EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPARTMENT OF CARTOGRAPHY AND GEOINFORMATICS

Mobile application for indoor wayfinding

Kalamkas Yessimkhanova

student of Cartography MSc

Dr. Gede Mátyás

Associate professor, PhD

ELTE Department of Cartography and Geoinformatics

Budapest, 2019

Contents

Introduction

The human being is set up in the way that it always needs to know where he is and requires sense of direction. Especially, when it comes to huge buildings, such as university campuses, shopping malls, hospitals, airports, business offices it is more and more difficult to orientate and navigate yourself. Nowadays, in the technological breakthrough century, GPS technology allows easily and precisely to navigate in space. However, GPS works only outdoor due to inability of signals penetration through buildings and other obstacles. For this reason, indoor navigation became vital and valid way of positioning inside buildings.

Considering the importance of indoor navigation, the purpose of the thesis is to develop mobile application of Northern Building of Eotvos Lorand University in Lágymányos campus. Inasmuch as ELTE is one of the biggest universities in Hungary, it has several buildings. As an object of indoor navigation I have chosen Northern Block due to my studies during these years were holds here specifically. Moreover, Faculty of Science and Faculty of Social Sciences are also based in this building. So, this mobile application can help both Hungarian and international students, freshmen particularly, to find classrooms effortlessly and fast.

There are many methods of developing indoor navigation system. Since there is no common standard for indoor positioning system it is enabled to build up mobile application for indoor wayfinding in arbitrary way.

My work based on previous research of Eszényi Krisztián, he successfully developed web application of Northern building of ELTE.

The whole working process was supervised by Dr. Gede Mátyás.

Theoretical part

1. What is Indoor Positioning System?

Indoor Positioning System (IPS) is a system applied to determine position of objects inside buildings and constructions where satellite navigational system is unavailable. IPS has several titles, such as indoor navigation, indoor wayfinding, indoor positioning, however, the concept of IPS remains the same.

Satellite positioning does not operate in covered areas, thus, it has led to the formation of brand new technologies to make indoor positioning possible. (Kaluža et al., 2017)

Indoor positioning is more complex than the outdoor positioning as the implementation of technologies used indoors may require some additional infrastructure. (Kaluža et al., 2017)

2. Methods of Indoor Positioning System

Since GPS is unreliable in interior spaces due to inability of signals to penetrate through obstacles an IPS must use other positioning methods. So, there is a classification of IPS which is revealed in the Figure 1 below.

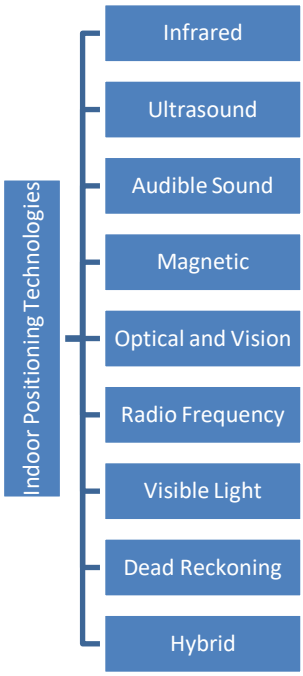


Figure 1. Indoor Positioning Technologies (Sakpere et al., 2017)

In this chapter only most popular technologies are described, which are widely used nowadays. There is a great deal of companies, which offer their services in indoor navigation installation. So, mainly they use Bluetooth Low Energy, Wi-Fi and Ultra-wideband technologies and it is explained by their considerable advantages.

Bluetooth technology is a cordless connection standard for exchanging data at a close range.

The primary features of Bluetooth technology are low-cost Bluetooth appliances, low energy consumption, minor range, reliability and worldwide use. Nowadays, almost all mobile devices are manufactured with a provision for standard Bluetooth and for the Bluetooth Low Energy (BLE), which is installed Bluetooth microcontroller with dual operating mode. Mobile operating systems that support the BLE at the moment are: Android 4.3 and higher; iOS 5 and higher; Windows Phone 8.1; Blackberry 10. (Kaluža et al., 2017)

BLE technology is defined by a quite small size of transmitters, cheap price, and low power use with the prospect of some years of work on a small power source (AAA batteries). BLE technology is compatible with mobile devices, tablets and computers. It uses the beacons (transmitters) and location devices. Device detects a signal from a transmitter and approximately computes the remoteness to it and so calculates the location of a device. (Kaluža et al., 2017)

So, basically, BLE technology uses received signal strength indication (RSSI) assessment to define beacon's position. Moreover, it is possible to send notification to a user through BLE transmitters. It is favorable for shopping malls, for instance, thus, the shops are able to promote their services or sales. Operating range of BLE beacons is up to 70 meters and the high accuracy they can reach is 1 meter. BLE beacons are presented from large numbers of providers in different shapes and sizes. (White Paper)

BLE is one of the most popular technologies among the others because of its easy installation and low price.

Wi-Fi Positioning System is a good alternative to GPS where satellite signals are not available. Wi-Fi positioning system is a system that tracks and defines the position of an object within the limits of coverage area using the WLAN developed infrastructure. Wi-Fi uses radio waves to establish connection to the existing network.

Comparing to BLE technology WP system requires an external power source, greater installation costs and more expensive hardware. It is not necessary to be connected to a Wi-Fi network; however, Wi-Fi connection has to be enabled. Emitted signals are stronger and they cover the range is up to 150 meters, but precision is relatively low and it constitute 5-15 meters. (Kaluža et al., 2017)

Wi-Fi devices, access points and server these are the components of WPS. WPS can be executed by propagation and fingerprinting techniques.

Certainly that WPS as any system has advantages and disadvantages. For instance, convenient installation due to possibility of using already existing WI-Fi infrastructure might be positive superiority. Significant disadvantage, except its low accuracy, is impossibility of positioning for iOS devices.

Ultra-wideband technology

UWB is a short-range high-speed radio technology for wireless communication. (Sakpere et al., 2017)

Ultra-wideband sensors have fixed position in the network infrastructure and they use running time of the light to compute the distance of an object. The method measures the passing time of light between an object and several receivers (Time Difference of Arrival). (White Paper, 2018)

UWB signals can pass through both walls and other barriers. Moreover, UWB technology can work with other radio signals with no failures. So, UWB method of indoor navigation might be impeccable solution for places with radio-frequency interference such as, hospital or industrial areas. Besides, applying of more UWB transmitters and their intelligent arrangement may lead to a wider coverage area, real-time tracking, superior positioning accuracy and reducing of the effect of signal disturbances. (Quinn, 2018)

Considerable accuracy (up to 30 centimeters) and nearly low interference make this method favorable for the areas where required high precision (mainly industrial zones). Juxtaposing with BLE beacons, UWB sensors have short lifetime. Apart from this, installation and sensors cost significantly expensive. Therefore, before selecting UWB method for indoor navigation there should be thinking over the plan of action, budget and expecting results.

Visible Light positioning system is wireless technology which uses special LED and fluorescent lamps as a contact channel. The LED sends flickering light so phone camera can detect it and the signal is converted into an encrypted form and redirected to the server. VL technology has such advantages like longer lifespan of the lamps, does not depend on batteries and accuracy of location quite high (less than 1 meter). However, the great weakness is that there is should be visual contact of phone camera with the lamps.

Otherwise, the device will not receive the information back if visual contact is not reached. (Kaluža et al., 2017)

Marker-based method provides an inexpensive way finding within a building that relies solely on camera phones. The working principle of this method that mobile phone camera obtains visual data from markers to calculate its position relative to the marker. (Mulloni et al., 2019)

There are three attributes of this system – mobile device with camera, markers and server. Markers constitute barcodes, QR codes and fiduciary markers. The server is used for tracking objects and storing data like building plan, some additional information about construction.

This system decreases cost noticeably in comparison with other positioning systems. However, the system has poor accuracy, interference from various effects such as bright light and motion blur. In certain cases, privacy aspect interests may be an issue since server stores position information for tracking and navigation intentions. (Sakpere et al., 2017)

Hybrid method

Basically, various systems can be used in a hybrid operating mode. Hybrid modification of indoor navigation technologies has a tendency for popularity since it is a combination of two or more methods of IPS which allows improving accuracy, robustness and performance significantly. Because of complexity of hybrid IPS it may affect to its cost and time consuming process while installation.

