participant 5**

At the fifth participant, it was shown that the platform could be tested by a person without any experience of navigating the visually impaired people. He described the test scenario as unclear because he is accustomed to other types of scenarios in his job as a QA tester.

The participant managed, despite a few troubles, to complete all the assigned tasks without a moderator's help.

He was the only one to look to the existing itinerary map to check if the itinerary already exists.

The biggest problem he had in editing the route, which he said was non-intuitive and useless. Just like everyone else he would like to drag and drop the points. Otherwise, the prototype has been described as useful and handy and most appreciated the whisperers, route tracking, and Google search engine.

### **5.3.3 Experiment findings and discussion**

Testing the high-fidelity prototype was successful, and the result has no longer as many reminders as its predecessors. That means that most functional requirements have been understood and accepted. The participants only suggested a few design changes and modifications to existing features.

#### **Changes in design**

1. Placeholder or tooltip for searching to let the user know the query format.

The main problem was that several participants did not know whether it was a full-text search or not. So they did not know what query format to enter, so they would welcome a placeholder or a tooltip to tell them.



2. Color the program sections and highlight the buttons.

This demand came from the operator with poor eyesight who didn't like the white color everywhere. It would be a good idea to solve this problem and to make the interface accessible even to the people who have trouble with eyesight by highlighting some parts of the program, such as buttons.

#### **Suggestions for changes to existing features**

1. Start/destination search using name of the point of interest.

All the participants first tried to enter the destination using the name of the point of interest along with the street name and expected that the whisperer would offer it to them. That did not happen, so they had to find another way to find out the address of the point of interest. It would certainly be worthwhile if the whisperer offers even points of interest

2. Show on the map the direction of view in street view.

Some of the participants had difficulty with navigating at crossings because they did not know which direction street view offered them. This problem could be solved by indicating the direction of view on the map.

3. Moving a street view marker on the map along with the street view.

4. Drag and drop to edit the route.

Some participants were so attentive and careful that they noticed that street view marker did not move on the map according to their movement in street view, so they did not know where they were on the route.

5. Closing/tagging of segments that the user has already checked.

It would be great if the segment's already checked text areas could be closed or tagged to prevent further editing and to let the user know where he terminated.

6. After editing, re-generate the route on the pavements.

The participants did not like that when the route was modified, the route did not automatically set itself on the pavements.

#### **5.3.4 Recommendations for the next phase**

For the next phases of the project, we recommend modification and extension of the interface of the designs obtained by testing each prototype. Most of them have been implemented and tested (except for the hi-fi prototype results), but there are also features that were not implemented at this time because tools and APIs did not know them yet (e.g., highlighting the part of the route that corresponds to the active segment). Therefore, we would like to welcome any initiative to resolve them.

## 6 Conclusion

### 6.1 Evaluating the achievement of the project objectives

We successfully analyzed the functional requirements and designed a platform for optimizing the workflow of the operators of tele-assistance centers for navigation of visually impaired people.

We observed in the work of operator of Navigational Centre for the Blind[1]. It helped us identify the problems in the operator's workflow that we have tried to solve. Gradually, we created a sketch, wrote down use cases, some of which we introduced using a storyboard. Then we started to design the interface itself.

The first paper mockup was created, which we consulted with the operator, followed by a paper and electronic low-fidelity prototypes that we each tested with 3 participants.

According to the results of their tests, a throw-away prototype of the web application was implemented, demonstrating most of the platform's functionalities. This prototype was then successfully tested with 5 participants who liked its design and had only a few suggestions to improve it.

The result of this project is not an application that can be deployed into production, but the processing of all issues, which can serve as a source of information for the next phases of the project.

### 6.2 Recommendation to other phases of the project

The output of this work can be used as a basis for the design of the application, which can once become an indispensable helper of operators of tele-assistance centers for the navigation of the visually impaired people.

# Appendix A

## Paper mockup

This appendix introduces other parts of the paper mockup presented in chapter 3.4.

### A.1 Layer of noise

The layer of noise which you can see on Fig. 15 is a layer showing noise pollution of streets and other parts of the city. Red means high pollution, green no pollution.

### A.2 Layer of slope

On Fig. 16 you can see the realisation of a map of the slope.

Map of the slope is visualized as a map of contour lines colored by a color of its slope e.g. red for ascending, blue for descending.

