EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPARTMENT OF CARTOGRAPHY AND GEOINFORMATICS

Indoor Navigation Systems in ELTE South Building

Raed S. Wahsheh

Student of Cartography MSc

Dr. Gede Mátyás

Senior Lecturer, PhD

ELTE Department of Cartography and Geoinformatics

Budapest, 2018

Contents

Introduction:

What is Indoor Navigation System (IPS)?

An **indoor positioning system** (**IPS**) is a method to determine place of objects or people inside a construction using radio waves, magnetic fields, acoustic signals, or other sensory information collected by mobile or tablets. (Wikipedia, 2018)

Another Description of IPS:

The easiest way to describe Indoor Positioning Systems (IPS) is that it's like a GPS for indoor surroundings.

Indoor Navigation Systems (IPS) can be used to detect people or objects inside buildings, usually via a mobile or tablet. Although the technology is newer than GPS, services that influence IPS are rapidly gaining grip in places like malls, hospitals, airports and other indoor sites where navigation and other location-based services (LBS) can verified to be crucial. (SenionLab)

The Importance of Indoor Navigation Systems

It is important to expect the future about (IPS), how much it will be usable? How far different nations is important for them, for example in urban areas it's more usable regarding to large indoor buildings such as shopping malls, skyscrapers, institutions or airports in crowded cities, but in rural areas where people live in small houses and work widely in outer open spaces such as farms it could be less important.

Leslie Presutti, the Senior Director of Product Management at Qualcomm says "*IPS can be as popular as outdoor navigation systems when consumers see benefits in either their time or pocketbook*". (The future of indoor positioning, 2016)

I disagree with Ms. Leslie Presutti about the equality of importance as outdoor navigation systems; I think it needs more research about people's habits and activities. How many times positioning system users spend their time outside and inside? How percentage is the indoor places coverage of earth regarding to the outdoor places? Of course it's less, but comparing to the time spent there it could be less or more important.

Indoor positioning

In the outdoor navigation systems the GPS is most usable device, GPS cannot be used inside because the radio signal cannot pass through the walls. (Gartner, 2010)

Relation to GPS

Global navigation satellite systems (GPS or GNSS) are not appropriate to establish indoor areas. To make positioning signals widely spread, combination of GPS and indoor positioning can be found.

Increasing the microchip processing power made the GNSS receivers more accurate. Nowadays, GNSS receivers can receive satellite signals in major indoor places and continues trying to find the 3D position.
Whenever the sensitivity of receivers are increased the technology of A-GPS is useful, then the calendar and other information are send by a mobile device. (Wikipedia, 2018)

Route Presentation and Communication

A different significant side of indoor navigation is how to convey route information efficiently (Gartner and Uhlirz 2005). A well prepared route form will allow users to easily find their way. According to the format of its presentation we will differentiate several indoor navigation systems: map, written or oral instruction, signs, photos, movies, 3D display, etc. (Gartner, 2010)

Challenges:

Finding extremely accurate location: Finding an accurate position indoors requires an accuray of about 2 meters or less.
For example, it allows us to differentiate between entrance of two offices, detect the floor and to find the vehicle in the parking.

Current available outdoor navigation and mapping softwares use the Global Positioning Systems (GPS). GPS is a useful technology which is available in most modern mobiles, and it was examined and succeeded for navigating in the world. But the accuracy of GPS is not the best result, the range of accuracy is between 2 to 15 meters and this is not the range required for indoor positioning, even if GPS worked inside indoor places then its not useful for accuracy at least.

Some developers used WiFi and Bluetooth sensors to determine an indoor location.
But the challenge is that WiFi and Bluetooth invention were made to transmit data not to determine the location, therefore the result would be an inaccurate location when using these inventions. (SPREO development team, 2014)

Indoor positioning: Nowadays, indoor navigation systems uses radio signals (Wi-Fi, Bluetooth, Radio Frequency IDentification, etc.) for finding position, which leads to cause errors in signal and weakness, like the Radio Frequency confusion.
More researches and tests of the indoor positioning systems are required. (Gartner, 2010)

Designing for inaccuracy: Indoor navigtion systems can tell you which room you’re in while outdoors cannot tell you exactly for instance where you are standing next to. This depends of course on different factors such as the way how you are holding your phone. (Farina, 2011)

Industries Use Cases for Indoor Navigation & Location

- **AIRPORT & STATION (Help travelers and passengers):** To show all important points, locate machines for payment, WC and ticket offices, and any travel information needed, locate Gates and aviation offices.
- **OFFICE & INDUSTRY (Better orientation):** Locate offices or halls, buffet, reception service.