3. Application of Indoor Positioning System

Present time cutting-edge technology allows constructing a building of any complexity. Therefore, there is a need to implement appropriate indoor navigation system for specific use cases. As a case it might be shopping malls, supermarkets, business offices, hotels, airports, railway stations, hospitals, exhibitions, fairs, museums, university campuses, amusement parks, parking and industry zones. Using indoor navigation applications enables customers to orientate themselves, finding their way and reaching the required terminal point with no wasting time, physical effort and nervousness.

4. Short review of IPS market

In the digitalization time there is supply and demand of indoor positioning systems. Moreover, this area is not studied fully and requires development due to the continuous growth of technologies. Therefore, many companies have started their activities as start-up projects and became large and successful companies. If you start to search companies dealing with IPS in Google searching system you can find a large number of websites and contacts. Some of them produce sensors and beacons, others do mapping for indoor navigation system and still others accomplish installation and implementation of IPS. As in any sphere of industries and business, there are progressive, large and international companies. So, i would like to mention such companies as Infsoft, IndoorAtlas, Senion, Indoo.rs, Esri, Visioglobe and others.

Practical part

The aim of this master thesis is to develop a mobile application that facilitates indoor navigation using already existing database of the ELTE Northern block. The application determines the position of the queried room with the possibility of switching floors, zooming in and out. The project consists of two parts. The first part is manipulating of primary data and the second – coding. Data processing was done in QGIS open-source geographic information system desktop application and the development of the application was carried out in HTML page with JavaScript codes. The application works on Android operating system and for visualizing and running the application was used Android Studio integrated development environment. It is quite important to mention that the application does not show current location of the user and works offline. Due to impossibility of applying BLE beacons, UWB sensors or WiFi technology it was decided to develop application which runs in offline mode.

As a source data were used existing database which was compiled by Eszényi Krisztián. Eszényi Krisztián is a graduated master student who had successfully developed web application for indoor navigation of the Northern building of ELTE. His project was accomplished in QGIS software. All the necessary data were provided by my supervisor Dr. Gede Mátyás.

Analyses of baseline data

Before i started to manipulate the data it was very crucial to have an analyses of given data. I should mention that the attributive data is in Hungarian language, so I made some translation, for example, *ajtok* is doors, *epulet* - buildings, *termek* - halls(rooms), *utvonalak* - routes. Northern block of ELTE consists of underground floor, ground floor and 7 floors. Therefore, there are files in .qgs file format with the shapefiles of each floor with doors, rooms, routes and you can see them on the Figure 2.

0.qgd	10.04.2018 9:07	Файл "QGD"	0 КБ
0.qgs	10.04.2018 9:12	QGIS Project	50 КБ
0.qgs~	10.04.2018 9:12	Файл "QGS~"	50 КБ
ajtok_00.dbf	05.03.2018 15:07	Файл "DBF"	88 КБ
ajtok_00.prj	01.03.2018 12:19	Файл "PRJ"	1 КБ
ajtok_00.qpj	01.03.2018 12:19	Файл "QPJ"	1 КБ
ajtok_00.shp	05.03.2018 15:07	Файл "SHP"	11 КБ
ajtok_00.shx	05.03.2018 15:07	Файл "SHX"	3 КБ
epulet_00.cpg	19.11.2017 14:55	Файл "CPG"	1 КБ
epulet_00.dbf	19.11.2017 14:55	Файл "DBF"	2 КБ
epulet_00.prj	19.11.2017 14:55	Файл "PRJ"	1 КБ
epulet_00.qpj	19.11.2017 14:55	Файл "QPJ"	1 КБ
epulet_00.shp	19.11.2017 14:55	Файл "SHP"	2 КБ
epulet_00.shx	19.11.2017 14:55	Файл "SHX"	1 КБ
termek_00.cpg	19.11.2017 14:48	Файл "CPG"	1 КБ
termek_00.dbf	27.04.2018 12:25	Файл "DBF"	39 КБ
termek_00.prj	19.11.2017 14:48	Файл "PRJ"	1 КБ
termek_00.qpj	19.11.2017 14:48	Файл "QPJ"	1 КБ
termek_00.shp	27.04.2018 12:25	Файл "SHP"	22 КБ
termek_00.shx	27.04.2018 12:25	Файл "SHX"	2 КБ
utvonalak_00.dbf	10.04.2018 9:11	Файл "DBF"	5 КБ
utvonalak_00.prj	26.02.2018 17:44	Файл "PRJ"	1 КБ
utvonalak_00.qpj	26.02.2018 17:44	Файл "QPJ"	1 КБ
utvonalak_00.shp	10.04.2018 9:11	Файл "SHP"	35 КБ

Figure 2. An example of given files representation

Figure 3 reveals how the files look like if you open them in QGIS software.

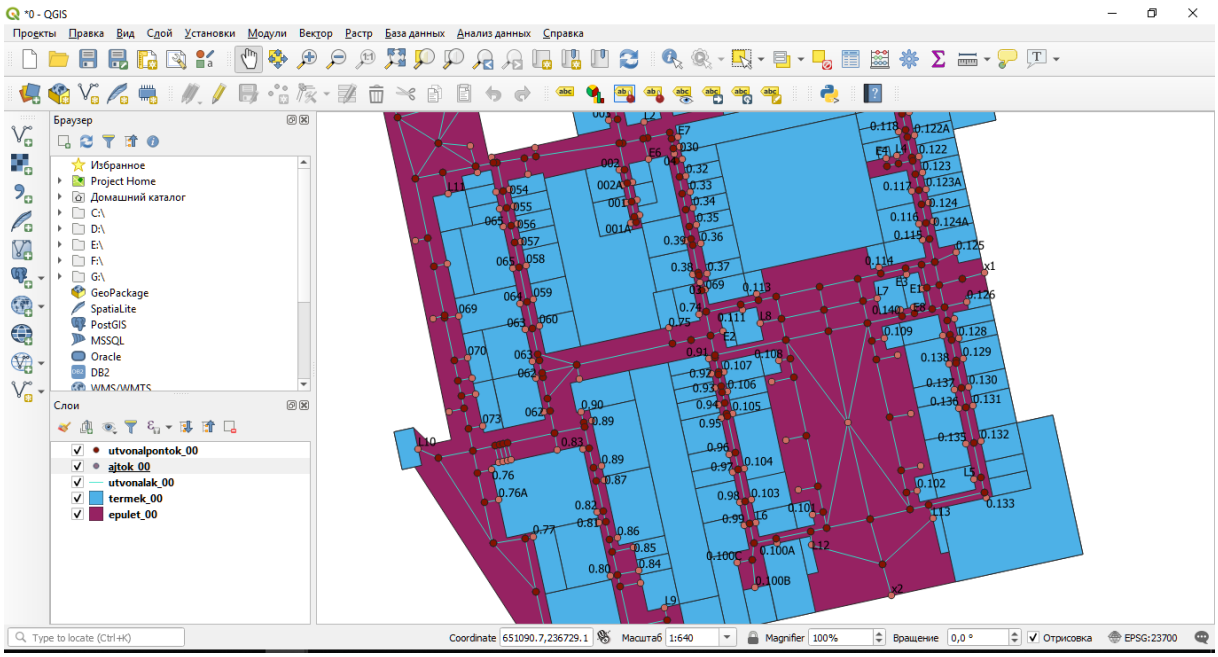
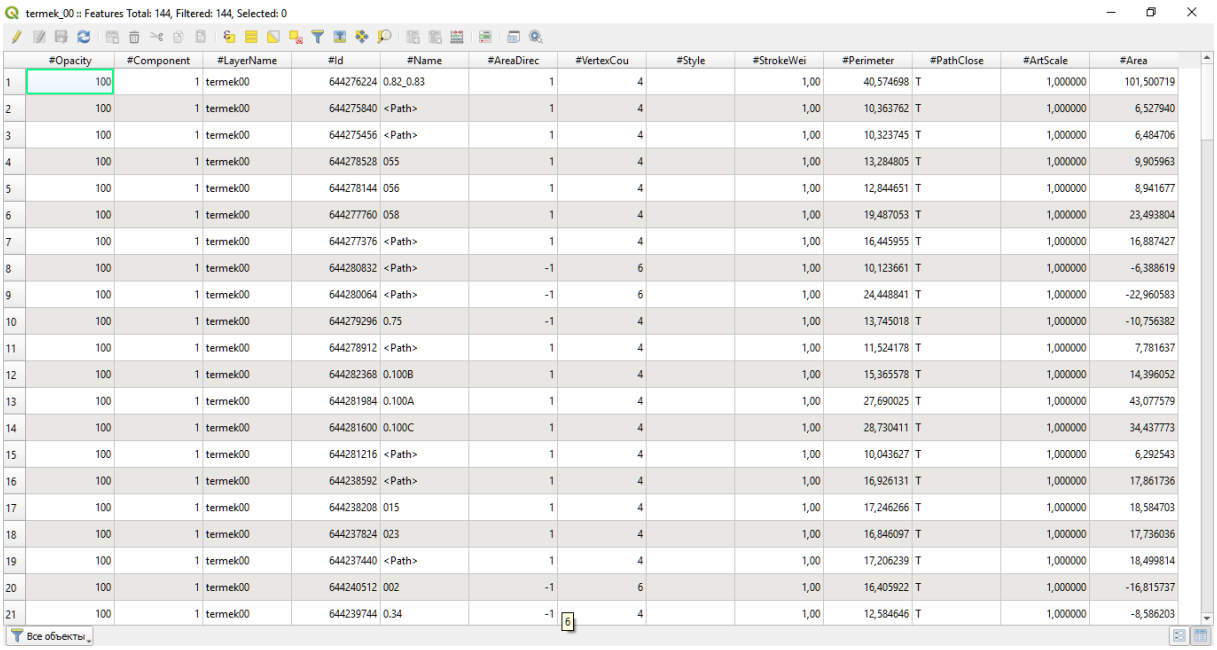


