

TARTU UNIVERSITY
FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE
STUDY PROGRAM “INFORMATION SYSTEMS DEVELOPMENT“

Vladimir Suhhorukov

**DEVELOPMENT OF SOUND BEACON BASED INTERIOR NAVIGATION
SYSTEM FOR THE BLIND PEOPLE**

Bachelor's thesis

Supervisor: M.Sc. A. Säask

NARVA 2020

CONTENTS

List of abbreviations	4
Introduction	5
1. The problem and a review of existing solutions	7
1. 1. The problem	7
1. 1. 1. General information	7
1. 1. 2. Interior navigation problem.....	9
1. 2. Existing solutions	11
1. 2. 1. Simple beacons.....	11
1. 2. 2. Beacons-informers	12
1. 2. 3. Speaking City	13
1. 3. Disadvantages of existing solutions	15
1. 4. Goal of the project and basic requirements for the system	16
2. Design of the system	18
2. 1. Structure of the system.....	18
2. 1. 1. Main components	18
2. 1. 2. Dataflow	20
2. 1. 3. Use cases	20
2. 2. Functional requirements.....	22
2. 2. 1. Website.....	22
2. 2. 2. Mobile application	23
2. 2. 3. Server API.....	24
2. 2. 4. Beacons control software	24
2. 2. 5. Beacon.....	25
2. 3. Technologies	25
2. 3. 1. .NET Core	25
2. 3. 2. PWA.....	28

2. 3. 3. Arduino NodeMCU.....	29
3. Description and usage of implemented system.....	32
3.1. Kibilights.com.....	32
3. 2. Spark	36
3. 3. Lighthouse.....	38
3. 4. The link to the source code	41
Conclusion	42
Summary in estonian.....	44
References	46
Non-exclusive license	47

LIST OF ABBRIVIATIONS

API – application programming interface

CMS – content management system

CSS – cascading style sheet

GPS – global positioning system

GUI – graphical user interface

HTML – hypertext markup language

HTTP – hypertext transfer protocol

HTTPS – hypertext transfer protocol secured

IDE – integrated development environment

JSON – JavaScript object notation

NFC – near-field communication

OCR – optical character recognition

PWA – progressive web application

SSE – server-sent events

UC – use case

VPS – virtual private server

INTRODUCTION

Around 124 million of people all over the world are blind or visually impaired. Every day in their lives they face various problems. One of these problems is difficulties during navigation inside buildings, especially in unfamiliar ones.

The goal of this thesis is to design and develop a software part of a system, which would help blind and visually impaired people to navigate inside buildings without using visual orienteers and using sound beacons instead.

To achieve this goal, we need to complete the following tasks:

- Conduct a research to find out, which solutions are already available on the market.
- Create a list of disadvantages of existing solutions.
- Develop a mobile application for the end user.
- Develop a control software for the beacons.
- Develop a web server for the system administrators.

The first chapter gives general information and describes the problem itself. We try to imagine how modern IT technologies can substitute visual orienteers to help blind people navigate inside buildings. The next section of the first chapter is research into technical solutions represented on the international and Estonian market, which can to some extent solve the problem. Also, the set of requirements for a new system is described based on disadvantages of existing solutions.

The second chapter addresses the design of the new system. The new system is a navigation system for visually impaired people which uses sound beacons instead visual orienteers. A blind person can activate any beacon installed in the building using his mobile phone and get directions to it hearing the beacon's sound. Also, in his mobile phone he can obtain information about each beacon, and get detailed instructions, how to get to the most important parts of the building.

The system is intended primarily for blind and visually impaired people. We hope that government structures and other big and small organizations will be interested in it to increase accessibility of their facilities for all categories of society. We already have ramps for wheelchair users as a standard in new buildings. Hence it would be good to have sound beacon systems in public places as well, for example in colleges, schools, hospitals, government departments, train stations, airports and so on.

The second chapter also describes a structure of this system and functional requirements for each part of it. Finally, technologies, which can be used to implement the system are researched, and arguments are presented for choosing proper ones.

The third chapter describes the implemented system. The author presents information about the mobile application, the website, the server API and the beacons' control software. There are some interface examples and usage examples for users, i.e. visually impaired people, and for administrators of the system. Also, there are some conclusions about what was done well, and what could have been done better, as well as a link to the GitHub, where one can download the full source code of each part of the implemented system.

1. THE PROBLEM AND A REVIEW OF EXISTING SOLUTIONS

This chapter represents a short research of the topic considered in this work. It addresses the relevance of the problem and its solutions.

There are a few examples of difficulties blind people face in their daily life, and how technologies can help them. The main emphasis is the topic of this work: the problems in navigation inside unfamiliar buildings.

The next part describes some solutions we have nowadays, how they solve the problem, their advantages and disadvantages.

Finally, a new solution is proposed. Answers to questions, like why a new solution is needed and which are its benefits are presented. Also, the description of the new system is provided and fundamental requirements to comply with are listed.

1. 1. The problem

1. 1. 1. General information

More than seven billion people live on the Earth at the moment. According to the World Health Organization's data, ten percent of them are people with disabilities. It is the greatest minority in the world. 39 million of these people are totally blind, 124 million are visually impaired. All in all, it is more than the whole population of Russia, and it is more than 10 times the population of Estonia. Every five seconds one adult person in the world becomes blind and every minute one child does.

Modern medicine has made a very big leap forward compared to previous centuries, especially in surgery and organ transplantation. But, unfortunately, it still cannot heal everything, for example, cancer or HIV. The same thing applies when we are talking about blind people. In numerous complicated situations medicine is powerless. For example, we have cataract treatment methods, but there is still no way to treat retinitis. So, millions of people all over the world stay blind for their whole lives without hope to see again with their own eyes. And they have to adapt somehow, overcoming many difficulties every day, doing usual things, which other people will not even consider a problem.

But medicine is not the only way to help blind and visually impaired people. Modern technologies, devices and software can adapt the surrounding environment for these people, and dramatically increase their standard of living. And they already do. Thanks to a special software author of this work can actually write it himself.

That is why this topic is relevant and important, and it has been chosen for this thesis.

When we talk about people with disabilities, one of the key words is independence. For example, a blind person can almost always rely on other people who can help in daily life. But there will always be situations when it is not possible. In addition, everyone wants to be a person. So, it is very good, when there are opportunities to do ordinary things independently: use a computer, cook, do chores at home, walk around the city, find something inside an unfamiliar building.

There are many aspects of life where inventions and technologies can help a visually impaired person. We will not speak here about well-known things, like braille writing, but it makes sense to mention some devices which usual people do not face.

Nowadays there is a number of little gadgets making a blind person's daily life easier and more independent. First of all, there are different beeping, talking and vibrating things intended for home duties. For example, a talking watch, a talking thermometer, a talking kitchen scale, a beeping liquid level analyzer, a talking color detector, a beeping light sensor and so on. For example, using a light sensor blind person can understand if the light is turned on in the room (it increases sound frequency in bright spaces and decreases it in the dark), using a liquid level analyzer he can safely and precisely pour a cup of hot coffee.



Figure 1. Beeping liquid level analyzer. Source: silmalaegas.laegas.ee.

One of the most interesting and technological devices for home use is a voice marks system. It consists of NFC marks and device for voice recording. A person can stick marks onto different things (seasoning jars, DVD-disks and so on), and prerecord a voice message. After that, he can hear a prerecorded message by touching a particular NFC mark, and easily find a DVD with his favorite movie.

There are also very important and helpful software products like screen readers, which allow the user to interact with the computer, OCR technologies, which can recognize text on pictures or in PDF files consist of images without text layer which are totally inaccessible for screen readers' users.

But despite many existing devices and software, there are still a lot of problems to be solved to make visually impaired people's life better. One of them is independent navigation, particularly, navigation inside buildings, which is the topic of this work.

1. 1. 2. Interior navigation problem

The first navigation device for blind and visually impaired people was a simple white cane.

A white cane is a device used by many people who are blind or visually impaired. A white cane primarily allows its user to scan their surroundings for obstacles or orientation marks but is also helpful for onlookers in identifying the user as blind or visually impaired and taking appropriate care. (Science Direct 2020)

It can have different length depending on user's height. A white cane can be collapsible or straight with a telescopic mechanism.

To scan the area for obstacles using a white cane person moves it from left to right and vice-versa in front of him keeping the far end on the ground. Some people during using this tool are also actively hitting it to the ground. The reflected sound helps them to understand type of material in front (grass or asphalt) or even approximately imagine size of the room they are in.

A white cane is still the most reliable and simple tool for visually impaired people. It can be used equally well inside and outside of buildings. But it allows to scan a very limited range. Because of that a white cane is very good to move by earlier learned routes or in places with special orientation marks on the floor like some airports. It does not suit for

exploring new routes, large areas without clear orienteers and to explore an unfamiliar building. For these purposes a blind person still needs help from other people to walk by route couple times and learn the orienteers. Otherwise it will be not efficient or may even dangerous.