- **RETAIL & SHOPING CENTER (Create attractive added value shopping experience):** Gather data and offers, marketing depending on location, for instance closest tobacco shop to entrance 1.
- **HOSPITAL (Patients services, orientation):** Indoor navigation for patients and their guides, doctors and nurses, monitoring the patient who needs help and monitoring whole staff which are near.
- **TOURISM & MUSEUM (Offering a rich content):** Map of important tourist buildings, navigating to them by car, bicycle, public transport or walking. Maps showing number of rooms in a particular building, saving time and cost to avoid undesirable buildings.
- **EVENT & EXHIBITION (Create interaction with invitees):** Reception desk, submitting guidance maps in exhibitions and conferences. Navigating to points of interest and when to arrive there. (COPPELIS)

Mapping Indoor Building Space is a Smart Business Move

ArcGIS is a great mapping platform in real estate and workplaces.

The next observation from GIS workers is along these lines: "Do you know how hard it is for us to quickly answer this simple question and see the results on a campus, building, or workplace map?"

This discussion holds true for different users as well as those that have used GIS for several years. Often, if the user is already using GIS technology to find data about location for his outdoor possessions he will ask, "Why our indoor space not stored well in GIS?

There are some of answers to this question, but the response is concentrated on the business requirements and existing state of each client for map management and location-based data.

The reason of bringing indoor buildings into GIS because it helps to get better and efficient operations reduces risks and makes staff productivity better.

Guests and clients can use navigation applications on smart phones. Employees can sit beside people they deal with to make cooperation better. They can also use map navigators on their mobile phones to find rooms for meeting. It can be applied by using a mapping hub or portal.

The interest is to minimize influences for guests, employees in a building so everyone can plan his visit before arriving to the building.

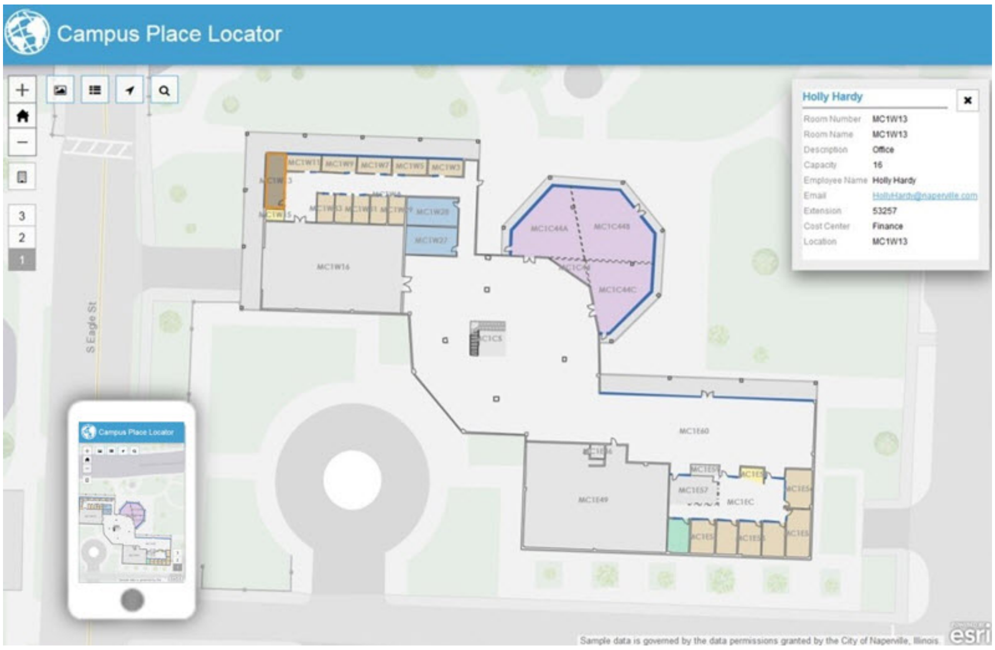

Figure 1: Maps can help you locate employees' offices and quickly find their email addresses and telephone numbers

ESRI, PenBay and GISi are working with a various users to provide:

- Essential data model for universities, colleges, and travel connections.
- Commercial system projects and workflows.
- Patterns, Map applications (e.g., see figure 4)

These models, patterns and workflows let users easily work with ArcGIS in to their possessions and practice their business. (Young, 2015)

Future Solutions:

Latest future expectations about indoor navigation systems came by Google. The solution is to use LED bulbs in combination with IPS (Indoor Position Systems) to get assistance inside buildings to determine your location and take you to the desired location.