Figure 3. Primary data in QGIS software

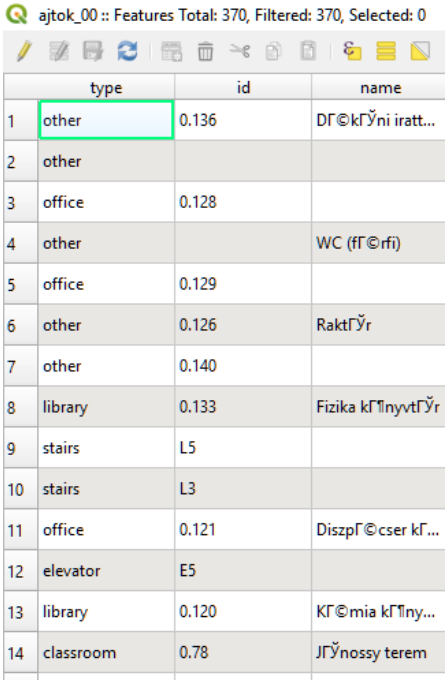
There are also attributive data for each layer in QGIS. For instance, as you can see on the Figure 4 the layer *termek_00*, which means rooms on the ground floor, has such attributive data as *#LayerName*, *#id*, *#Name*, *#Area* and others.



	#Opacity	#Component	#LayerName	#Id	#Name	#AreaDirec	#VertexCou	#Style	#StrokeWei	#Perimeter	#PathClose	#ArtScale	#Area
1	100	1	termek00	644276224	0.82_0.83	1	4		1,00	40,574698	T	1,000000	101,500719
2	100	1	termek00	644275840	<Path>	1	4		1,00	10,363762	T	1,000000	6,527940
3	100	1	termek00	644275456	<Path>	1	4		1,00	10,323745	T	1,000000	6,484706
4	100	1	termek00	644278528	055	1	4		1,00	13,284805	T	1,000000	9,905963
5	100	1	termek00	644278144	056	1	4		1,00	12,844651	T	1,000000	8,941677
6	100	1	termek00	644277760	058	1	4		1,00	19,487053	T	1,000000	23,493804
7	100	1	termek00	644277376	<Path>	1	4		1,00	16,445955	T	1,000000	16,887427
8	100	1	termek00	644280832	<Path>	-1	6		1,00	10,123661	T	1,000000	-6,388619
9	100	1	termek00	644280064	<Path>	-1	6		1,00	24,448841	T	1,000000	-22,960583
10	100	1	termek00	644279296	0.75	-1	4		1,00	13,745018	T	1,000000	-10,756382
11	100	1	termek00	644278912	<Path>	1	4		1,00	11,524178	T	1,000000	7,781637
12	100	1	termek00	644282368	0.100B	1	4		1,00	15,365578	T	1,000000	14,396052
13	100	1	termek00	644281984	0.100A	1	4		1,00	27,690025	T	1,000000	43,077579
14	100	1	termek00	644281600	0.100C	1	4		1,00	28,730411	T	1,000000	34,437773
15	100	1	termek00	644281216	<Path>	1	4		1,00	10,043627	T	1,000000	6,292543
16	100	1	termek00	644238592	<Path>	1	4		1,00	16,926131	T	1,000000	17,861736
17	100	1	termek00	644238208	015	1	4		1,00	17,246266	T	1,000000	18,584703
18	100	1	termek00	644237824	023	1	4		1,00	16,846097	T	1,000000	17,736036
19	100	1	termek00	644237440	<Path>	1	4		1,00	17,206239	T	1,000000	18,499814
20	100	1	termek00	644240512	002	-1	6		1,00	16,405922	T	1,000000	-16,815737
21	100	1	termek00	644239744	0.34	-1	4		1,00	12,584646	T	1,000000	-8,586203

Figure 4. Attributive data of rooms

It is important to have separate layer for doors. It is related to the reason that one room might have several doors. So, this layer (Figure 5) has *#type*, *#id* and *#name* attributes.



	type	id	name
1	other	0.136	DГ©kГЎni iratt...
2	other		
3	office	0.128	
4	other		WC (fГ©rfi)
5	office	0.129	
6	other	0.126	RaktГЎr
7	other	0.140	
8	library	0.133	Fizika kГ¶nyvtГЁr
9	stairs	L5	
10	stairs	L3	
11	office	0.121	DiszpГ©cser kГ...
12	elevator	E5	
13	library	0.120	KГ©mia kГ¶ny...
14	classroom	0.78	JГЎnossy terem

Figure 5. Attributes of the doors layer

Data processing

The first step of data manipulating was correcting and filtering of attributive data. The application shows the plan of each floor, rooms with numbers, stairs, elevators and toilets. Parameters of the data were selected on the basis of the information is projected to display. However, the query will be executed by door number. It explains by the reason that some rooms have two or more doors. Moreover, it is connected to the idea that students look for rooms by its unique number, which means door number. Some room numbers were not indicated in the given data, so I made walking research in the building and filled lack data.

The second step of data processing is exporting each layer (doors, rooms, floors) to GeoJSON file format. GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). (*The GeoJSON Specification (RFC 7946))*

While exporting the files I changed the coordinate system from EOV/ HD72 Hungarian projected coordinate system with datum of 1972 to the geographic coordinate system WGS84. Figure 6 illustrates the exporting process in QGIS application.