There are some digital solutions to solve this problem outside buildings, to walk around the city, find addresses and navigate to them. For example, Seeing Assistant Move. It is an application for IOS. It uses GPS and Google Maps (for turn-by-turn navigation). The key feature is ability to remember coordinates of any point and show directions and distance to it. User just holds his mobile phone, and the device starts to vibrate, when the smartphone is turned to right direction. So, user can either walk through some route one time using other people help, save key points into the mobile phone and receive directions to them later, or even just select address he needs, and the system builds a route automatically. This means that the application allows a blind or a visually impaired person to some extent independently orientate in the city, including places he has never been to, or even walk in the forest.

It would be great to have something like this to navigate inside buildings as well. GPS signal is not available here, and it does not feel altitude.

In a common situation, when a person gets into an unfamiliar building, he has various orienteers. First of all, he can see doors, stairs, elevators. In addition, often there are schemas illustrating what and where is situated, and there is ability to use arrows and text signs (“Exit”, “WC”, numbers of rooms or auditoriums). A blind person in the same situation can only count on help of somebody, because, as we mentioned before, a white cane is useless and dangerous here. Very rarely, for example in the airport, there is a relief road on the floor leading to some key points, and almost always it leads to special assistant services.

So, we need something can substitute visual orienteers inside a building to make it available to blind and visually impaired people.

The most logical idea is to use sound. Listening to surrounding sounds is a basic way for blind people to find orienteers. They use it every day. Walking around the city, scanning the closest space with a white cane, they listen and remember specific sounds for each area: birds, mechanisms, cars and so on, and this information can help to navigate in these

places in future. It is possible to use the same approach inside a building, but again, the building has to be explored before, and it will no work in an unfamiliar place.

So, we need some sound based guiding system, some beacons which can become pointers and signs for blind and visually impaired people inside an unfamiliar building.

We already have something like this on the streets of our cities – ticking and beeping traffic lights. They not just tell a blind person, when he can cross the road, but they show, where a crossing actually is. Of course, there already are some solutions for sound based navigation inside buildings. We will take a look at a couple of them in the next section. We will see what can they do, what opportunities they offer and what are the disadvantages. And after that we will offer a new solution keeping in mind the disadvantages of existing systems.

1. 2. Existing solutions

1. 2. 1. Simple beacons

This is the simplest solution which uses the ticking traffic lights' approach.

It is a device installed for example above a room's door or entrance of a building and it produces sound constantly or by user's request. It can be powered either by electricity outlet or by batteries. If a beacon has the opportunity be called upon by the user, it has its own remote control device, one per each beacon.

Using this system a blind person can find, for example the entrance of a building or a specific room, an auditorium and so on. For example, these beacons are installed in some Estonian blind societies to indicate entrance of a building.



Figure 2. Sound beacon Landsort. Source: *silmalaegas.laegas.ee*.

These systems are represented in Estonian market by a few models. For example, sound beacon Landsort or Hällö. Prices are from 80 to 1300 euros.

One of the examples on the market is to use a beacon with remote control to find your hotel room. User can stick a device above his room's door. Next time, when he gets to the floor, he can call the beacon with the remote control and find a door by sound.

1. 2. 2. Beacons-informers

These systems are usually used for providing a blind or visually impaired person with information about the area he finds himself in.

The system consists of two parts. The first part is the beacon itself. It can produce sound and stores prerecorded information about an area where it is installed. The second part is the user's device. Using this device the user can receive information from a particular beacon and force it to make a sound to get a right direction to it. There can be a lot of such beacons in a building. User's device will interact with closest of them.

It is one of the most popular approach used for all over the world. One of the examples is the navigation system for visually impaired people Parus.



Figure 3. Bracelet Parus. Source: ucentr46.ru.

It consists of a base block and a bracelet-activator. Base blocks are installed in important parts of a building and store prerecorded messages with information like description of

the area, or how to get somewhere. When the user with a bracelet gets to a base block's area, the bracelet starts to vibrate indicating that there is beacon nearby, and the user can receive some information. The user has four buttons on a bracelet, each button for one particular prerecorded message. The message plays through the beacon's speaker, and everybody can hear it. Base block works from building's power network, a bracelet is powered by batteries. Communication is via a radio channel. A bracelet's price is about 100 euros, a base block costs about 300 euros.

Next example is the silent sound beacon for the blind Mobilis. It is similar to the Parus system, but instead a bracelet user has a mobile application. Mobile phone connects to a beacon via Bluetooth. The beacon starts to make a sound when user is about 15 meters away and stops doing it when the user goes away. Also, the user can listen to a prerecorded message using his mobile phone headset. The price is about 100 euros.



Figure 4. Beacon Mobilis above an entrance. Source: mobilis.prom.ua.

1. 2. 3. Speaking City

Speaking City is a big and complex solution, which cannot be attributed to previous categories. That is why this information is placed in a separate section. The system is developed in Russia, and nowadays it works in testing mode in several cities.

In the center of the system is a carriable user's device, which connects to many other devices in different places. The system covers a number of aspects connected to moving around the city. The system tries to be a universal solution for blind people's navigation. Special devices-beacons are installed in different places like public transport, traffic lights, outside and inside buildings.

For example, when a visually impaired person gets close to a traffic light, first of all he can hear the usual ticking sound, changing when it is allowed to cross the road. In addition, using his personal device and its headset he can listen to information like where this particular traffic light is situated, the name of the street the user is going to cross and so on.

Another example is when a blind person is standing in a bus stop. If a bus is arriving, and it has the Speaking City module on board, the user with his device can listen to information about the route's number, from and to where it is going, is it a bus or a trolley bus. If it is a bus he needs, the user presses a button on his device, and it activates sound beacons above the bus door.



Figure 5. Speaking City carriable user's device. Source: www.speakingcity.org.

And, of course, this system covers a problem from our topic. When a visually impaired person gets into a building where the Speaking City is installed, he can receive information from beacons in the building, such as the location of a door or the beginning of a corridor, is it a door to a WC or to an accountment department, in which direction

the door opens and so on. Afterwards the user can force a beacon to make a sound to use it as an orienteer for navigation.

Speaking City user device can be purchased only by Russian citizens having documents confirm their disability. Because of this it is hard to provide device's price here. And it is obvious that other components of the system for buses, buildings and so on are quite expensive.

1. 3. Disadvantages of existing solutions

We will define now some basic disadvantages of existing solutions in the context of the described problem and in the context of life in Estonia.

If we talk about simple solutions described in section 1.2.1, there are two main problems:

1. Most of these beacons are intended to work constantly, indicating for example the entrance to a building. It means that the constant sound will annoy the usual people who do not need these beacons. For the same reason, we cannot organize fully a functional guiding system, because a lot of sounds from everywhere will be confusing.
2. These systems have no opportunity to control it from the user's side, and if it has, each beacon needs its own remote control device.

In conclusion we can say that solutions from this category do not suit for organizing a navigation system for blind and visually impaired people inside an unfamiliar building which can to some extent substitute visual orienteers. It can be used only for indicating a couple of key places in a building, for example, the main entrance.

Solutions from section 1.2.2. are more functional and flexible in use, but they have disadvantages too:

1. Most of them are connected to a building's power network. It will cause problems during installing these devices in larger numbers.
2. To be able to use systems like this, a visually impaired person has to buy additional devices.

So, despite the functionality of these beacons (for example audio messages with the description of the current area) it is impossible to mark each door inside a building with them. It will be very difficult to install and also very expensive.

Speaking City seemingly looks good. But it is a very complicated and expensive system which tries to solve a wide range of problems at the same time. To install and run a system like this we need effort from at least the local government. In addition, the user has to buy special devices using government support program for people with disabilities, otherwise it will be very expensive.

And here some general disadvantages of the described existing solutions:

1. It either too simple or too complicated and is not suitable for covering most points of interest inside a building and making them controllable from the user's side.
2. Almost always a visually impaired person has to buy an additional device and carry it with him.
3. Most of the solutions are rather expensive for both blind people and the facility that wants to install the system.
4. Most of systems are not available on the Estonian market.

Keeping in mind these disadvantages, in the next section we will define the basic requirements for a new system which could become an alternative.

1. 4. Goal of the project and basic requirements for the system

Now we will formulate the goal of the project and define basic requirements for the offered solution.

Once again shortly about the problem we have described above.

Blind and visually impaired people have no opportunity to use visual orienteers for navigation inside a building like usual people have. So, it is extremely hard, not efficient and not safe to move around unfamiliar buildings. Also, difficulties may arise even in before explored areas, for example, a college building has many auditoriums on each floor, and it is hard to remember where exactly a particular one is.

The goal of the whole project is the development and implementation of the system, which can partly substitute visual orienteers with sound beacons which can be activated by users and which will give a direction for the users to different parts of the building's infrastructure and provide them with information about the current area and how to get to the desired point.

As this is a big and ambitious project, it is not possible to complete it within this thesis alone, so the goal of this thesis is developing and implementing only a software part of this project.