The GPS loses its signal inside buildings, therefore it's important to use IPS which depends on technologies such as Wi-Fi, Beacons, Bluetooth, etc.

Google thinks to use visual light communication for IPS to let the indoor navigation technology more precise.

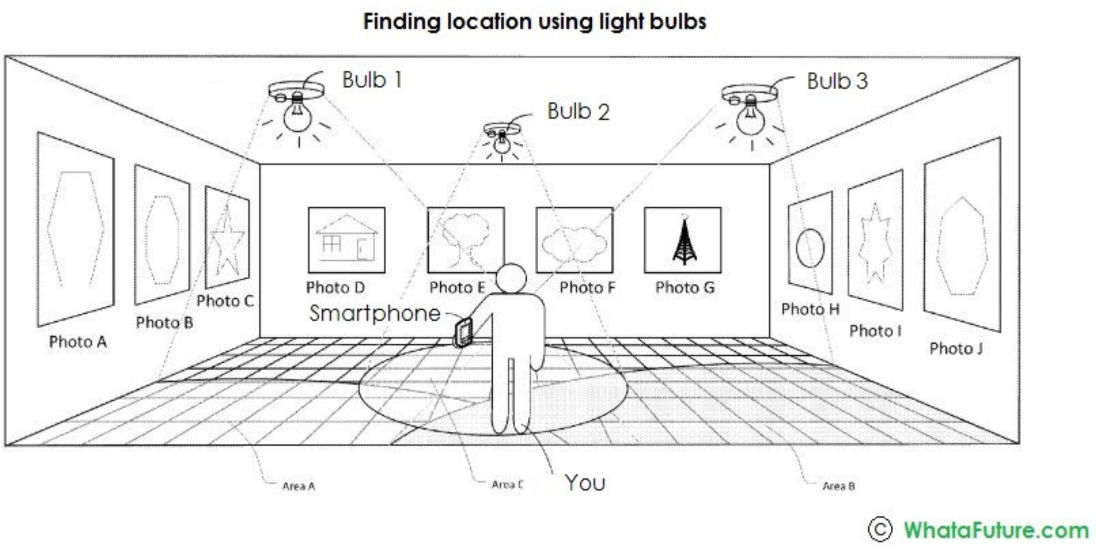

Figure 2

In a building, LED bulbs are everywhere. Each room in a building uses to have a different kind of light orders and then different frequency, intensity, color like property of light differ between points.

Google Maps will append a photo sensor to your mobile device to find the variances, then it will be used with method of triangulation to locate your place.

Other sensors will be used such as accelerometer, gyro-meter, and compass for orientation to get a better accuracy. (WhataFuture.com)

Practical Mission:

It is important to mention that my job was administrated by advisor *Dr. Gede Mátyás*, In addition, sometimes with discussion and cooperation with my colleagues *Eszényi Krisztián, Barancsuk Ádám and Csaba Szigeti*.

In February, I was consulting with my advisor and colleagues all the challenges for my project, what is the process, what building I must cover which was decided later as the Southern Building of ELTE University and this refers to my thesis title? Which floors and which software and devices I should use to collect data from field.

I gathered all data needed to be used for this project by support from my colleague, each data was provided like this from my colleagues:

- **Building.shp**, it is the structure of the south building of ELTE University, there is 1 record in this shapefile, with fields: name "Déli épület" which means south building and ID with number "1" value, also with other two unimportant fields which is generated automatically "FID", "Shape".
- **Floor_00.shp**, it is the structure of the underground floor, and consists of the following fields: "ID" with value "9", "building_i" with value "1" and "floor_seq" with value "-1".
- **Rooms_00.shp**, it is the structure of rooms for underground floor, and consists of 159 rows with fields named: "ID", "Floor_ID", "room_numbe", "room_name", "room_type", "org_unit_i" and "capacity".
- **Floor_0.shp**, it is the structure of the ground floor, and consists of the following fields: "ID" with value "1", "building_i" with value "1" and "floor_seq" with value "0".
- **Rooms_0.shp**, it is the structure of rooms for ground floor, and consists of 120 rows with fields named: "ID", "Floor_ID", "room_numbe", "room_name", "room_type", "org_unit_i" and "capacity".
- **Floor_1.shp**, it is the structure of the first floor, and consists of the following fields: "ID" with value "2", "building_i" with value "1" and "floor_seq" with value "1".
- **Rooms_1.shp**, it is the structure of rooms for first floor, and consists of 134 rows with fields named: "ID", "Floor_ID", "room_numbe", "room_name", "room_type", "org_unit_i" and "capacity".