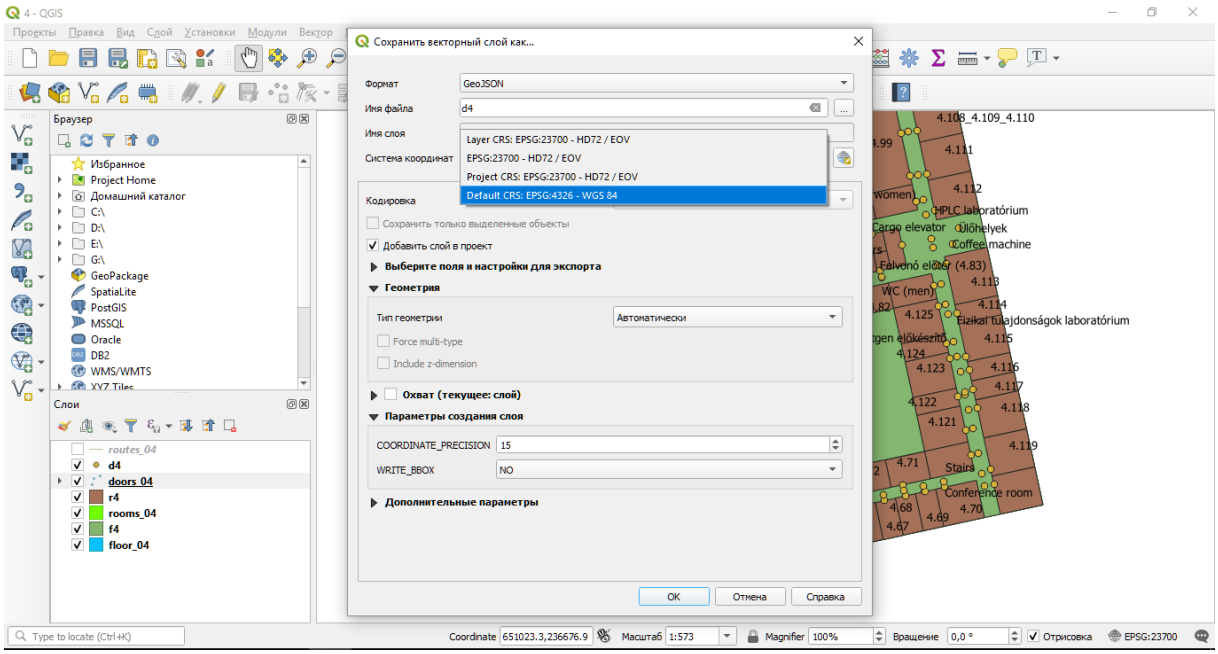


Figure 6. Exporting a file in QGIS desktop application

Exporting to GeoJSON format is required since GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984, and units of decimal degrees. (*The GeoJSON Specification (RFC 7946)*)

All exported files are stored in the “Data” folder with the floor number and demonstrated in the Table 1 below.

Table 1. Exported files

File names	Interpretation of file names
d0.geojson	doors at level 0 (groundfloor)
d1.geojson	doors at level 1
d2.geojson	doors at level 2
d3.geojson	doors at level 3
d4.geojson	doors at level 4
d5.geojson	doors at level 5
d6.geojson	doors at level 6
d7.geojson	doors at level 7
d11.geojson	doors at level -1
f0.geojson	floor 0 (groundfloor)
f1.geojson	floor 1
f2.geojson	floor 2
f3.geojson	floor 3
f4.geojson	floor 4
f5.geojson	floor 5
f6.geojson	floor 6
f7.geojson	floor 7
f11.geojson	floor -1
r0.geojson	rooms at level 0 (groundfloor)
r1.geojson	rooms at level 1
r2.geojson	rooms at level 2
r3.geojson	rooms at level 3
r4.geojson	rooms at level 4
r5.geojson	rooms at level 5
r6.geojson	rooms at level 6
r7.geojson	rooms at level 7
r11.geojson	rooms at level -1

GeoJSON supports following geometry types as points, linestrings, polygons, multipolygons. On the Figure 7 you can see the screenshot of the GeoJSON file for 4th

floor which shows doors as a point and rooms as a multipolygon geometry type with coordinates in decimal numbers.

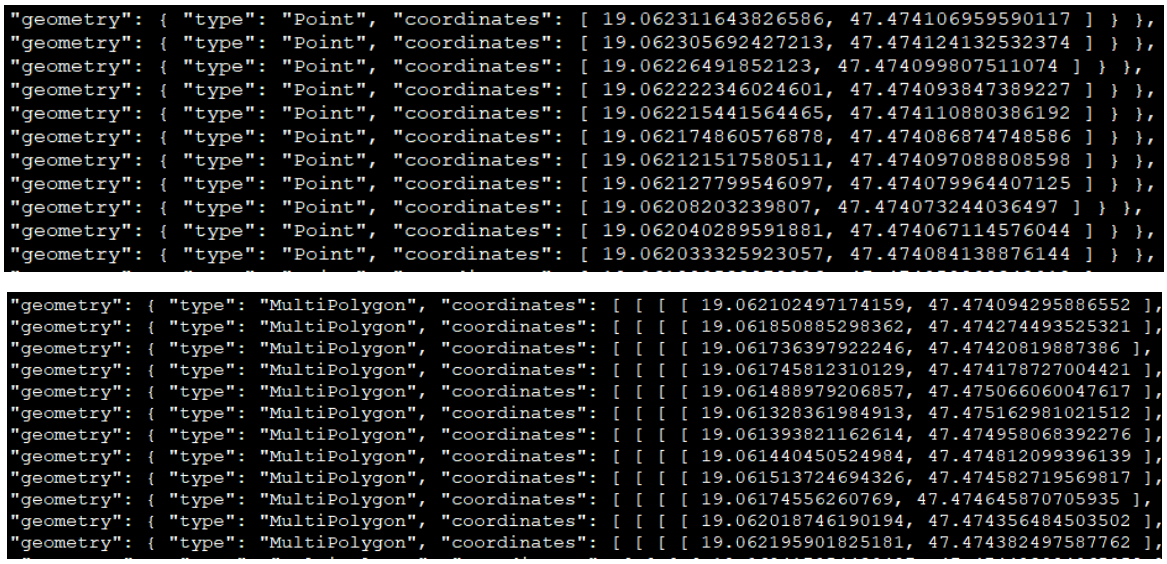


Figure 7. GeoJSON geometry type example

The third step of data handling is development of mobile application. The method of creating application was based on the source data, opportunities and my knowledge in coding. Due to the lack of my knowledge in programing languages such as Java, Python, CSharp, it was decided to create HTML page with JavaScript codes. Basically, there is a WebView function in Android Studio which allows to show a webpage inside an application. Apart from this, the function allows to run the application locally on the phone, which means you do not need to have an access to the internet. Application runs only on Android operating system.

The WebView class is an extension of Android's View class that allows you to display web pages as a part of your activity layout. It does not include any features of a fully developed web browser, such as navigation controls or an address bar. All that WebView does, by default, is show a web page. (*Website for Android Developers*)

The folder with all the data consists of following files:

- data (folder with GeoJSON files);
- img (folder with icons for search and clear functions in .svg file format);
- modules (folder with CSS and JavaScript modules);
- ol (folder with OpenLayer modules for displaying map data);
- index.html (initial HTML webpage);

- mobileview.html (HTML webpage created as an emulator of mobile device screen);
- nb.jpg (image of ELTE Northern building);
- north.html (HTML webpage with map object and all the layers).

In order to make clear all the steps of coding part I divided the explanation in to the three parts. So, they are:

a) Creating HTML webpage
b) JavaScript script
c) Running application in Android Studio

All the codes are available on the supplement CD; therefore I would like to mention that I am going to emphasize only on the key points of building the application and difficulties in the running the application.

I would like to mention the sources for coding part because it is hard to reference certain codes. All the codes were taken from different sources like tutorial websites, video lessons on YouTube platform and internet forums. However, mainly the codes were used from the following websites:

1. Personal webpage of Dr. Gede Mátyás with the lectures and tutorials for students. There are educational materials in "Education" section on the webpage http://mercator.elte.hu/~saman/

   Materials related to HTML and JavaScript basics are enclosed in headings "Open Source Web GIS" and "Scripting languages in webcartography".
2. Official website of OpenLayers web-browser https://openlayers.org/

Creating HTML webpage

Basically, the application consists of two webpages. The first webpage include the header, the link image and the image caption. The header is the name of the university building which is ELTE Lágymányos Campus, Northern Building and the address of it, Budapest, Pázmány Péter stny. 1/A, 1117. It is very important to indicate that this is Northern block due to there is also Southern in front of it.

The look of the application is shown on the Figure 8.