Main requirements for the system:

- The system must have sound beacons that can be activated by the user and be silent when there is no need for them.
- The system and sound beacons must be designed in a way that it is possible to use a lot of beacons to cover most of possible points of interest in a building.
- The system and the beacons must be cheap and easy to install.
- The system must provide the user with information about each beacon: what does it indicate and some additional useful details.
- The system must be able to lead the user to key areas of a building from beacon to beacon.
- The beacon's sound must be not annoying for other people and should not be like an alarm signal.
- Blind or visually impaired people do not have to buy any additional devices for using the system, just only have a smartphone with them.
- All user interfaces of the system (website, mobile application, software for system administrator) must be fully accessible and convenient to use for screen readers' users.

A screen reader is a form of assistive technology (AT) that renders text and image content as speech or braille output. Screen readers are essential to people who are blind, and are useful to people who are visually impaired, illiterate, or have a learning disability. Screen readers are software applications that attempt to convey what people with normal eyesight see on a display to their users via non-visual means, like text-to-speech, sound icons, or a braille device. (MDN Web Docs 2020)

2. DESIGN OF THE SYSTEM

This chapter describes the design of the system and key considerations of the development plan.

Firstly, the information about the main parts of the system, and how they interact with each other is given. Then there are functional and nonfunctional requirements for each component of the system. Finally, there is a short review of available technologies we can use to implement the system and arguments why we will choose a particular one.

2. 1. Structure of the system

2. 1. 1. Main components

Using basic requirements described in section 1.4. we can now offer a structure for the system and define its main components.

It will be a rather complex system. We have to find a way to connect the user's mobile phone with each beacon to have the opportunity to activate them separately. In addition, besides a sound signal, the user must be able to get detailed information about each beacon. At the same time, we should keep beacons much more simple and cheap. So, we need some middleware which receives requests from user, controls beacons, stores data and provides the user with information. Also, it would be good, if there would be a resource with general and additional information about the system, like user guides.

Based on the foregoing, the system may consist of the next parts:

- Mobile application
- Website
- Server API
- Beacons control software (local server)
- Beacons

Now we will shortly analyze each part separately.

The website is the central part of the system. Firstly, all information about the project and the system is published here: description, instructions for installation of mobile application, user guides. The user can install the mobile application from here. Also, here is a control panel for the system's administrator, where he can manage user accounts and

information about beacons and facilities where they are installed. All data and server API are situated here too.

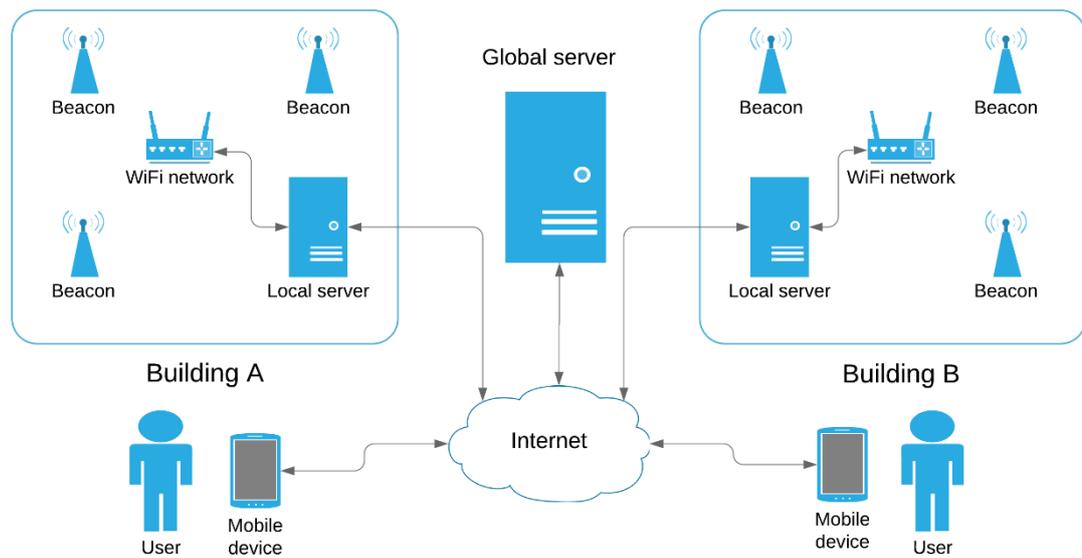


Figure 6. Schema of the system.

The mobile application is the main tool for blind or visually impaired people to use the system. It can be installed from the website. Through the mobile application the user receives information about the facilities where the system is installed, about beacons and routes. And, most importantly, from here the user can activate a particular beacon.

The server API connects the mobile application and the beacons control software. It receives requests from the mobile application and calls a particular facility's beacons control software sending it information about the beacon which must be activated.

Beacons control software is a special application located in a building where the system is installed. This application must be installed on machine which has constant access to the internet to be able to receive a request and activate a particular beacon.

A beacon is functionally the simplest part of the system. Since all information about beacons is stored on a server, and user can obtain it from there, the only task of a beacon is to react to a command from the beacons control application and make a sound.

It is a short description of the system's parts. More information about the functionality of each of them and implementation details is provided in the next sections.

2. 1. 2. Dataflow

Now we take a look at a couple examples of the dataflow to better understand how the system works.

The first one is from administration panel to a database. It is one of the most important pathways. Here the system's administrator using his special section on the website fills in all information about a facilities where the system is installed, about beacons and routes, including description and hints for user (this is the information the user will see in his mobile application), and some technical information needed for the system to work.

Dataflow from the user to the database. It is a registration process user has to pass on the website. To use a mobile application the user must have an account in the system. It is needed to avoid third party use of the system's server API.

Mobile application – server – mobile application. Using his mobile application a blind or visually impaired person can receive information about the facilities where the system is available, about beacons with its description, about routes and so on.

And the most important one which represents key functionality of the system.

Mobile application – server API – beacons control application – beacon. When user wants to activate a particular beacon, the mobile application generates a request with information about the facility and the particular beacon which must be activated. This request goes to the server API. After that the server sends the request to the particular beacon control application. And finally, beacon control application activates the particular beacon. The beacon makes the sound for a few seconds giving direction to user.

This chain of messages from the mobile application to beacon may looks quite long and complicated, but there are some technical reasons to make so, and this variant was considered as the best for this system. More technical details can be found in the next sections.

2. 1. 3. Use cases

To more clearly understand how the system will work here are some key use cases.

Firstly, we will define the system's user categories.

- User. It is a blind or visually impaired person who will use the website and the mobile application.
- Administrator. Depending on the situation, it could be either one person who manages website content, manages information about facilities and beacons, and sets up beacon control software, or this position could be divided in the next way:
 - Administrator of a website. Manages a website content, user accounts.
 - Facility administrator. Representative of organization which has installed the system in its building. Fills in an information about the facility in the system: beacons description, routes and so on.

UC 1. Facility administration

Purpose: Fill in information about the facility where the system is installed, description of beacons, fill in information about routes to key places in the building.

Primary user: Facility administrator

Preparatory circs: User is registered and authorized in admin role.

1. User accesses the website using a desktop computer and logs in using his admin account.
2. User selects the facility he needs from the list.
3. User adds a new beacon: he fills in “Name” field (for example, “Auditorium 321”), fills in “Description field” (for example, “Second floor. Computer classroom.”) and clicks “Add” button.
4. User adds a new route: he fills in “Name” field (for example, “Main entrance – auditorium 321”), fills “Description” field (for example, “Standing near main entrance, activate west exit beacon. There is a stairway near the west exit. Go upstairs activating the 2nd floor entrance beacon, then the 3rd floor south entrance beacon. After reaching the 3rd floor, activate Auditorium 321 beacon.”) and clicks “Add” button.
5. User opens newly created route and adds to it appropriate beacons.

UC 2. Beacon control software configuration.

Purpose: Configure a beacon control application to make it able to receive commands from the server and activate beacons.

Primary user: Facility administrator

Preparatory circs: No

1. User runs up the beacon control application.
2. User edits a configuration file.
3. User registers the application as a service in the operation system.

UC 3. Registration in the system.

Purpose: Register in the system to be able to use the mobile application.

Primary user: User (blind or visually impaired person)

Preparatory circs: No

1. User accesses the website using his desktop computer or his mobile device.
2. User clicks “Register” link.
3. User fills in the next fields: “Email”, “Password”, “Confirm password”.
4. User clicks “Register” button.

UC 4. Activating a beacon.

Purpose: Activate desired beacon to move in its direction.

Primary user: User (blind or visually impaired person)

Preparatory circs: User is registered in the system and has the mobile application installed on his device.

1. User opens the mobile application on his device.
2. User selects the facility he is currently in and gets a list of beacons.
3. User selects the beacon he needs and clicks “Call” button.
4. If user loses direction to the beacon, he calls it again.

2. 2. Functional requirements

In this section we will define functional and nonfunctional requirements for each component of the system. After that we can select technologies and proceed to the implementation of the system.

2. 2. 1. Website

Functional requirements:

- The website must contain general information about the system.
- The website must contain instructions for installing and using the mobile application.