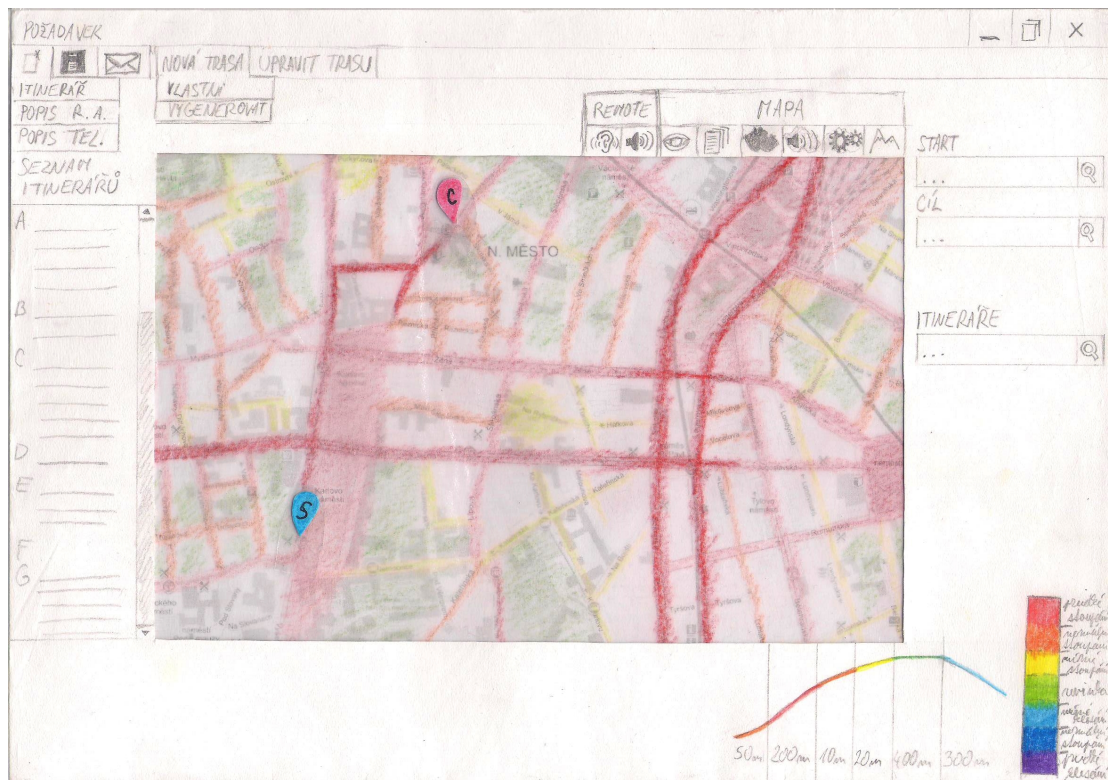


Figure 15 Mockup: Layer of noise

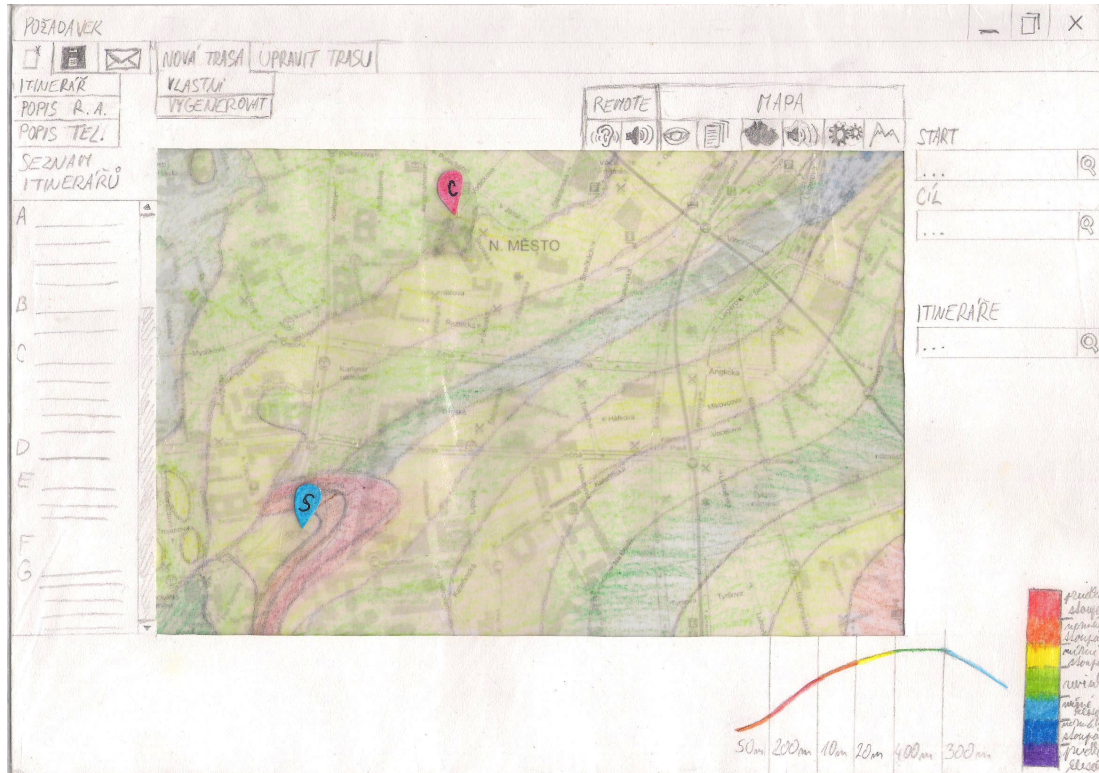


Figure 16 Mockup: Layer of slope

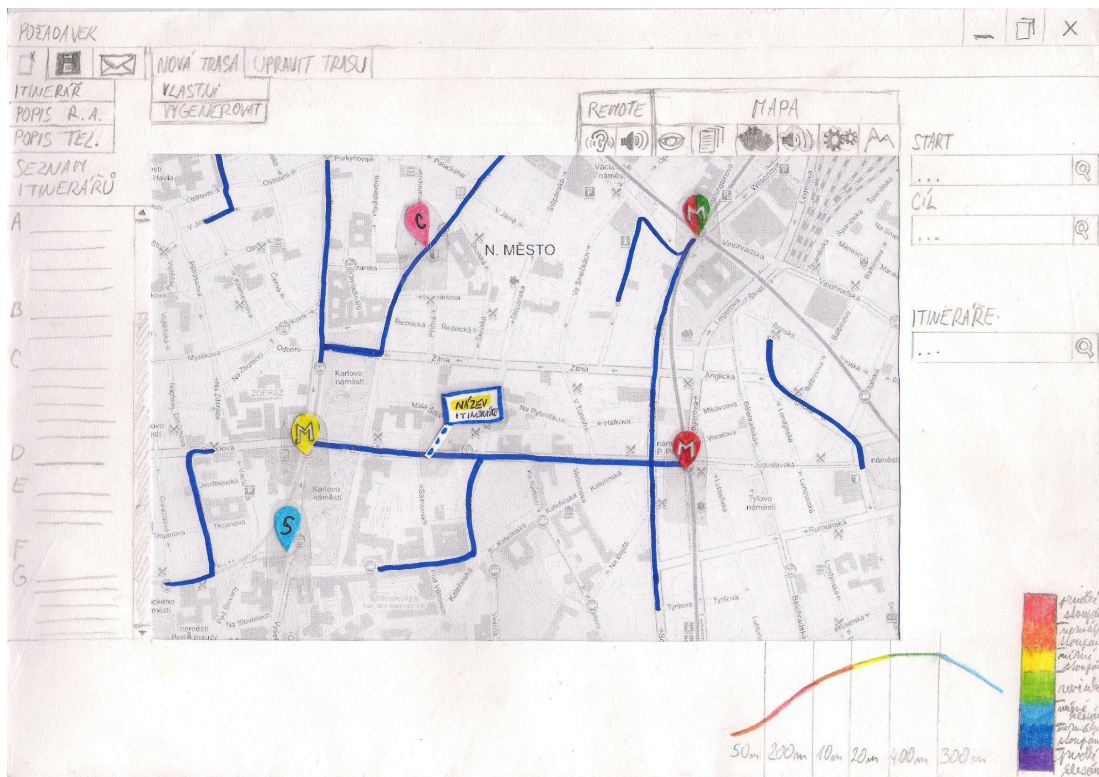


Figure 17 Mockup: Layer of existing itineraries

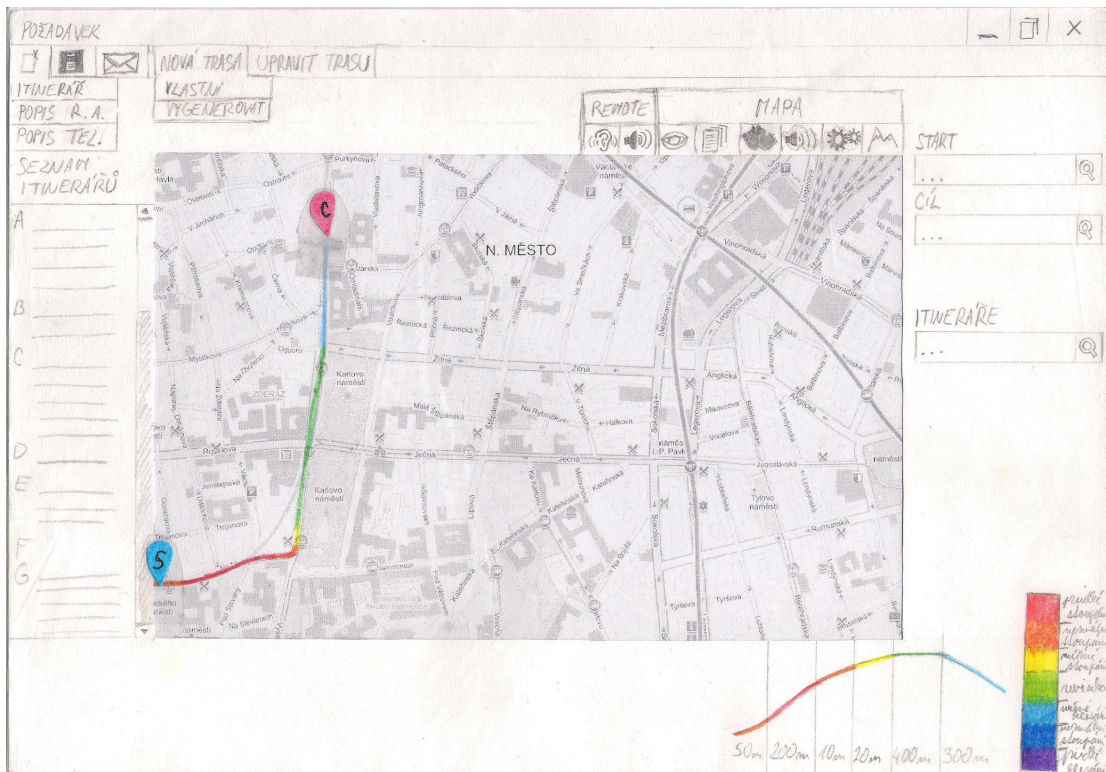


Figure 18 Mockup: Coloring the route according to slope of its segments

### A.3 Layer of existing itineraries

Fig. 17 shows a map with highlighted routes which corresponds with each existing itinerary, so the operator can look at this map if there is some itinerary which can be reused in a new one.

On the Fig. 17 are also shown markers for itineraries of the metro stations, start and destination and a label of some existing itinerary.

### A.4 Coloring the route according to slope of its segments

On Fig. 18 is another possible solution for representation of the slope of the route. Every segment of this route is colored by a color of its slope. e.g. red for ascending, blue for descending.

# Appendix B

## Paper low-fidelity prototype

On Fig. 19 and Fig. 20 you can see all parts of the resultant prototype that was used for testing with users.

The prototype consists of an A3 sized interface that has been expanded by other tools depending on the interaction with it, and of the different layers, itineraries, markers, and other content that was put into the prototype and which was prepared especially for the test scenario.

### Fig. 19 content

- The interface on the A3 sturdy paper itself
- Map tools
- Itinerary tools
- List of existing itineraries
- Map of Prague 2
- A transparent template with a particular route view
- Itinerary generated by Naviterier[8]