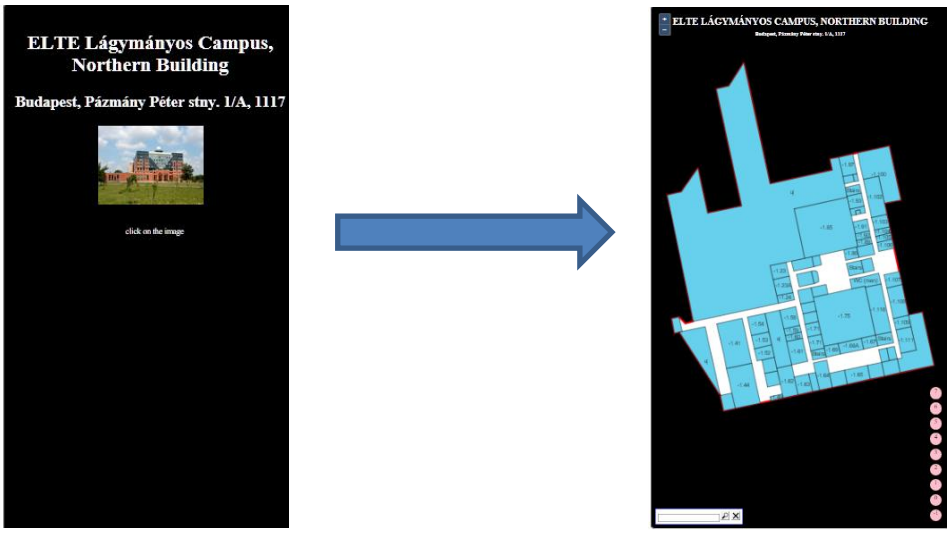


Figure 8. Look of the application

I put the linked image to the initial page, because of design purpose, it is easy to recognize the building and the link leads to the plan of second page, which is the indoor map indeed. Below the image there is the tag “click on the image” to guide a user. Moreover, I put timer for the image link for 5 seconds by JavaScript codes below. It means that if a user will not click on the link, then page is going to be redirected automatically in 5 seconds.

```
<script>
        setTimeout('location="north.html"',5000);
</script>
```

The header, indoor map, floor switcher buttons and search box are fitted on the second page.

Styling is one of the significant parts of coding. Perfect matching colors, font size, typeface of the letters altogether make an application well designed. Moreover, an application should be user-friendly; it means all the functions have to be comprehensive and quite clear and easy in use. I set up the font color as white on the black background colour. From my point of view, it is the best colour combination because of its high contrast. The map itself is light blue colored rooms on the white colored floors. Room numbers indicated in black and search box in color of white. Floor switchers have two colors, when the floor is set up it is highlighted in green, in other cases all the buttons are pink.

Since the aim of the thesis is to create easy application I did not put any additional data so that not to overload it with unnecessary functions.

JavaScript

JavaScript language allows embedding complicated functions into simple HTML webpages. For this reason, the combination of HTML pages with JavaScript codes is most common way of developing applications. Below there is a code how to include in HTML body JavaScript codes, so that the page will be able to display JavaScript embedded codes.

```
<html>
    <head>
        <title>MScCartography Thesis Project</title>
        <meta charset="utf-8" />
        <link rel="stylesheet" href="ol/ol.css" type="text/css">
        <script src="ol/ol.js" type="text/javascript"></script>
```

Moreover, it is possible to use OpenLayers in JavaScript which is appropriate for layered map. So, basically, the second page is a bunch of layers which work as a single mechanism.

OpenLayers is an open-source JavaScript library for displaying map data in web browsers as slippy maps. It provides an API (application programming interface) for building rich web-based geographic applications similar to Google Maps, for example. OpenLayers supports GeoRSS, KML (Keyhole Markup Language), Geography Markup Language (GML), GeoJSON and map data from any source using OGC-standards as Web Map Service (WMS) or Web Feature Service (WFS). (*OpenLayers*)

In order to have all the settings and styling working properly it is necessary to have modules for that in a folder. All the applied for the application modules and styling settings are shown on the Figure 9.

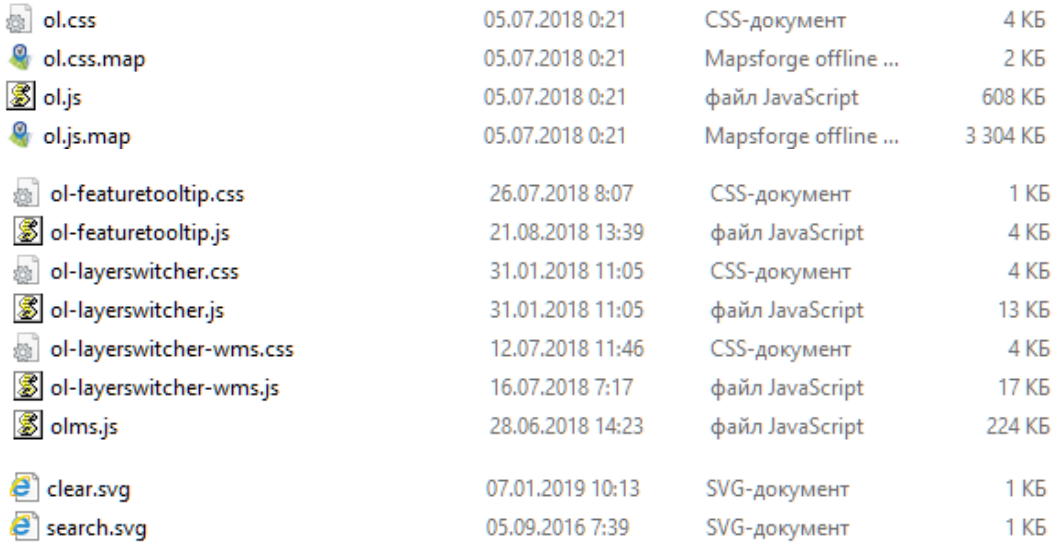


Figure 9. Modules for JavaScript

So the structure of the second page looks in the following way. All the layers are listed in the same order as it is in the script.

- Room style;
- Floor style;
- Door marker style;
- Search results;
- Doors at level 11;
- Floor 11;
- Rooms at level 11;
- Map object;
- Floors at all levels;
- Rooms at all levels;
- Doors at all levels;
- Fit view to building outline;
- Search for room;
- Clear results;
- Level switcher.

Layers for room style and floor style are very similar due to the logic of designing. Only differences are colors and fonts. Apart from this, the most important is the name of variable. Thus, the statement `var` is different for each layer as it is the name of it. Codes for room and floor styles are below.

```
// room style
            var rStyle=function(f,r) {
                return new ol.style.Style({
                    stroke: new ol.style.Stroke({ color: 'black', width: 1}),
                    fill: new ol.style.Fill({ color: [102,208,237] }),
                    text: new ol.style.Text({ font: '13px sans-serif', text: f.get('#Name') })
                })
            }
// floor style
            var fStyle=function(f,r) {
                return new ol.style.Style({
                    stroke: new ol.style.Stroke({ color: 'red', width: 4}),
                    fill: new ol.style.Fill({ color: [255,255,255] }),
                    text: new ol.style.Text({ text: f.get('#Name') })
                })
            }
```

Style “Door marker style” is mandatory in consequence to display search results. Codes below reveal that search results are going to be shown with the mark red circle with the

radius 7 pixels and width of 3 pixels and the number of the door will be indicated in black color with the font size 20 pixels in Sans-serif font face.

```
var dStyle=function(f,r) {
                return new ol.style.Style({
                    image: new ol.style.Circle({
                        stroke: new ol.style.Stroke({ color: 'red', width: 3}),
                        //fill: new ol.style.Fill({ color: [0,0,255,.2]}),
                        radius: 7
                    }),
                    text: new ol.style.Text({
                        text: f.get('id'),
                        textAlign: 'left',
                        offsetX: 8,
                        color: 'black',
                        font: '20px sans-serif',
                        backgroundFill: new ol.style.Fill({ color: [255,255,255,.5]}),
                        stroke: new ol.style.Stroke({ color: 'white', width: 3}),
                        padding: [0,2,0,2]
                    })
                })
            }
```

“Search results” layer is using `dStyle` as default style, what means that variable `sr` gets the attributes of the variable `dStyle`. Codes below show function “Search results”.

```
var sr=new ol.layer.Vector({
                source: new ol.source.Vector(),
                style: dStyle
            });
```