- All information on the website must be provided in English, Estonian and Russian.
- The website must have a registration mechanism for new users.
- Vulnerable parts of the website must be protected from third party use.
- The website must have an admin section with next functionality:
 - Managing users.
 - Managing facilities, beacons, routes and all connected to it information.
- The website must have an adaptive design to be correctly displayed equally well on a desktop computer and on a mobile device.
- The website's interface must be fully accessible and convenient for screen readers' users.

Nonfunctional requirements:

- The website must be constantly accessible from the internet.
- There must be adequate response time on users' actions.
- Vulnerable data, like the users' passwords, must be stored in encrypted form.
- The structure of the website and its source code must be flexible and easily extensible.

2. 2. 2. Mobile application

Functional requirements:

- It must be possible to install the mobile application on a user's device.
- The mobile application's user interface must support at least English, Estonian and Russian.
- The mobile application's interface must be adapted for mobile device screen.
- The mobile application's interface must be fully accessible and convenient for screen readers' users.
- The mobile application must provide the user with information in which country, in which city and in which facilities the beacon system is installed and available for use.
- The mobile application must provide the user with detailed information about beacons and routes to most important parts of the building.
- The mobile application must be able to activate a particular beacon.
- The mobile application must use user's account to connect to server API.
- The mobile application must have a settings page to give the user opportunity to change some key things, like the language of the interface.

Nonfunctional requirements:

- The mobile application must be stable and not crash, for example, when the internet connection is lost.
- The structure of the mobile application and its source code must be flexible and easily extensible.

2. 2. 3. Server API

Functional requirements:

- The server API must have methods which return information about facilities, beacons and routes.
- The server API must have methods which return information about which facilities with beacons control software are currently online, work properly and ready to receive activation signal for a particular beacon.
- The server API must have a mechanism which allows the beacons control software to connect to it and wait for an activation signal for a particular beacon.
- The server API must have a method which can receive information from the mobile application about a particular beacon, which must be activated, and pass this information to appropriate beacons control software.

Nonfunctional requirements:

- The server API must be constantly accessible from the internet.
- Methods of the server API must be protected from third party use.
- The structure of the server API and its source code must be flexible and easily extensible.

2. 2. 4. Beacons control software

Functional requirements:

- The beacons control application must be cross-platform.
- The beacons control application's user interface must be fully accessible and convenient for screen readers users.
- The beacons control application must be able to connect to the server API, to receive a beacon activation signal from it and to activate the appropriate beacon.

- If the beacon control application loses connection to the server API, it must keep trying to reconnect.

Nonfunctional requirements:

- The application must use multithreading to be controllable even if it is currently trying to connect to the server API or is trying to call a beacon.
- The application must work as a service in operation system to be able to run again, if it had crashed, or after operation system's reboot.

2. 2. 5. Beacon

Functional requirements:

- The beacon must be able to receive a signal from beacons control software and make the sound for a few seconds.

Nonfunctional requirements:

- Beacon's construction must be as simple as possible.
- The beacon must work autonomously from a battery as long as possible.
- The beacon's sound must be not annoying for other people and should not be like an alarm signal.

2. 3. Technologies

Since we have now basic understanding about how the system will look like and work, it is time to get closer to the implementation, and choosing the technologies we will use in the development of each part of the system.

2. 3. 1. .NET Core

The website and the server API are the central points of the system, which connect all the system's parts together, provide the mobile application with data and send activation signals to beacons control software.

When we talk about the development of webpages with dynamic content, besides HTML, CSS and JavaScript, first PHP language comes to mind. It is a one of the most popular technologies in the world to quickly make a server application, which works with user's

data and stores something in a database. Nowadays, more than 80% of websites work on PHP.

Firstly, this technology has a very low entry threshold. A few weeks of learning PHP is already enough to start developing some functional things. If we add here the using of CMS systems, like WordPress or Joomla, written on PHP, which you can use without having programming skills at all, PHP is the best way to start developing websites for beginners.

But this simplicity has another side. It is very easy to start writing code on PHP, but at the same time, it is very easy to start writing bad code on PHP, which will be very unsafe, and projects like these can be easily hacked. CMS systems partly solve this problem. But because of its popularity, hackers always try to find a new way to hack it, and developer must always update his CMS to the latest version, and often it results in bugs.

So, in order to use PHP in complex projects, like we have, and provide high level security, the developer must be very experienced, or we have to use some additional frameworks, which always have own limitations and problems. In addition, besides the website, we have the mobile application and beacons control application, and it will be very useful to use similar technologies in its development as much as possible, but PHP specializes only on websites. So, we need to consider other options.

Another option is to use Java programming language. Java is a very powerful technology which has proven itself in time. It is a cross-platform technology which you can use to develop desktop applications, mobile applications and complex server applications. Nowadays most business IT solutions work on Java.

But, in our case, this solution has its disadvantages too. Like in PHP, to make a high quality product on Java, the developer must be very experienced. Developing of a server application with Java needs a set of additional frameworks and libraries, like Spring Boot. In addition, a developer must be competent to set up working environment for application, install, for example Tomcat server, and edit a bench of configuration files. This problem can be partly solved using Java IDEs, which can automate a lot of development and deployment processes. But Java IDEs have considerable accessibility problems for screen readers users. As we mentioned before, the author of this work (and he is the leading developer of the system) is a visually impaired person, and he actively uses screen readers in his work. Because of accessibility problems efficient using of IDEs becomes

impossible, and too many things have to be done manually. So, Java language is not the right choice either.

IDE – “Integrated Develop Environments (IDEs) are software packages that attempt to provide comprehensive support for coding, testing, and debugging” (Zeil 2017).

There are additional relatively young and fast developing technologies that can be used for website and server API implementation, for example Python and Node.js. But IDEs intended for working with these technologies have the same accessibility problems as Java IDE has. In addition, to develop a complex application, the developer must put together a bench of libraries for different purposes, using appropriate package manager. Most of these libraries are developed by different teams, so its documentation is very scattered in the internet, and has different quality. In the author’s opinion, Python and Node.js suit for development of applications with microservice architecture, but our website and server API to have a monolith one.

Another relatively young, but proven technology is a .NET Framework from Microsoft and its last iteration .NET Core. It is a very powerful platform for the development of desktop applications, server applications and even mobile applications. And, starting from .NET Core version, it is now truly cross-platform.

Main programming language in .NET is C#. The platform has already on board all libraries to solve almost all common tasks. Additionally, the platform has a set of frameworks which contain tools and libraries for the development of specific systems, for example Windows Forms for creating GUI applications for Windows desktop, Xamarin Forms for mobile applications and ASP.NET for creating websites and server API applications. And, of course, if you need something special, there is the package manager with thousands of libraries from different developers.

The next important thing in our case is the accessibility of IDE for screen readers’ users. The main IDE for .NET is Microsoft Visual Studio, its Community version is free to use for students and single developers. This IDE has the highest level of support of screen readers. It allows visually impaired developers to efficiently and quickly develop and debug their applications.

So for the website and the server API development we will use ASP.NET Core, because it is a very powerful framework, which out of the box has almost all tools for creating

complicated and secured websites and server APIs, and which can be relatively easily installed on common VPS service with Linux.

Beacons control software will be implemented with .NET Core too. It will be a fully cross-platform simple console application, which can work on any machine whether it is a Windows or a Unix-like system installed there.

And in addition, there is one more problem which .NET will help to solve.

Beacon control software in the system must receive signals from the server, when and which beacon must be activated. It is not easy to implement using common HTTP protocol, which is a standard for internet communication. In HTTP protocol server can only answer to client's requests, and server cannot send any information to any client in any time, if the information was not requested. But we need here vice-versa behavior here: server must send the information about beacon need to be activated to particular client at a specific time. To achieve it, several techniques can be used, like Long Polling, SSE and Websockets. All of them are quite tricky, and we will not describe mechanisms of their work here. But for .NET Core Microsoft has the SignalR library. It hides all protocol's transport details and provides simple methods to establish realtime message exchange between server and client. So, we will use it in the server API and in beacons control application.

2. 3. 2. PWA

Nowadays there are two major operation systems for mobile phones: Android and IOS. And there are several approaches how to develop applications for mobile devices.

The first approach is to develop native applications for each operation system. It means that we use different programming languages and development tools for each particular system. For example, in Android we will use Java programming language, and in IOS we will use Swift or Objective C. In case of IOS we have to use even more specific hardware and IDE. It is impossible to compile IOS application on Windows or Linux, only computer with MacOS suits for this.

This approach has both obvious advantages and disadvantages. From one hand in this case we will get the most stable application, which uses all features of the operation system and device's hardware. But from another hand, we have to implement two applications with the same functions instead of one, and in some cases purchase additional

equipment. In order to save time and funds, we will not use this approach in the current project.