All those shapefiles provided to me by *Barancsuk Ádám* via email.

My device was the Samsung S5 android with weak performance and lots of interruption, sometimes I was repeating the capture to ensure data saving, I have been using the **Qfield ver 0.10.4 – kesch** version mobile software which is quite similar to the **Quantum GIS (QGIS)** software on PC's, it was also interrupting me in several occasions.

In this practical part I created the doors (point shapefile) to store doors in each floor, I renamed each shapefile according to its floor, for instance, doors_00 is for underground floor, doors_0 for ground floor … etc.

There are three fields for doors shapefile:

- Type (type name of the door), it should be as following:
  - Office.
  - Lab.
  - WC.
  - Stairway.
  - Elevator.
  - Classroom.
  - Computer Lab.
  - Lecture room (room for classes, but it has stairs and doors is in both floors).
  - Storage (a storage place).
  - Bufe (buffet).
  - Exit (exit door).
  - Kitchen.
  - Library.
  - Museum.
  - Other (any other description or an unknown place).
- ID (it refers to the unique door number or number of 2 doors when 2 rooms are connected with one door).
- Name (a description of the door type like name of professor of an office or gender of rest-room in case of type is WC).

FID	Shape *	type	id	name
108	Point	lecture room	0.804	Lóczy Lajos Terem
109	Point	lecture room	0.805	Fejér Lipót Terem
111	Point	other	0.805	Fejér Lipót Terem
126	Point	lecture room	0.821	Bolyai János terem
129	Point	lecture room	0.821	Bolyai János terem
122	Point	lecture room	0.822	Mogyoródi József Terem
125	Point	lecture room	0.822	Mogyoródi József Terem
117	Point	lecture room	0.823	kitaibel Pál terem
121	Point	lecture room	0.823	kitaibel Pál terem
27	Point	stairway	00.105	stairway
45	Point	other	00.107	Hőközpont 1.
49	Point	other	00.108	Hűtő
50	Point	stairway	00.109	stairway

Figure 3: Description of Doors Shapefile

Some exit doors which are not accessible could not be recorded because I need permissions from concerned people and this project was performed under the all possible permissions as a student who is working formally for a thesis topic and holds an official document that show others what I am doing, but not for a formal worker in the building.

Along the 3 field working weeks I encountered problems from some aggressive workers that didn’t admit my official task despite of showing them the official document, but major of them were helpful and some of them were supportive.

Sitting areas, halls without doors or equipment storage rooms such as the extinguisher or the teachers special devices are not considered too even if there is an ID number written and it is skipped, therefore you can see some door numbers missing in the database.

Here are some captured statistics:

- 417 doors were recorded for 3 floors.
- There are 20 doors with no information/empty or temporarily unused.
- There are 413 rooms in the database.

Concept of current dissertation (Indoor Navigation Systems in ELTE building):

In previous introduction, we have read about indoor navigation systems according to specific sensors, calculating the position is dependable on specific technology that reacts with the device which is hold by the human who wants to calculate the indoor position.

In our studying thesis there is no sensors used, there are unique door numbers, sometimes there are rooms which have more than a door, named with same ID number and adding to it sometimes an alphabetic letter such as “A” or “B’, generally we are not going to study and explain the door numbering system in the university, it was the task for architect engineers or the administration of the university.

The ID number of the door will not be the primary key as it maybe seems to, because some doors don’t have a number, they are empty rooms or not assigned yet, or could be on the way of renovation, therefore later we will see in the database a random number generated and considered as a primary key for a door (node).

Figure 4: doors are connected with routes

From the midpoint of the corridors we start the route and finish it at the same point, we draw the whole root in all corridors wherever there is a door; the door is connected with the route perpendicularly to complete the whole root.

Sometimes it’s not simple to draw the root in the middle of the corridor because the corridor’s width varies, therefore when this happens we take the midpoint of the narrowest place in the corridor and start drawing the route from there and connect it with the route in other places.