Layers “Doors at level 11”, “Floor 11” and “Rooms at level 11” are collaborated in one variable because it is possible to indicate them in one function by simply denoting `var d=[], f=[], r=[]`. It means that the content of the layer itself are going to be unfolded from brackets to separate layer. It is ought to point out that in all script of the project number "-1" is replaced by "11", it is related to the cause of peculiarity of coding. The symbol "-" leads to the crash of the operation sometimes or it is impossible to run the codes. At this step the main point is to note URL as the way GeoJSON files. As you can see from the codes below URL is `'data/d11.geojson'` for doors at level -1 which means that this layer shows the content of `'d11.geojson'` file. Apart from this, it is essential to claim in codes that the format of source is going to be GeoJSON.

```
var d=[], f=[], r=[];
          // doors at level 11
          d[-1]=new ol.layer.Vector({
        source: new ol.source.Vector({
                  format: new ol.format.GeoJSON(),
                  url: 'data/d11.geojson'
              }),
          style: null
          });
        // floor 11
          f[-1]=new ol.layer.Vector({
           source: new ol.source.Vector({
                  format: new ol.format.GeoJSON(),
                  url: 'data/f11.geojson'
              }),
        style: fStyle
          });
        // rooms at level 11
          r[-1]=new ol.layer.Vector({
          source: new ol.source.Vector({
                  format: new ol.format.GeoJSON(),
                  url: 'data/r11.geojson'
              }),
          style: rStyle
          });
```

The layer “Map object” is kind of a container for all layers, so that it is possible to display them in frame of the map borders. It is required to center the map object by indicating its coordinates. Zoom was set up by 10, thus, the map is shown in the middle of the phone screen and do not overlap other elements of the page. As an initial layer I selected underground floor -1, so when you run the application it is going to be shown the layer of -1 floor with all its attributes. Codes for the layer “Map object” are shown below.

```
var map=new ol.Map({
          target: 'map_div',
        layers: [d[-1],f[-1],r[-1]],
        view: new ol.View({
                  center: ol.proj.fromLonLat([19.062, 47.4745]),
                  zoom: 10
              })
          });
```

The string “`center: ol.proj.fromLonLat([19.062, 47.4745])`,” says to get coordinates from WGS 84 and the coordinates will be automatically converted into Web Mercator coordinate system. It happens because of the reason that in mostly web map applications Web Mercator coordinate system is used as a default.

Web Mercator, Google Web Mercator, Spherical Mercator, WGS 84 Web Mercator or WGS 84/Pseudo-Mercator is a benchmark in fact for Web mapping applications. It is used by literally all large online map providers such as Google Maps, OpenStreetMap, Esri and many others. Its official EPSG identifier is EPSG: 3857. (*Web Mercator projection*)

Layers “Floors at all levels”, “Rooms at all levels” and “Doors at all levels” are similar in their way of representation. Codes for these layers are shown below. The variables are declared by dependence `var i=0;i<8;i++` which means the function “`i`” will take the value equal to zero and less than 8. The contrast of layers is URL, which is various for each floor, rooms and doors at all levels. Due to displaying as an initial underground layers it is mandatory to insert the function “`map.addLayer(f[i])`” (case for floors). This code declares to add floors to the map object.

```
for (var i=0;i<8;i++) {
// floor at all levels
        f[i]=new ol.layer.Vector({
        source: new ol.source.Vector({
        format: new ol.format.GeoJSON(),
                url: 'data/f'+i+'.geojson'
        }),
                visible: false,
                style: fStyle
        });
map.addLayer(f[i]);
// rooms at all levels
        r[i]=new ol.layer.Vector({
        source: new ol.source.Vector({
        format: new ol.format.GeoJSON(),
                url: 'data/r'+i+'.geojson'
        }),
                visible: false,
                style: rStyle
        });
map.addLayer(r[i]);
// doors at all levels
        d[i]=new ol.layer.Vector({
        source: new ol.source.Vector({
        format: new ol.format.GeoJSON(),
                url: 'data/d'+i+'.geojson'
        }),
                style: null
        });
map.addLayer(d[i]);
}
```

The next statement sets the map viewbox to the building outline specified by bounding box coordinates “`map.getView().fit([2121856, 6019761, 2122054, 6019975]);`”.

The function search() carries out searching for specific request. In fact, the function accomplishes searching for doors number as I mentioned above. Coding for variable `f` is conducted by `concat()` method. It allows to join all the arrays for doors. Program gets all the features and determines them. Codes for this layer are revealed below. Additionally, there are some rooms like WC, stairs or elevators which do not have unique numbers on the

map. So, in a search box it is possible to type the name of attributes and to get desired result. Search query can be done only in English and it is not case sensitive.

```
// search for room
function search() {
   var q=document.getElementById('sQuery').value;
   var found=false;
// gather all doors from all layers
   var f=[];
   for(var i=-1;i<=7;i++)
   f=f.concat(d[i].getSource().getFeatures());
// looking for doors with id=q or name containing q case insensitive
   for (var a=0;a<f.length;a++) {
if (f[a].get('id')==q||(f[a].get('name')&&f[a].get('name').toLowerCase().indexOf(q.toLowerCase())>-1)) {
     sr.getSource().addFeatures([f[a]]);
     switchLevel(f[a].get('level'));
     map.getView().setCenter(f[a].getGeometry().getCoordinates());
     found=true;
//return;
              }
       }
     if (!found) alert('not found');
}
```

Function clearSearch() is created in order to reset search results. Codes are below.

```
function clearSearch() {
              sr.getSource().clear();
```

Function switchLevel() is needed to switch floors manually and automatically when you search for rooms. Codes are beneath.

```
function switchLevel(level) {
for(var i=-1;i<8;i++) {
        f[i].setVisible(level==i);
        r[i].setVisible(level==i);
        document.getElementById('l'+i).className=(level==i)?'current':'';
        }
}
```

The order of the layers is also important. Hence, if there is a change of the layers the program will not show some of them because of wrong hierarchy.

Running the application in Android Studio

Development of an application starts from the defining of the operation system on which is going to be to run the application. The working platform for the application is Android operation system. I used in my work the smartphone P8 Lite from Huawei brand. Some technical specifications of the phone are shown in Figure 10.

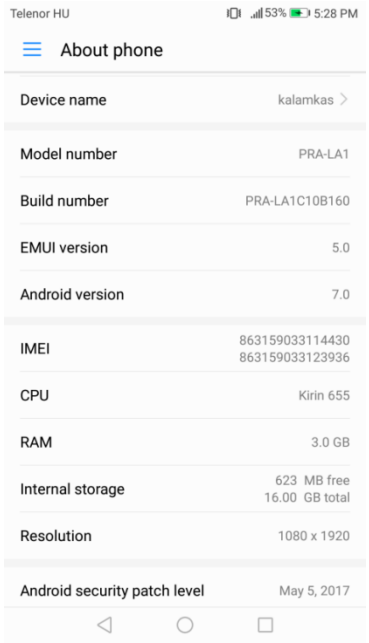


Figure 10. Technical specifications of Huawei P8 Lite smartphone

Since the aim of the thesis project is developing real application it was decided to create it in Android Studio environment for making applications. In spite of the possibilities of the running the application in a device emulator it is better to test the application on real devices, moreover, the final result is going to be the real working application.

So, Android Studio allows creating a project, giving a name of the application, changing its icon, resizing the screen and modifying displaying options and etc. The name of the application is “ELTE NorthBlock” with the icon of ELTE.

Steps of creating a project in Android Studio are illustrated in Figure 11.