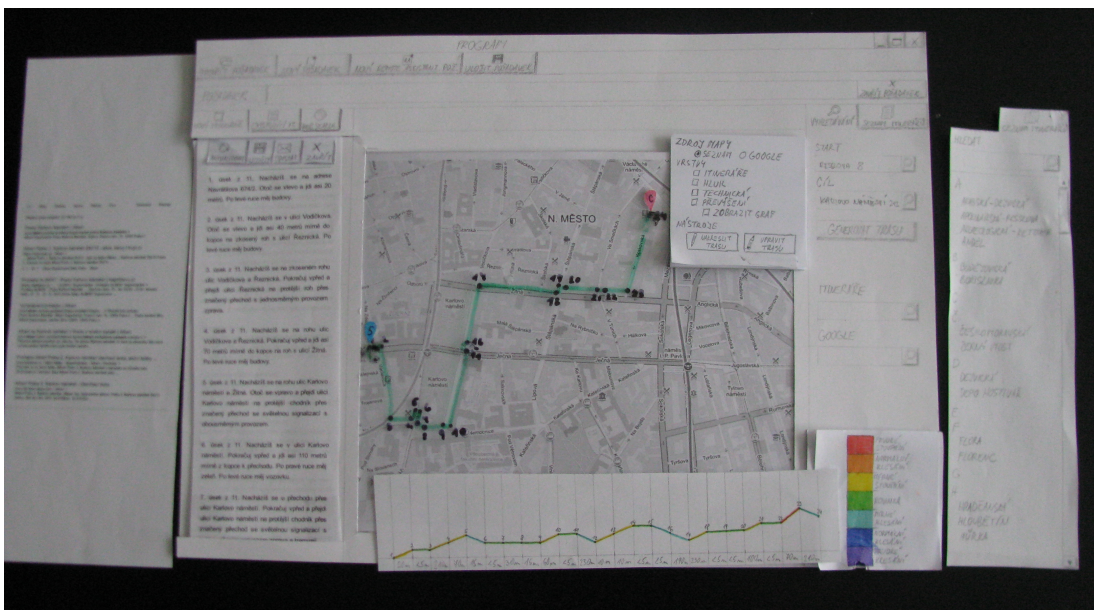


Figure 19 Paper prototype interface



**Figure 20** Paper prototype layers

- Google search results
- Graph of slope of concrete route with its scale
- Markers for:
  - start
  - destination

**Fig. 20 content**

- Street view tools
- Street view
- Layer of existing itineraries with its scale
- Noise layer
- Slope layer
- Markers for:
  - route direction in street view
  - itineraries of metro stations
- Labels of existing itineraries

# Appendix C

## Test scenarios

### C.1 Test scenario of paper low-fidelity prototype

An email from a client came. He is planning to go next week from the Resslerova 8 Business Academy to Tyflocentrum on Krakovská 21 and is asking for an itinerary for this route.

1. Create a new request.
2. Type and search for the start - Resslerova 8.
3. Type and search for the destination - Krakovská 21.
4. Generate a route.
5. Turn on route modification and change the route at your discretion so that the underpasses of the Karlovo náměstí metro station are used.
6. Set map layers of existing itineraries.
7. Evaluate, according to the itinerary map, which existing itineraries should be used and determine their age.
8. Update the generated itinerary.
9. Open the itinerary for Karlovo náměstí metro station.
10. Edit the itinerary freely (for example, passing through the metro station).
11. Suddenly, the client calls you through Remote Assistant and you are forced to stop working. Save and close the current request.

Calling your client via Remote Assistant. He is lost and needs to get to Albert supermarket on Karlovo náměstí.

12. Create a new RA request. The location of the client is automatically set to the map.
13. Find the address of Albert supermarket on Karlovo náměstí.
14. Enter the address as the destination of the route.
15. Let the program generate a route.
16. The client says he is interested in the slopes of the pavements. Turn on any of the slope description tools.



17. The client announces that the pavement is blocked by the front garden of the restaurant. Edit the route.
18. The client successfully reached the destination. Save the created itinerary.
19. Close the request.
20. Open the previous itinerary request.

## **C.2 Test scenario of electronic low-fidelity prototype**

You have a new request for an itinerary. The client wants to get out of Na Moráni 5 to Štěpánská 5.

1. Create a new request.
2. Check if the itinerary for the route does not exist
3. Generate a route.
4. Set map layers of existing itineraries.
5. Specify the existing itineraries that could be used.
6. For selected sections, consider whether they will need to be edited.
7. Edit the route.
8. Generate an itinerary for the modified route.

Another client calls - he has a meeting at the church on Karlovo náměstí, whose name he forgot.

9. Find churches on Karlovo náměstí.

... after you tell him the names of the churches, he remembers that it was Ignac.

10. Generate the route.

... The client reports that he has encountered an obstacle - the pavement repairs.

11. Solve the situation and note the obstacle in the itinerary.
12. The client successfully reached the destination. Save and close the request.
13. Open the previously stored XZ request.

### C.3 Test scenario of high-fidelity prototype

The email came from the client. Next week he meets with a friend at the bageterie Boulevard in Vodičkova street, and he needs to create an itinerary. He will come from Štěpánská 1.