Another way is to use one of cross-platform technologies. One of them was already mentioned before. It is Xamarin Forms for .NET Core. It allows writing the application's source code on C# language, and then it will be compiled for each platform. But, because of the lack of direct access to the operation system's features, for GUI it uses some standard components, which can be displayed a little bit different from device to device. For the same reason, it is hard to guarantee the same accessibility both for IOS's and Android's screen readers. In addition, a machine with MacOS is still needed to compile an application for IOS. So, currently this technology does not suit the project either.

Lately developers have gotten one more tool to create mobile applications. It is PWA technology. This approach allows developing mobile applications as a common web page using HTML, CSS, and JavaScript adding specific modifications in the source code.

These modifications allow the user to save the webpage to his home screen. Though it still uses a browser in its work, the application now looks and behaves like a native application: it has its own icon on home screen and there is no browser's GUI elements when it started up. Because of using HTML and CSS for application's GUI, it is possible to easily make the same looking interface for all platforms. And, the most important thing for this project, it gives us the opportunity to provide maximum accessibility for screen readers on both IOS and Android.

Considering all these facts, PWA technology will be the best solution for the system's mobile application for now.

2. 3. 3. Arduino NodeMCU

As it was mentioned before, the beacon must be the simplest part of the system. All the logic and data must be implemented in the mobile application, in the server API and in the beacons control software. So, a beacon must just be able to receive an activation signal and react to it.

It is a quite flexible system. The system's architecture like this, allows us to use a wide range of technologies developing a beacon's construction. And, if we choose a different beacon, there is no need to redesign other components of the project, except minor modifications of beacons control software.

For example, a beacon may consist of a battery block, some beeper to make the sound and a radio module for receiving signals via radio channel. In this case, the beacon control software must be installed on some device with a radio transmitter, programmed to send signals to different beacons. It could be any single board computer, for example, Raspberry Pi.

Summarizing this section, Table 1 represents all parts of the system and which technologies will be used to implement each of them.

Table 1. Technologies used in the system implementation

Component	Technology	Programming language
Website and server API	VPS server with Ubuntu, ASP.NET Core, Microsoft SignalR	C#, HTML, CSS, JavaScript
Mobile application	PWA	JavaScript, HTML, CSS
Beacons control software	.NET Core, Microsoft SignalR	C#
Beacon	Arduino NodeMCU	LUA

In this particular project, the beacon will work via a local wi-fi network. Each of the beacons will have its own static IP address. Beacons control software will have inside of it a table with beacons' IDs and corresponding IP addresses. When the server sends to the control application a message with beacon's ID, it sends a request to the IP address of the corresponding beacon, and the beacon activates. In this schema we do not need to use any additional equipment. Beacons control software may be installed on any computer in a local network constantly connected to the internet (it could be a single board computer too).

To test the system in this particular case as a platform for a beacon's construction we will use Arduino NodeMCU board. It is a cheap and high-quality solution that is widely used, for example in smart house controlling systems. It has a wi-fi module on board and can

equally well be programmed as a server and as a client, which can receive requests and send back answers. In this case, it is what we need.

3. DESCRIPTION AND USAGE OF IMPLEMENTED SYSTEM

As of the date of this document been written, the system's prototype is ready to be tested. All parts of the system have its basic logic and functions. So, we can now take a look at it, explore the user interfaces of the system's components, check out its functionality for users and administrators, and generally understand how it works.

Doing this we can also give the first evaluation of what has been done. Which parts were implemented successfully, and what could have been done better, and if components of the system comply with requirements described in the previous chapter.

3.1. Kibilights.com

The website is the core of the system. It contains all information about the project, user guides, opportunity to install the mobile application, information about users, about facilities and beacons, admin panel, server API which connects the mobile application and beacons control software.

In order to develop and deploy the website, a domain name has been purchased, and has been rented a VPS server. Now the website is available at kibilights.com. There is an Ubuntu operation system installed on the server with all components needed for ASP.NET Core application works.

The website's design has a dark scheme because a light text on a dark background is more readable for visually impaired people, who can still use magnification software. And of course, the website is totally available for screen readers.

The website is not finished yet. So, some information is still missing there, and some elements will be changed in future. But it already has basic functionality which allows the system to work and be tested. Some elements are described as follows.

According to the functional requirements the content on the website is represented on three languages: English, Estonian and Russian. During the first visit, the system chooses a language automatically, depending on several factors, like user browser's settings, geo positioning and so on. After that the user can change language at any time using appropriate buttons on top of the webpage.

The welcome page has a short description of the project and instructions about how to use the system. Also, there are links for login, registration and installing the mobile application. If the user is logged in as an administrator, in addition there is a link for opening the admin section of the website.

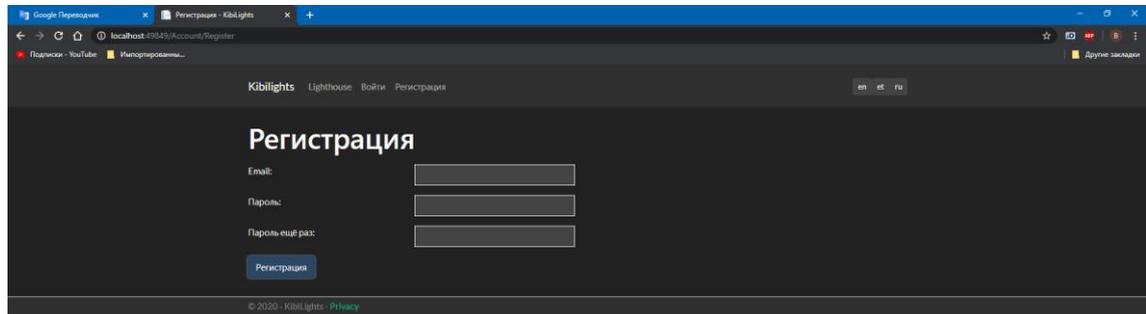


Figure 7. Kibilight.com registration form on Russian language.

As it was mentioned in previous chapters, to use the system the user must be registered in order to avoid a third-party use of server API. So, before installing the mobile application, user should fill in a simple registration form consisted of three fields: email, password and password confirmation. Users passwords are stored in the database in a encrypted way. After login, the system authorizes users using cookie files containing encrypted information about the user and his role.

A very important section of the website is the admin panel. It is intended for the main administrator of the website and for the administrators of facilities, if they want to manage information about beacons in their building by themselves.

In user control section administrator of the website can see information about all users registered in the system. Also, he can delete any user he wants, or create a new one. In creating a new user, administrator can set a role for new user, including a role for special users. These special accounts are used by beacon control software to get access to server API.

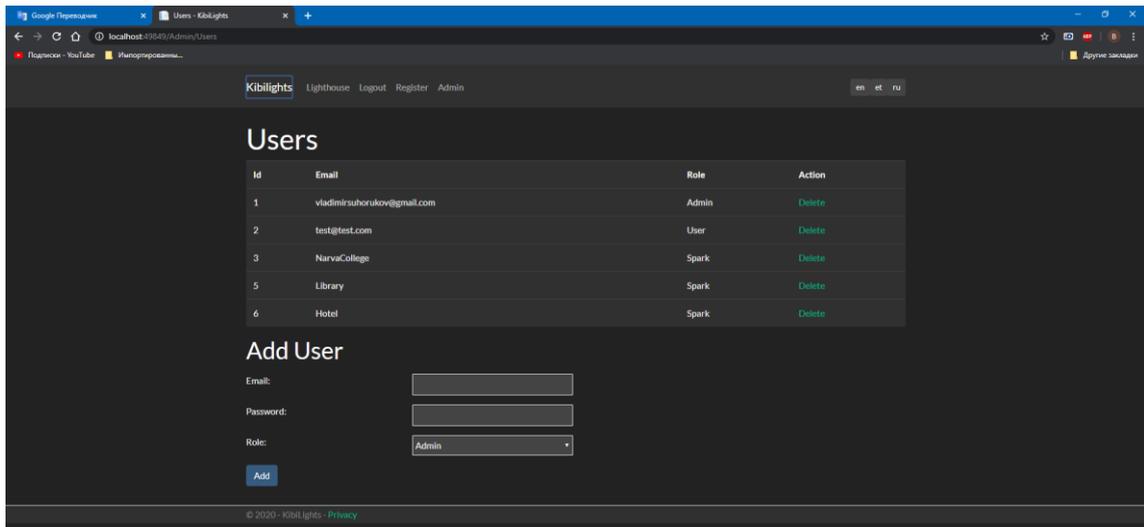


Figure 8. Users control panel.

Facilities control panel (Figure 9) is intended to manage facilities in the system. Each row in the table represents a building or organization where the beacons system is installed.