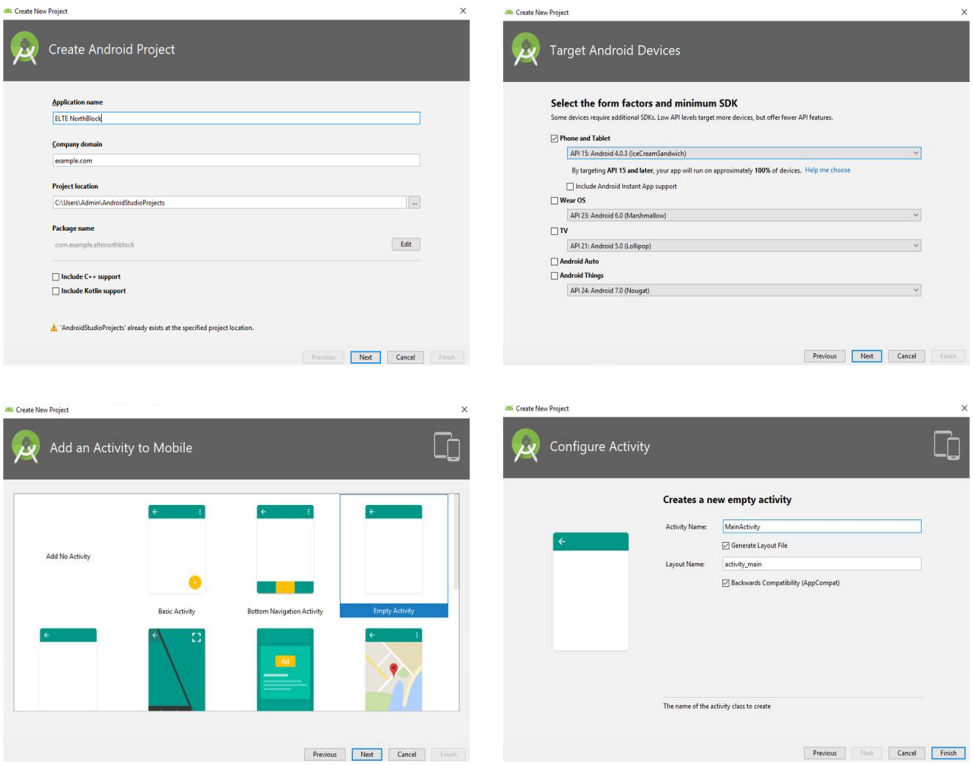


Figure 11. Steps of creating a project in Android Studio

To display webpage within mobile application I used WebView function which I described in previous chapter. Codes below enable to open "index.html" file together with "north.html" file.

```
package com.example.webviewtest;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        WebView wv=findViewById(R.id.wv);
        WebSettings ws=wv.getSettings();
        ws.setJavaScriptEnabled(true);
        ws.setAllowFileAccessFromFileURLs(true);
        wv.loadUrl("file:///android_asset/index.html");
        wv.setWebViewClient(new WebViewClient());
        }
}
```

It is possible to run an application both in emulator or real device. I connected my mobile phone to Android Studio, so when phone is connected it appears in pop-up window while running application as it is illustrated on the Figure 12.

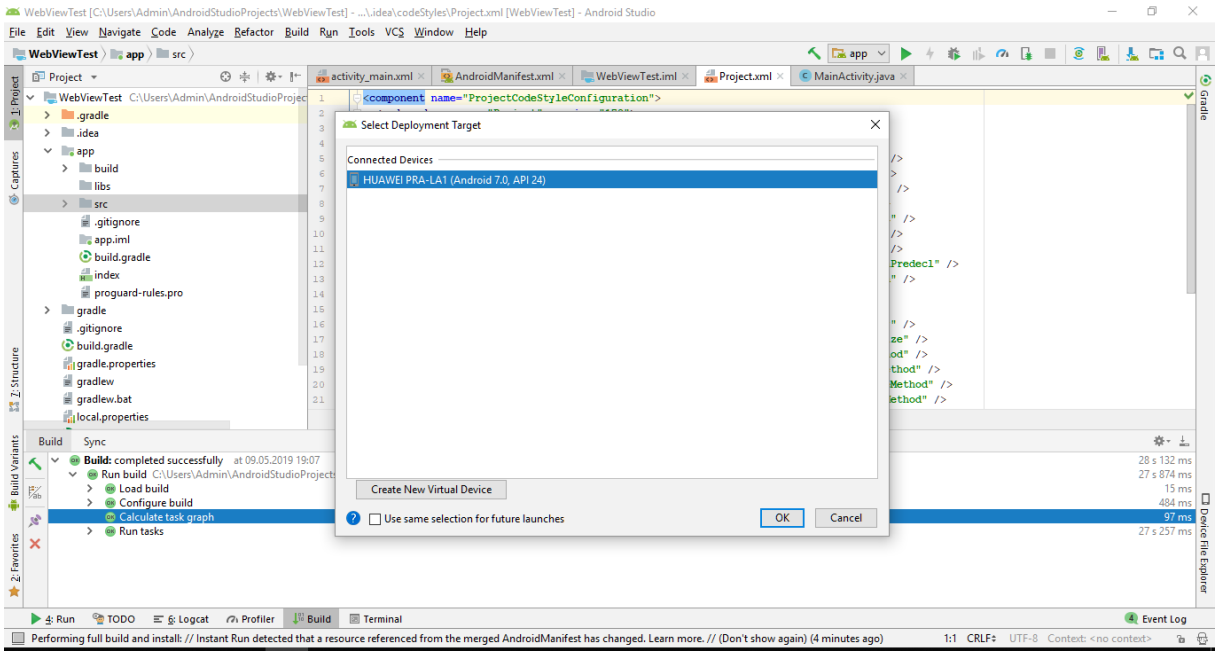


Figure 12. Connected devices in Android Studio

To connect the phone to the Android Studio environment I had to go to the phone settings and enabled USB debugging which is related to the advanced options of the phone. Figure 13 shows phone settings for connecting phone to the Android Studio.

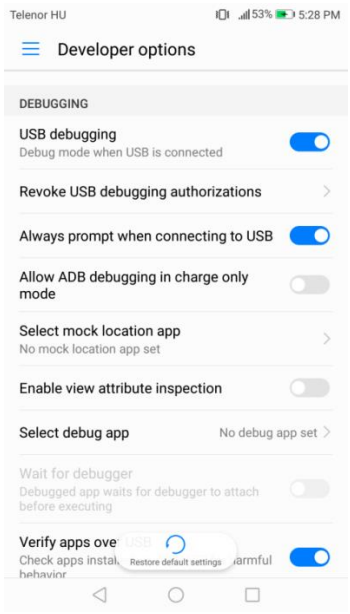


Figure 13. Phone settings

In the activities I changed screen size of displaying the application to the actual size of the phone Huawei P8 lite. Basically, it is also possible to change it to a smaller size of showing the content. However, I did prefer to remain full size because it gives you more spaces for indicating the content and more comfortable for eyes. Figure 14 reveals how to change the displaying settings.

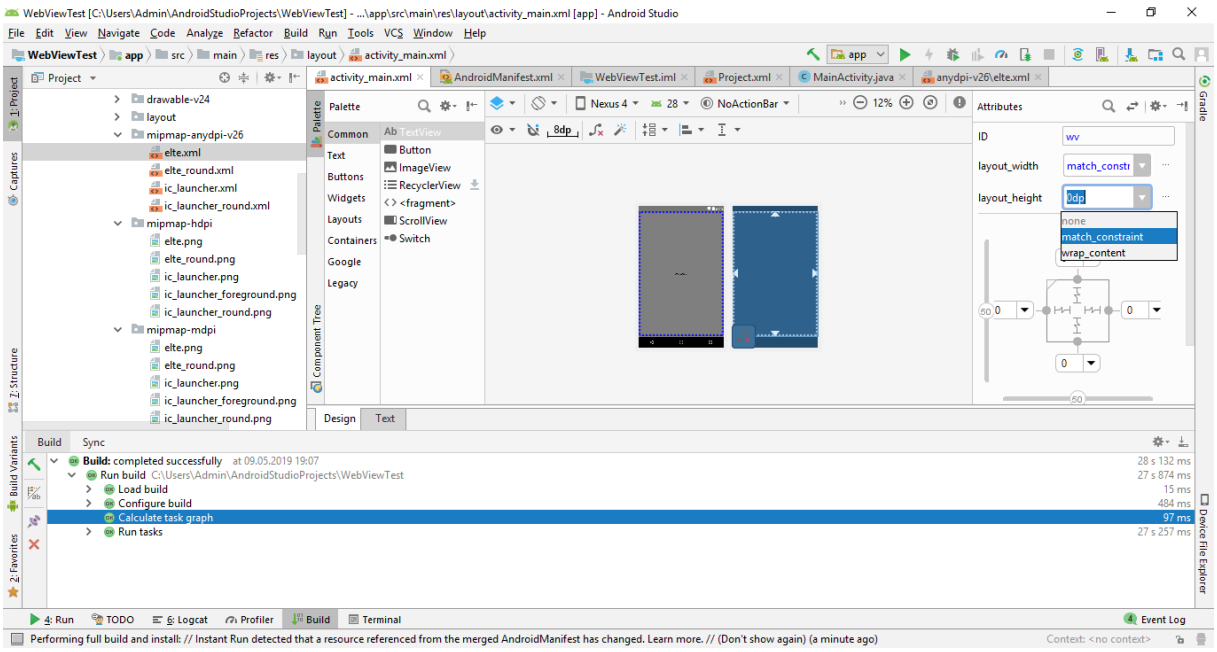


Figure 14. Application settings in Android Studio

Additionally, I set up the logo of ELTE for the application icon. I downloaded the logo of the university from the Internet in .png format, then modified it in the software and gave the name “elte” and I inserted in the codes afterwards. On the Figure 15 you can the codes and all the formats of the icon.