Considering the midpoint in the corridor to draw the route doesn’t mean we will get the shortest route between doors, but of course it is the ideal way to draw it because it is not logical to draw several routes in single corridor which may lead it to conflict each other.

Manipulating Data:

If you want to add a column to a table or create a table into a database, columns are created in different data types. The data types are categorizations of different values and many operations can be done on these data types, and the values of these columns and tables are stored in the data base.

ArcGIS contains an attribute table where the data can be viewed and manipulated; you can delete, update or add a column, you can work on different column data types. After you access the table in database by a Database Connection in the catalogue tree or by running a query in ArcMap, then the ArcGIS will filter any not supported data type. If you want to access the database tables direct you are not able to view the unsupported data types by ArcGIS interfaces and of course you cannot edit them. The same way when you copy unsupported data types with ArcGIS, the supported data types only will be copied, and when copying it to a geodatabase or any other database you cannot see the columns with unsupported data types. (ESRI, 2018)

Data Type:

I used ArcGIS for Desktop to manipulate my data; here is the main data type in ArcGIS which I used:

Text (String)

The text field includes a collection of string and/or numbers, For example a road name or any other text details.
Instead of repeating text description in a field a coded value could be used in the geodatabase in order to reduce size in databases and make it more efficient and faster. For example we can write the road type in the field which is originally text and assign number 1 to main roads then 2 to paved roads, etc.
The coded value anyway should be understood by the target user. (ESRI, 2018)

Text or (String) order can be either ascending (A to Z) or descending (Z to A). By the way numbers in this field are treated like text, which means they will be sorted alphabetically and will not be considered on their values. For example, sequence numbers 1,116,3,22,202 will look like this 1,116,202,22,3 after sorting.

Also any other numeric operations like (max, min, sum, avg, etc.) could not be performed on this data type.
If the whole field is converted to a number field then the field's value could be considered as numbers and make different calculations. (DEMPSEY, 2013)

Because the fields in doors shapefile are string it usually would not be a primary key, frequently numbers are, primary key in this shapefile would be generated later when we store it in database.

Techniques:

Here are the main techniques I used in ArcGIS for Desktop:

Snapping:

Snapping makes drawing easier, simplest and accurate and where errors are eliminated. In ArcMap we can start editing the spatial data and enable snapping feature, then when drawing we can end a pointer and snap it to endpoint, vertex, edges or any other element when we place the pointer close to certain tolerance, then it lets us easily locate a feature where we need exactly.

Snapping toolbar could be enabled by right clicking the mouse button in the toolbar then clicking on snapping, you will find all options and settings in this toolbar.

Snapping could be available not only by editing our data in ArcGIS, but can be found in georeferencing or when using the measure tool. The snapping is already turned on when opening your application and the available snapping features are endpoint, vertex, point and edges.

You can completely close the snapping feature when you disable the mark next to "Use Snapping". (ESRI, 2016)

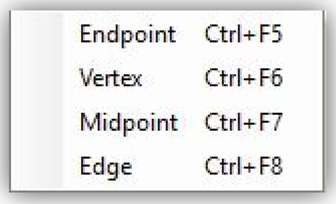

Figure 5: Snap To Feature Options

Most usable Feature Option:

Midpoint (Midpoint for nearest polygon or polyline). (ESRI, 2017)

Midpoint feature facilitates defining the midpoint of a segment, rather than using a measure tool to define the distances to calculate the middle point.

The midpoint is helpful in different situations, such as defining the center line of a road or defining a center point in a corridor and there are many other examples. When we draw the sketch sometimes we don't need to finish it at the edge, endpoint, it could not precisely finish at the midpoint if we want to do it manually, therefore we use the midpoint tool.

Parallel to Line:

A parallel feature allows the user to draw a parallel line depending on a present line and will be parallel to it. When editing the data use right click of the mouse on the present line which you want parallel to and choose parallel, ArcGIS copies this feature and calculates the slope to make the new line parallel to the present one.

Perpendicular to Line:

Perpendicular feature could be used to finish the sketch perpendicular to another present sketch. Similar way to parallel feature by right clicking the mouse on the present segment or line which needed to be perpendicular to we can choose the perpendicular option from the menu.

Many examples on perpendicular feature, for instance, a secondary road line must finish perpendicular to an existing main road.

The perpendicular feature is somewhat related to parallel feature, for instance creating a line Which is parallel to another that ends on the same segment or line means that the created line is also perpendicular at that line.

Parallel and perpendicular, both features are related to lines (polylines) and could not be used with other feature classes.