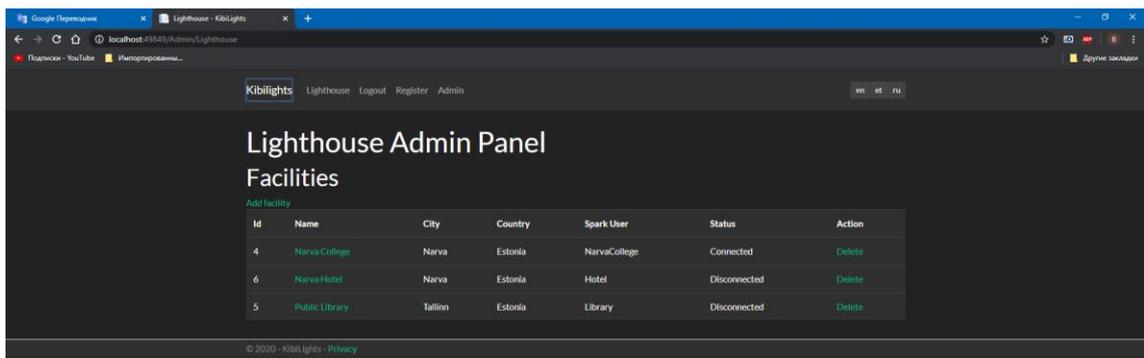


Figure 9. Facilities control panel.

Here is information about the facility's name, city and country. Also, the table shows the status of each facility – is it currently connected to the system and ready to work. And there is the possibility to delete any of the records or create a new one as well.

When the user clicks on a facility's name, he goes to beacons control page of this particular facility.

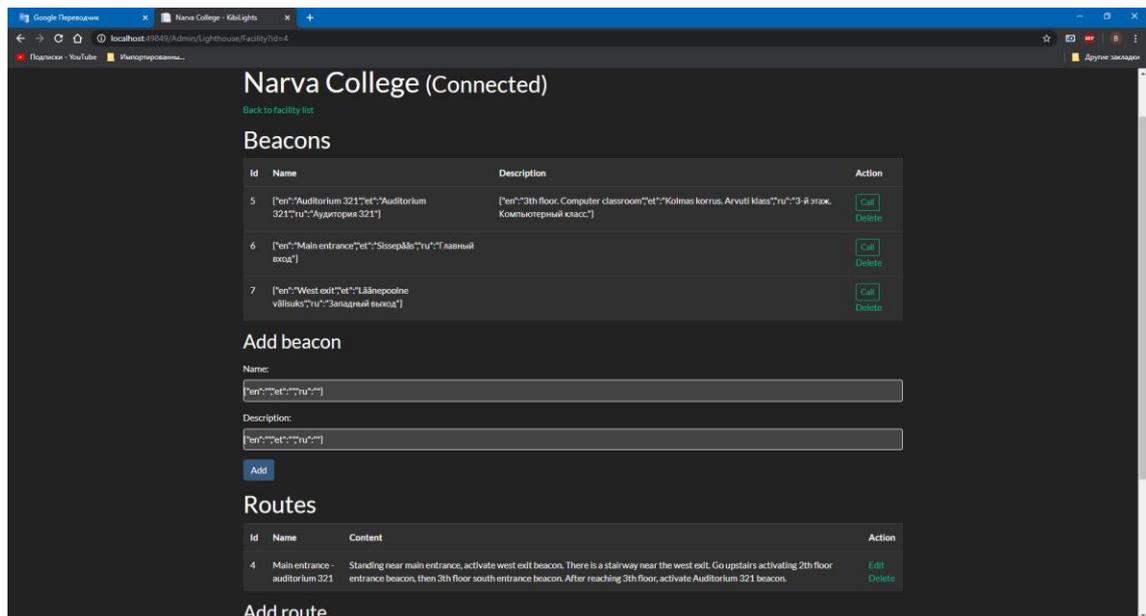


Figure 10. Beacons control page.

There is the information about the beacons installed in the building and about routes, related to this particular facility. Under each table there is a form for adding a new beacon or a new route. Besides an ability to delete or create a beacon, there also is “Call” button near each beacon. This button allows activating a particular beacon and forcing it to make the sound. It can be used for testing purposes during installation of the system in the building.

Each name or description of a beacon or of a route can be entered in several languages. In order to do this, administrator should use JSON format. For example:

```
{“en”:"Dining room”,“et”:"Söökla”,“ru”:"Столовая”}
```

There is an appropriate template in each field. After that, the mobile application will parse JSON strings, and will show the name or description in a language according user’s settings. If the administrator does not want to use this feature, he can enter names and descriptions as a usual text, and in the mobile application it will be displayed exactly as it was entered.

After creating records about all beacons installed in the building, administrator can add some routes with information about how to get from one to another place in the building. It could be some key areas, for example how to get from main entrance to the administrative part, from the main entrance to the elevator, or how to get from one floor to another, if the building has a tricky structure. After adding a route, by clicking on its

name the administrator will go to route edit page, where he can add beacons related to this route in order to provide the user of the system with navigation from beacon to beacon.

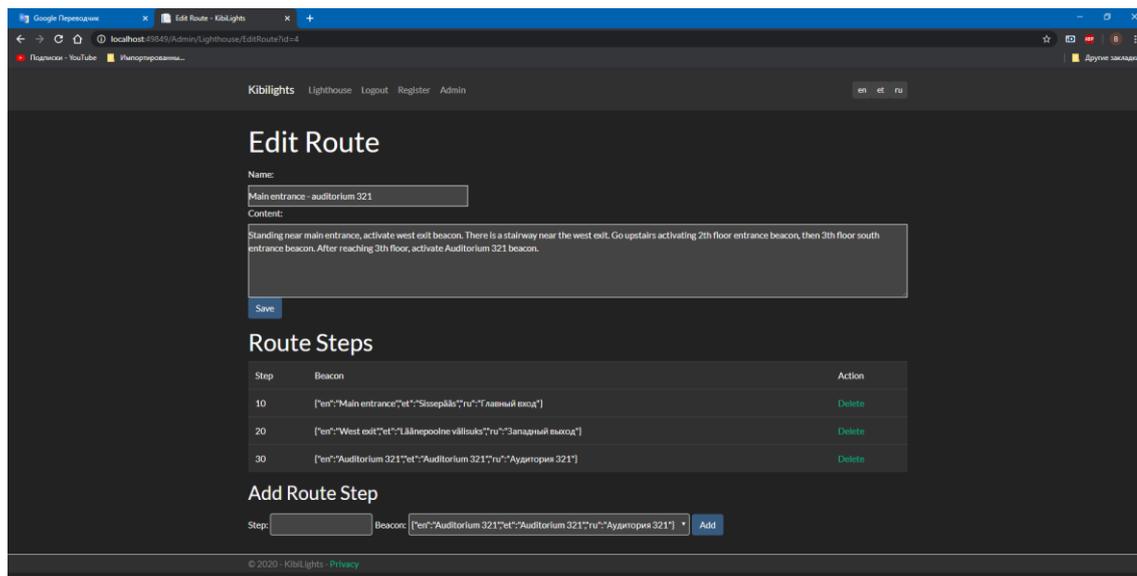


Figure 11. Route edit page.

So, the beacons management page in the admin section of the website is the key place, where administrator provides all information about beacons and routes, which then will be used in the user's mobile application.

3. 2. Spark

Spark is the beacons control software. It is a simple cross-platform console application, and it controls all beacons in the building, receiving signals from the server API and activating the appropriate beacon.

As we mentioned in the previous chapter, in this particular realization the system uses local wi-fi network in its work. So, a building must have a wi-fi network to be able equipped with the system. Nowadays in most of the cases this is not a problem.

Because of Spark is a cross-platform application, it can be installed on any machine. It could be a single board computer, computer with a Linux operation system, MacOS or Windows. The main requirement is that the machine must be part of a local network, and be constantly connected to the internet. The most suitable options to use are a single board computer, like Raspberry Pi, or a server machine, if an organization has something like this and it does not break the security policy.

Spark itself is a simple console application without graphical interface. It displays information using usual text, and controlled with text commands, which user enters manually.

A screenshot of a Windows command prompt window. The title bar shows the file path: D:\cs\kibilights\spark\bin\Release\netcoreapp3.1\Spark.exe. The window content shows the following text:

```
Spark v 1
Connected to hub.
test
Beacon id:
6
Calling 192.168.0.102
```

Figure 12. Spark command line interface.

When Spark runs for the first time, it creates a file with configuration, which must be edited. A configuration file is a text in JSON format. A file contains next information:

- Name of the server where the API is located. In this case it is kibilights.com
- Name and password of the special user, which was created in the website’s admin section, and which is needed for Spark could be authorized by the system’s API.
- Information about IDs of beacons registered in the system for this particular facility and corresponding to them IP addresses of beacons physically installed in the building. We will remind, that the current realization of the system uses static IP addresses for identification of each beacon in the local network.

Currently administrator of the system has to edit this file manually. In the future this process will be automated.

When the configuration file is edited, Spark is ready to work. Now, thanks to Microsoft SignalR technology, it will connect to the server API, and will wait signals from the server in order to activate a particular beacon using its ID received from the API and the corresponding static IP address. Spark generates a HTTP request for the local network, like “192.168.0.101/call”, and an appropriate beacon makes the sound, and sends back an “ok” message.

If Spark loses connection to the server API for some reason, it is trying to reconnect for several seconds, until attempt will be successful. Also, it is very useful to register Spark like a service in the operation system. It allows Spark to restart, if it will crash for some reason, or if the operation system will be rebooted.