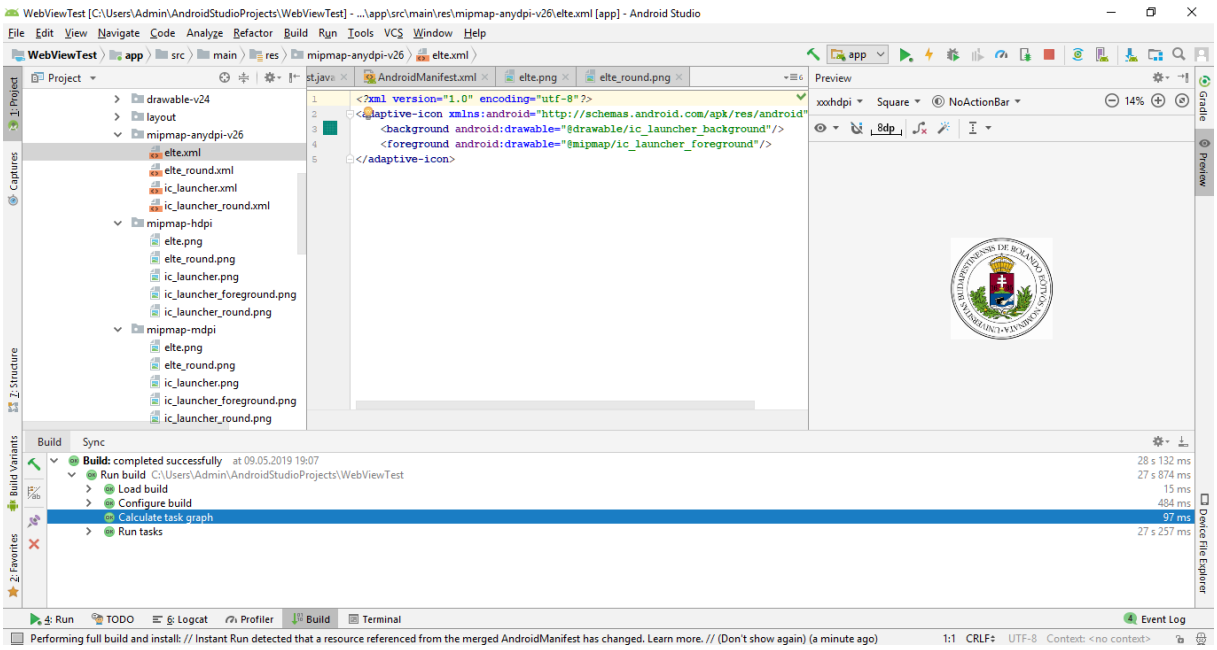


Figure 15. Icon settings in Android Studio

Challenges

While doing any project we face to different problem. So, as it is my first experience in creating an application and I am not proficiency in coding I have met some challenges. First of all, the understanding the differences of screen sizes of various devices. Hence, while coding this aspect should be taken into consideration. That is why it was confusing for what kind of font sizing to use because they are several. Even though using pixel sizing method is not a favorable for displaying webpages on mobile devices screen it remains absolute. Thus, even if you change gadgets for running the application all the settings for font sizes stays the same as it was initially coded. However, if the project is going to be successful and implemented the way of font sizing should be mulled and possibly changed to "em" units because it is scalable format. It means font size changes depending on the device screen size.

Designing and styling the application itself was also a bit challenging. Designing of application means to select necessary functions and additional data. So i made the application simple, easy in use with no extra data inside. Styling the application is hard in terms of finding perfect color matching for all elements of application. Moreover, size of application elements is also important because everything what is displayed should be readable.

The most big challenge and not solved is the searching rooms by attributive name. Ideally, when you request in the search box name of the room like “WC or elevator” it should

appear for that floor where you are or would like to see. Nonetheless in fact, the query shows all the attributes with the requested name. So, if you queried "elevator" then you retrieve as result all the elevators in the whole building.

Future plans

Once the mobile application already exists in testing mode it is easier to upgrade it. There are some tasks not done yet. So, as future plans there are several purposes such as elaborating not properly working functions of the application, updating the design, uploading the application in Google Play service, so the users can download it on their devices. Furthermore, it is possible to update the quantity of the attributive data in a way of adding the names, types, schedules of rooms.

Summary

Before implementing an indoor navigation system, it is quite important to understand the goal of the initiative, what resources do you have, which devices are going to be used and what is the audience. So, correct managing of the whole project is the first step towards successful achievement. Due to the growing popularity of the use of mobile applications it was quite actual of developing it for indoor navigation. Thus in the master thesis I tried to build an application which is might be upgraded furthermore in the future. The application "ELTE North Block" can be used by ELTE and exchange programs students, staff and guests of the university. The language of the application is English, however, it is easy in use, so any other language speaker is able to understand the logic of the application and utilize it.

In the thesis text all the stages of the project are described step by step and supported with the images where it is required.

References

Kaluža, M. – Beg, K. – Vukelić, B. (2017): *Analysis of an indoor positioning systems*. Zbornik Veleučilišta u Rijeci, Vol. 5 (2017), No. 1, pp. 13-32

Mulloni, A. – Wagner, D. – Schmalstieg, D. – Barakonyi, I. (2009): *Indoor positioning and Navigation with Camera Phones*. PERVASIVE computing

OpenLayers

https://en.wikipedia.org/wiki/OpenLayers Accessed 14 May 2019.

Quinn, M. (2018): *Ultra Wideband (UWB) for indoor Positioning*. https://www.leverege.com/blogpost/ultra-wideband-uwb-indoor-positioning

Sakpere, W. – Adeyeye-Oshin, M. – Mlitwa, N.B.W. (2017): *A state-of-the-art survey of indoor positioning and navigation systems and technologies*. South African Computer Journal

The GeoJSON Specification (RFC 7946)

http://geojson.org/ Accessed 14 May 2019.

Web Mercator projection

https://en.wikipedia.org/wiki/Web_Mercator_projection Accessed 14 May 2019.

Website for Android Developers

https://developer.android.com/guide/webapps/webview Accessed 14 May 2019.

White Paper, (2018): *Infsoft Indoor Positioning and Services*

https://cdn.infsoft.com/www/images/solutions/basics/whitepaper/infsoft-Whitepaper-EN-Indoor-Positioning_download.pdf Accessed 14 May 2019.

Acknowledgements

I would like to thank my supervisor the associate professor Dr. Gede Mátyás for mentoring me during the whole time of writing the thesis. From the first consultation about discussing the topic of the thesis he directed me to the right sources. Dr. Gede Mátyás gave me access to all required data. He was patient and supportive considering my lack of the knowledge in coding. Dr. Gede Mátyás answered to all my questions about confusing aspects of the coding and explained me some tricky questions and points of building an application. Additionally, Dr. Gede Mátyás was also replying to all my email on time, so i did not have to wait for his responses and feedback for a long time.

Additionally, I would like to mention Eszényi Krisztián. He completed master thesis in the previous year and built the database of ELTE Northern building. So that, I was able to use his research as an initial data for developing the application and completing my project.

DECLARATION

I, undersigned *KALAMKAS YESSIMKHANOVA* (NEPTUN CODE: *DVENCH*), declare that the present master's thesis is my original intellectual product in full and that I have not submitted any part or the whole of this work to any other institution. Permissions related to the use of copyrighted sources in this work are attached.

I AGREE / DO NOT AGREE to the publication of the accepted master's thesis in pdf form on the website of the Department of Cartography and Geoinformatics.

Budapest, *15th of May, 2019*

_____

(signature of the student)