Creating a segment using an angle and a length:

For creating segments with a predefined angle, use the Direction feature. The desired angle value can be input, angle use east as 0 and is positive with counterclockwise. For example, choose click on the assumed direction and enter 60 for angle, then the segment will be constrained to a 60 degree angle.

Use the Direction/Length feature for a combination of Direction and Length feature; in the length feature you can input the desired length for your segment. (ESRI, 2016) (Mann, 2006)

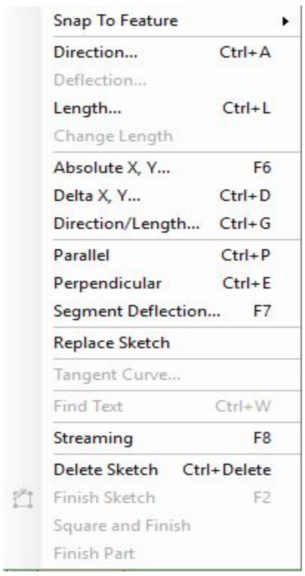

Figure 6: Snapping Features in ArcMap 10.4

Feature Vertices to Points:

It creates a new shapefile that contains points from the vertices of the input shapefile whether it's in the mid, start or end, reach this feature from the ArcToolbox. (ESRI, 2016)

Feature to Line:

It creates a new shapefile that contains lines from converting polygons to lines or splitting lines and explode them at the intersection points, reach this feature from the ArcToolbox. (ESRI, 2018)

Working with group layers in ArcMap:

It is necessary to separate each floor in order to avoid the data conflict because each floor has its own shapefiles, grouping layers allowed me to separate each floor and easily operate with the data, when grouping the layers I can give a name for each group which particularly is the floor name, it also helped me changing the orders for each floor, for example, it was important to place the layers in this order:

1. **Doors**: it is a point layer, important for visibility.
2. **Route**: line or polyline shapefile must not be overridden by polygons.
3. **Rooms**: rooms are a polygon layer that is important to appear within the building shapefile.
4. **Floor**: it is important to distinct which floor it is but under the building shapefile.
5. **Building**: the background layer which is less important to be visible.

To make grouping of specific layers just select the layers wanted to be grouped and click right mouse button and choose "group".
I had 3 groups for the 3 floors which a worked for, their name were the word "floor" followed with the floor number.

Data Frame Properties:

It is the given name of the project in ArcMap, it includes also a description, the unit used "meter", in my project the closest and appropriate name is "South Building", the field where I was working, not necessary here to define a coordinate system because it's an indoor navigation project "no sensors".

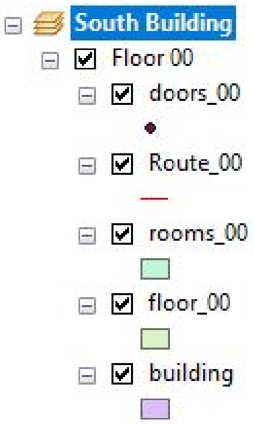

Figure 7: Data Frame Name and Group of Floor "00"

Data Processing:

After data is ready it must be stored in a database, therefore the simplest way to do this is via QGIS, actually not all data required for this mission, these data we need (*See Figure 10*):

1. **Doors**: Doors are needed to specify the beginning and ending point of the route, it is the (id) which will be used later in the programming step.
2. **Intersections**: it is a point layer which was generated from the doors and routes layers, doors are added from same existing shapefile and the other nodes are generated from routes layer, whenever there is an edge, vertex (start or end) of a segment or a node opposite to a door it is added to this layer. The technique used was mentioned before; it is the (Feature Vertices to Points).
3. **Edges**: Edges are the lines, it is the lines that created for the desired routes, and it is connected later together to create the whole route.

Data can be imported into different databases such as SQL Server, PostgreSQL, PhpPgAdmin or Oracle, here we will show first how to use *PostgreSQL* databases, but first we must create PostgreSQL connection to connect to the current database in PostgreSQL, to do this we use shortcut "CTRL + Shift + D" or by choosing "Layer" then "Add Layer" then the "Add PostGIS Layers" option, then we click for "New" connection.