Now, when all information about beacons filled in on the website, Spark is configured and works, and beacons are physically installed in the building, the system is ready to be used by blind or visually impaired people with the mobile application.

3.3. Lighthouse

We named the mobile application Lighthouse. It can be installed from kibilights.com. As it described in previous chapter, it is a PWA application. So to install it, user should visit the website of the project, click on the “Lighthouse” link, and then find a click button “Save on my home screen” in his browser. Saving to a home screen procedure can be a little bit different from browser to browser, and from device to device. PWA technology is quite new, so it is recommended for the user to have reasonably modern device with the latest version of the operation system.

The same as the website, the mobile application uses a dark schema on its graphical interface. GUI supports three languages (English, Estonian and Russian) and is totally readable with screen readers as well, it is VoiceOver on IOS and TalkBack on Android.

After installation, the icon of the application will appear on a user’s home screen, and user can run it. If the application starts for first time, the user will see the login form, where he should enter his email and password he used when he registered on the website.

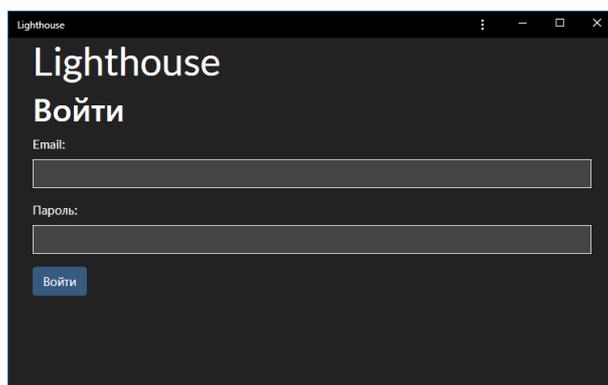


Figure 13. Lighthouse login page in Russian language.

After user clicks “Login” button, the application requests from the server an access token. If email and corresponding password are present in the database, the server sends back generated access token, which will be used in the requests to server API. The For interacting with the server HTTPS protocol is used, so all messages between the mobile application and server API are encrypted. All user’s settings and favorites’ lists are stored locally on the user’s device, and the system does not collect and does not store any

personal information, excluding email and encrypted version of password needed for registration.

After successful login, and again, if the application starts for the first time, user goes to the settings page (in a future, the settings page will be available at any time as well). Here the user should select a country, city and language of GUI, if automatically detected language does not satisfy him for some reason. Country and city settings affect which facilities user will see in facilities list. In the future city and country will be detected automatically, but in current implementation this information must be assigned manually.

When everything is done, the user clicks “Save” button.

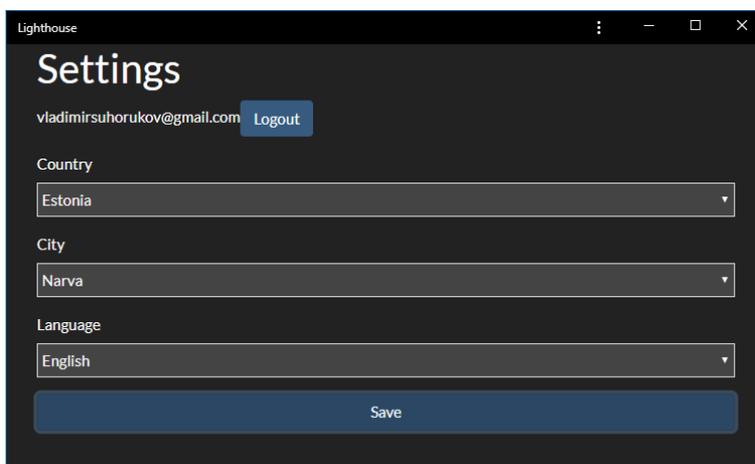


Figure 14. Lighthouse settings page.

And, finally, the user goes to the main page of the application. This page will appear directly after application starts when user runs it the next time.

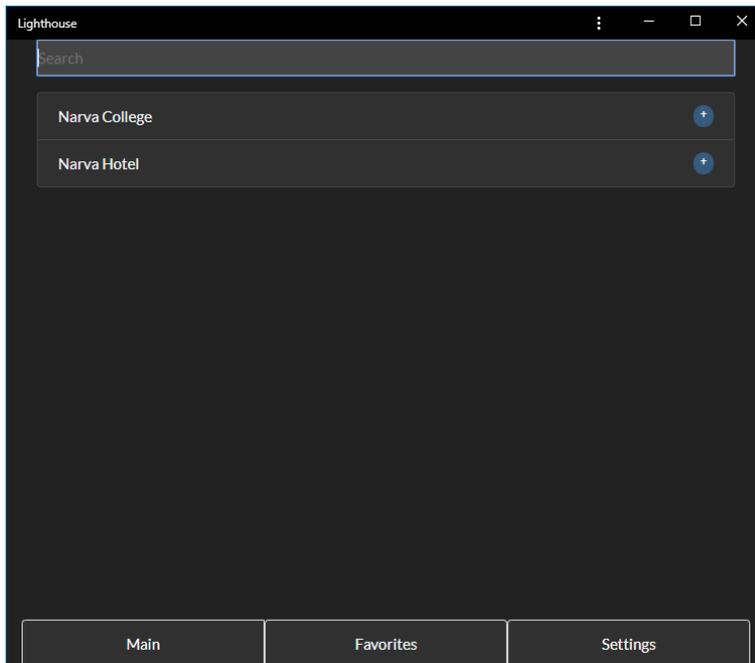


Figure 15. Lighthouse main page.

Here user can see a list of facilities in his city where the beacons navigation system is available. On the top of the page there is the search field in case the facilities list is too long. On the bottom of the page there are buttons for switching between the main page, the favorites page and the settings page.

In the facilities list the user can see names of available facilities and buttons for adding a facility to the favorites list. In GUI this button is represented by a plus sign to save a place on a screen, but it has invisible text label for screen readers saying “Add to favorites”.

Now, to see beacons installed in a particular facility, the user needs just tap on the desired facility name.

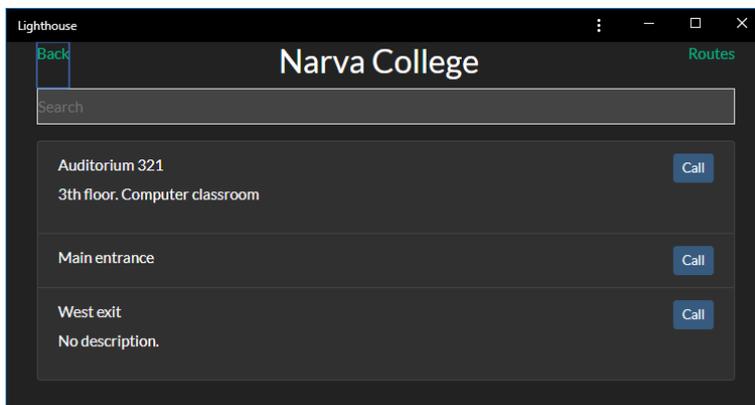


Figure 16. Lighthouse beacons list.

This page is similar to the previous one. It has a list of beacons installed in currently selected building, search field, “Back” link and a link for switching between beacons and routes lists.

In the beginning, the beacons list contains only the names of beacons. If the user taps on any name, detailed information about this beacon will appear. If the administrator of the facility filled information using JSON format, description of the beacon will be displayed in the language which was selected on the settings page. If detailed information is missing, an appropriate message will be displayed.

And finally, if user is now somewhere near the beacon in the building, he can tap “Call” button to force it make the sound and use it as an orienteer.

Using a “Routes” link on the right from a current facility name user can go to routes list available in this facility. The link will change its name to “Beacons”, and routes list will appear.

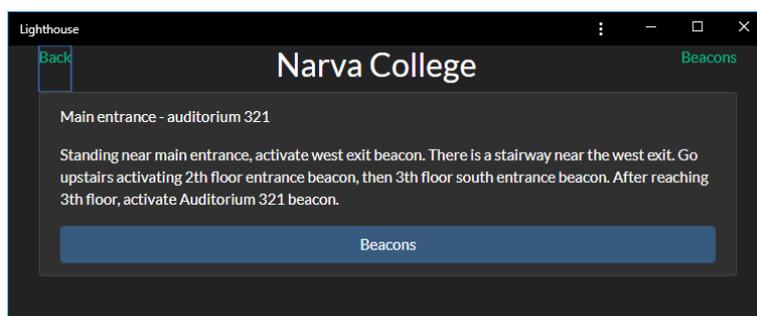


Figure 17. Lighthouse routes list.

Similar to the beacons list, there are routes names at the beginning. Tapping on a route name will display a detailed information about a particular route. Each route has “Beacons” button below. If the user taps this button, a list of beacons connected to this route will appear, and beacons’ order will be correspond to the route’s logic. Now, tapping appropriate “Call” buttons, hearing the sound of beacons and moving from one to other, the user can use the from beacon to beacon navigation feature.