1. Create a new request.
2. Find the bageterie address.
3. Make sure the itinerary does not exist.
4. Generate the route.
5. Set map layers of existing itineraries.
6. Identify existing itineraries that could be used and evaluate their age.
7. Edit the route at your discretion (for example, use part of an existing itinerary).
8. Generate an itinerary for the modified route.

Calls a client using the Remote Assistant application.

9. Create a new RA request. The location of the client is automatically set to the map.

He was headed to the Albert supermarket on Charles Square, but he lost himself.

10. Find the address of the supermarket.
11. Generate the route.

During a call, the client encounters an obstacle - the front garden of the restaurant.

12. Note an obstacle in the Itinerary.
13. Modify the route.

The client successfully reached the destination.

14. Close the request

Another request for the itinerary came. This time from Resslova 8 to Řeznická 15.

15. Make sure this request no longer exists. If so, open it.

## Bibliography

- [1] SONS. *Navigational centre for the Blind*. URL: <http://navigace.sons.cz/> (visited on 05/17/2017).
- [2] SONS. *Czech Blind United*. URL: <http://www.sons.cz/> (visited on 05/17/2017).
- [3] Remote Assistant s.r.o. *Remote Assistant*. URL: <http://www.remoteassistant.me/> (visited on 05/17/2017).
- [4] Enrico Bertini, Maurizio Rigamonti, and Denis Lalanne. “Extended excentric labeling”. In: *Computer Graphics Forum*. Vol. 28. 3. Wiley Online Library. 2009, pp. 927–934.
- [5] Jan Balata et al. “Automatically generated landmark-enhanced navigation instructions for blind pedestrians”. In: *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE. 2016, pp. 1605–1612.
- [6] Pushpak Karnick et al. “Route visualization using detail lenses”. In: *IEEE transactions on visualization and computer graphics* 16.2 (2010), pp. 235–247.
- [7] John Nicholson, Vladimir Kulyukin, and Aliasgar Kutiyawala. “ON AUTOMATED LANDMARK IDENTIFICATION IN WRITTEN ROUTE DESCRIPTIONS BY VISUALLY IMPAIRED INDIVIDUALS”. In: ().
- [8] CTU in Prague. *Naviterier*. URL: <http://www.naviterier.cz/> (visited on 05/17/2017).
- [9] Pixton Comics Inc. *Pixton*. URL: <http://www.pixton.com/> (visited on 05/22/2017).
- [10] Stanford University (USA) Helsinki University of Technology and NTNU(Norway). *ProWork Project*. URL: <http://www.proworkproject.com/prowork/design-probe.html> (visited on 05/21/2017).
- [11] LLC Balsamiq Studios. *Balsamiq Mockups*. URL: <http://balsamiq.com/products/mockups/> (visited on 05/22/2017).
- [12] Interaction Design Foundation. *Interaction Design: Concept Definition*. URL: <http://www.interaction-design.org/literature/topics/interaction-design> (visited on 05/23/2017).
- [13] Seznam. *Mapy.cz API*. URL: <http://api.mapy.cz/> (visited on 05/24/2017).
- [14] Google. *Google Custom Search*. URL: <http://developers.google.com/custom-search/> (visited on 05/24/2017).
- [15] Micromata GmbH. *Java API for KML*. URL: <http://labs.micromata.de/projects/jak.html> (visited on 05/25/2017).
- [16] Mozilla Foundation. *Document Object Model API*. URL: [http://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model](http://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model) (visited on 05/25/2017).
- [17] The jQuery Foundation. *jQuery API*. URL: <http://api.jquery.com/> (visited on 05/25/2017).

## Bibliography

- [18] Mozilla Foundation. *CSS Flex Boxes*. URL: [http://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Using\\_CSS\\_flexible\\_boxes](http://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes) (visited on 05/25/2017).
- [19] Robert Spence. *Information visualization*. Vol. 1. Springer, 2001. ISBN: 0-201-59626-1.
- [20] SC Eick, Joseph L Steffen, and Eric E Sumner. “Seesoft-a tool for visualizing line oriented software statistics”. In: *IEEE Transactions on Software Engineering* 18.11 (1992), pp. 957–968.
- [21] Maneesh Agrawala and Chris Stolte. “Rendering effective route maps: improving usability through generalization”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 241–249.
- [22] He Liu et al. “Visual analysis of route diversity”. In: *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. IEEE. 2011, pp. 171–180.