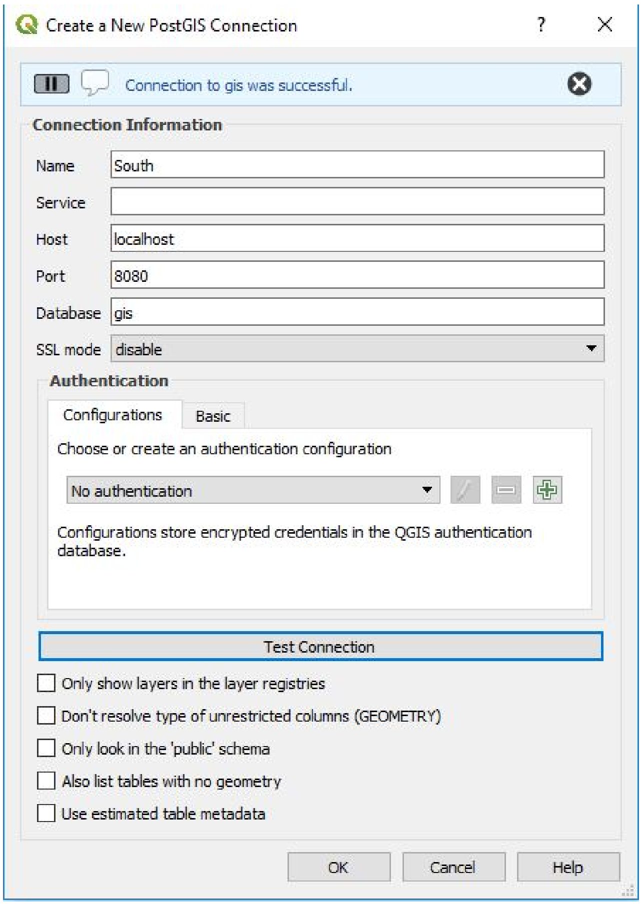

Figure 8:Successful PostgreSQL Connection

A connection is a method which gives us permission to have access to the database which was created before, without this connection we are unable to add tables, views, functions, etc.

I must insert name of database which is “*gis*” in my status, and using the schema named “routing”, a schema is a collection of tables, views, functions, sequences, etc.

In previous steps I used the *PostgreSQL* sample to view database and access to it, but in my project the database is managed using phpPgAdmin. PhpPgAdmin is web-based administrative tool for PostgreSQL. (Creative Commons, 2013)

Figure 9: phpPgAdmin Web Based Databases

PhpPgAdmin is web based PostgreSQL viewer powered by PHP, PostgreSQL version in my project is PostgreSQL 9.6.1.

Why choosing PhpPgAdmin web-based viewer?

- Easy to access via internet from anywhere.
- Data Security, no one can access my data except predefined users.
- Data Storage saves free more space on our personal computers.

My database name is “gis”, and I can access phpPgAdmin by a link to see my database, and also I can manage my data, I can update, delete or add new tables or do anything as I do in PostgreSQL.

Merging Intersections with Doors and Eliminating Duplicated Nodes:

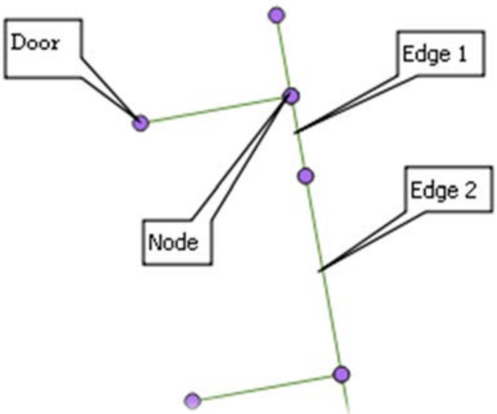

Figure 10: Intersection Nodes Separates Edges

When I described before this dissertation, it aims to create a series of graphs (edges) to make the whole route, the connection between the edges are the Intersections, whenever I have more nodes I will get more edges and the route would be more accurate.

To draw the whole route we need all nodes in database, not just doors but node intersections also, as intersections contains the doors and vertexes of lines therefore doors and other nodes will be duplicated.

We must eliminate the duplicated data to avoid confusing the user and errors in programming; I can eliminate duplicated data either by altering the table in PhpPgAdmin or exporting the data and removing duplicated data via PostgreSQL and importing it again into the PhpPgAdmin from my user.

Creating a table in my database of intersections and doors with eliminating duplicated nodes requires using this query:

```
Create table nodes_0_del as (
   select n.gid, d.type, d.id, d.name, n.the_geom from
      (select min(gid) as gid, the_geom from intersections_0_del group by 2) n
      left join
      doors_0_del d
      on n.the_geom=d.the_geom order by 1
)
```

0 is the level of the floor, it is the ground floor, and del refers to “déli” which means “south” in the Hungarian language referring to the south building of ELTE where the working field is.