3. 4. The link to the source code

The source code of the project can be downloaded from the GitHub: <https://github.com/Shyopinggo/kibilight>

CONCLUSION

The first chapter of this document provides general information about the topic of the work and describes the problem. Blind and visually impaired people face many problems in their life, and ability to navigate inside buildings is one of them. At the same time, modern technologies can solve many problems, and there is a strong evidence that this problem can be solved, too, so we proceeded in the first chapter with the research of the available technological solutions and tools, which already exist to help blind and visually impaired people to navigate inside buildings.

Research showed that available solutions have various disadvantages, and they cannot fully substitute visual orienteers for blind people. Based upon these disadvantages, we developed a set of requirements for a new system, which tries to get a little closer to the goal.

The second and the third chapter of this document describe the design and implementation of this new system. Here is a list of the requirements we consider fully implemented in the current version of the system:

- The system must have sound beacons, which can be activated by a user and which are silent when there is no need for them.
- The system and sound beacons must be designed in a way that there is a possibility to use a lot of beacons to cover most of possible points of interest inside a building.
- The system must provide a user with information about each beacon: what does it indicate and some additional useful details.
- The system must be able to lead a user to key areas of a building from beacon to beacon.
- A blind or visually impaired person does not have to buy any additional devices to use the system, except having a smartphone.
- All user interfaces of the system (website, mobile application, software for system administrator) must be fully accessible and convenient to use with a screen reading software.

In the future the author will try to find ways to complete the hardware part of the system, too. Currently the author does not have abilities to efficiently develop and test parameters of a physical device, like its energy consumption or type of the sound it makes. Perhaps a collaboration with an experienced engineer would help him to do it.

These are the steps which yet needs to be done in the hardware part of the project:

- The system and beacons must be cheap and easy to install.
- A beacon's sound must not be annoying for other people and should not sound like an alarm signal.

Currently the software part of the system is designed in a way to make beacons the simplest part of it, so that different beacon's hardware's implementations would have a minimum effect on the system structure in general.

As the goal of this thesis was to develop a software part of the project, the author can conclude that this goal was fully achieved.

SUMMARY IN ESTONIAN

Tänapäeval on maailmas hinnanguliselt 124 miljonit pimedat ja nägemispuudega inimest. Iga päev puutuvad need inimesed kokku raskustega sellistes esmapilgul tavalistes olukordades, kus terve inimene isegi ei kujuta ette probleemi olemasolu. Üks sellistest olukordadest on suutlikkus iseseisvalt orienteeruda tundmatutes hoonetes. Ja seal, kus meditsiin hätta jääb, tulevad appi tänapäevased tehnoloogiad.

Selle töö eesmärk on arendada välja tarkvaraline osa süsteemist, mis võimaldaks pimedatel ja raske nägemispuudega inimestel iseseisvalt liigelda hoonetes, kasutamata selleks visuaalseid orientiire, mis nende olukorras ei ole võimalik.

Autor on veendunud, et sellistest süsteemidest võivad olla huvitatud riigi- ja muud asutused, mis tahavad parandada ligipääsetavust oma avalikele hoonetele kõikide sotsiaalsete rühmade, sealhulgas puudega inimeste jaoks. Näiteks on ratastoolis inimeste jaoks paigaldatavad kaldteed muutunud uute hoonete projekteerimisel juba standardiks. Seepärast on kujunenud arvamus, et nägemispuudega inimeste liikumist lihtsustavaid süsteeme hakatakse tulevikus rohkem kasutama sellistes avalikes hoonetes nagu koolid, ülikoolid, riigiasutused, rongi- ja lennujaamad jne.

Diplomitöö esimeses peatükis kirjeldatakse probleemi, esitatakse olemasolevad lahendused ja kehtestatakse nõuded uuele süsteemile, arvestades olemasolevate süsteemide puudusi. Esmalt antakse ülevaade raskustest, millega nägemispuudega inimesed igapäevaelus kokku puutuvad ja tuuakse mõned näited selle kohta, kuidas saavad tehnoloogilised lahendused aidata neist raskustest üle saada. Seejärel tuleb käsitlemisele diplomitöö konkreetne teema.

Probleem seisneb selles, et hoones, eriti tundmatus hoones liigeldes ei saa nägemispuudega inimesed juhendada ülejäänutele kättesaadavatest visuaalsetest orientiiridest – alates sellest, et pimedad ei näe ümbritsevaid esemeid, ja lõpetades orienteerumist lihtsustavate viitade, ruumide numbrite, asendiplaanide ja muuga. Probleemi üks lahendus oleks asendada visuaalsed orientiirid heliga. Probleemist tulenevalt on edasi uuritud turul pakutavaid süsteeme, milles kasutatakse vaegnägijate jaoks helimajakaid. Nende põhjal luuakse selle töö raames arendatava süsteemi jaoks uued nõuded.

Diplomitöö teises peatükis käsitletakse tulevase toote projekteerimise ja arendamise küsimusi. Kirjeldatakse süsteemi põhikomponente, iga osa funktsionaalseid ja mittefunktsionaalseid nõudeid, võimalikke kasutusolukordi ja andmevoogusid ning valitakse süsteemi realiseerimiseks sobivad tehnoloogiad.

Süsteemi keskmes on helimajakad, mida saab paigaldada hoone eri osadesse: koridoride sissepääsude kohale, liftiuste kohale, auditooriumite uste kohale, kui tegemist on näiteks ülikooli õppehoonega, jne. Paigaldatud majakad ei too kuuldavale ühtegi heli ja neid saab aktiveerida ainult vaegnägija käsul, kes kasutab selleks mobiiltelefoni rakendust. Pärast majaka aktiveerimist saab vaegnägija orienteeruda visuaalse tähise asemel majaka heli järgi. Süsteemi teiseks eripäraks on teekonnad majakate vahel, mis kajastuvad rakenduses ja mida saab kasutada hoone asendiplaani asemel, et leida tee ühest hoone osast teise.

Diplomitöö kolmas peatükk on täielikult pühendatud süsteemi praeguseks valmis tehtud osade kirjeldamisele. Siin tuuakse välja süsteemi töö eripärad ning esitatakse liideste ja koostoime olukordade näited.

Süsteemi peamisteks eelisteks olid väike maksumus, hoonesse paigaldamise lihtsus ja tõsiasi, et puudus vajadus lisaseadmete järele, mida nägemispuudega inimene oleks pidanud juurde ostma. Vaja on vaid mobiilirakendust. Lisaks võimaldab süsteemi disain paindlikult läheneda majaka riistvara väljatöötamisele, sest kõik vajalikud andmed ja loogika on kantud serverirakendusse ja majakate juhtimiseks kasutatavasse platvormist sõltumatu rakendusse.

Eeltoodut arvestades on süsteem valmis kasutuskohas katsetamiseks ja riistvara täpsemaks viimistlemiseks. Kuna diplomitöö eesmärgiks oli just projekti tarkvaralise osa väljaarendamine, siis autori arvates on see eesmärk täielikult täidetud.

REFERENCES

Blindness and vision impairment. World Health Organization. Available at <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>, accessed April 20, 2020.

Lock, Andrew 2018. ASP.NET Core in Action. MANNING. Shelter Island.

Metzgar, Dustin 2018. .NET Core in Action. MANNING. Shelter Island.

Parus navigation system. Techno-park. Available at <http://techno-park.info/taxonomy/term/58>, accessed April 20, 2020.

Screen reader. MDN Web Docs. Available at https://developer.mozilla.org/en-US/docs/Glossary/Screen_reader, accessed April 21, 2020.

Silent sound beacon. Mobilis. Available at <https://mobilis.prom.ua/p902185974-besshumnyj-zvukovoj-mayak.html>, accessed April 20, 2020.

Sound beacon Hällö. Silmalaegas. Available at <https://silmalaegas.laegas.ee/kataloog/toode/28>, accessed April 20, 2020.

Sound beacon Landsort. Silmalaegas. Available at <https://silmalaegas.laegas.ee/kataloog/toode/26>, accessed April 20, 2020.

Sound beacon with remote control. Silmalaegas. Available at <https://silmalaegas.laegas.ee/kataloog/toode/27>, accessed April 20, 2020.

Speaking city. Spetstekhnopribor. Available at <https://www.speakingcity.org/en/>, accessed April 20, 2020.

White cane. Science Direct. Available at <https://www.sciencedirect.com/topics/nursing-and-health-professions/white-cane>, accessed April 20, 2020.

Zeil, Steven 2017. Integrated Development Environments. Old Dominion University. Available at <https://www.cs.odu.edu/~zeil/cs350/f17/Public/IDEs/index.html>, accessed April 20, 2020.

NON-EXCLUSIVE LICENSE

I, Vladimir Suhhorukov 03.02.1986,

1. herewith grant the University of Tartu a free permit (non-exclusive license) to:
 - 1.1.reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2.make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright, Development Of Sound Beacon Based Interior Navigation System For The Blind People, supervised by A. Säask,
2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Narva, 21.05.2020