We repeat the same query to create tables for all floors in the building, available floors in my data is -1, 0 and 1; therefore I will create the nodes_-1_del and the nodes_1_del.

Some node intersections does not appear distinctly on the map, but we can see its data in the attribute table, this returns to the “*Feature Vertices to Points*” tool that we used and then generated these nodes automatically, but this does not matter because this query task is to eliminate duplicated nodes.

Adding level field to nodes_*_del and edges_*_del:

The level field represents the following indicators:

- (-1): refers to underground floor of ELTE south building.
- (0): refers to ground floor of ELTE south building.
- (1): refers to first floor of ELTE south building.

(Level) field type is (integer) in all tables (nodes_del) and (edges_del), the field could not be null, by default each level is assigned to its floor number.

The (level) field is added to tables (nodes_-1_del), (nodes_0_del), (nodes_1_del), (edges_-1_del), (edges_0_del) and (edges_1_del).

The nodes_*_del table contains following columns:

- Gid (integer).
- Type (character varying (80)).
- Id (character varying (80)).

- Name (character varying (87)).
- The_geom (geometry (Point)).
- Level (integer).

The edges_*_del table contains following columns:

- Gid (integer).
- FID_Route (integer).
- Id (integer).
- The_geom (geometry (LineString)).
- Level (integer).

Merging various levels into one nodes_all_del and one edges_all_del table:

After creating tables for all floors, these tables must be merged together, the 3 tables of (nodes_*_del) are merged together and named (nodes_all_del), while the 3 tables of (edges_*_del) are merged and the output is (edges_all_del).

Let's see example output of table (nodes_all_del) in figure (11):

gid	type	id	name	the_geom	level
10005	other	1.826	Stúdió	01010000001B0870F2BADE2341B5B78FA766DC0C41	1
5	NULL	NULL	NULL	01010000000C32C268B8DE23415616F1CF6BDC0C41	0
6	NULL	NULL	NULL	01010000001FD7BC25B9DE23415BF3DB2428DC0C41	0
-9994	NULL	NULL	NULL	0101000000A93B7AA9E6DE23410C8641724BDC0C41	-1
10007	other	1.820	Hajós György terem	0101000000C9A63390B3DE2341CD56105B6BDC0C41	1
-9993	storage	00.822	Raktár	01010000004ACC35ACE0DE234108FD3ED75CDC0C41	-1
10008	NULL	NULL	NULL	01010000009E6EC69AB7DE2341EA65526272DC0C41	1
-9992	NULL	NULL	NULL	010100000043B6DAAAE5DE2341C07C3C9960DC0C41	-1
9	other	0.821	Raktár	010100000002C00980BADE2341C78558836EDC0C41	0
10011	NULL	NULL	NULL	0101000000101FE767A8DE2341E6CB55CD75DC0C41	1
11	NULL	NULL	NULL	01010000005D8860CBB2DE2341C59D7E0A74DC0C41	0
-9989	NULL	NULL	NULL	0101000000B417F17CF3DE2341DBD37A0DD8DC0C41	-1
12	NULL	NULL	NULL	01010000001C0483C9A8DE2341AA28554B72DC0C41	0
10012	NULL	NULL	NULL	010100000026A268F9ADDE23412DD1797C7FDC0C41	1
-9988	NULL	NULL	NULL	0101000000CD91D70DF1DE23413345364AD6DC0C41	-1
13	classroom	0.820	Hunfalvy János terem	0101000000BBF3C4E4B2DE234134D1DEF26ADC0C41	0
-9987	NULL	NULL	NULL	010100000042D536B0F3DE2341BDE29C32D8DC0C41	-1
10014	NULL	NULL	NULL	0101000000D88634E0B4DE23419E532A7D8BDC0C41	1
-9985	other	00.728	Műszer-és terepi eszköztár	010100000006751F41F4DE23412EE53AB1CBDC0C41	-1
15	NULL	NULL	NULL	010100000000295A302B6DE234174F3309A74DC0C41	0
10017	other	1.819	Matematika Tanszékcsoportok (Riesz Frigyes Terem)	0101000000794A8833ACDE23414C5E23CC8FDC0C41	1
-9983	NULL	NULL	NULL	010100000010885A5BF6DE234169FD5C21DADC0C41	-1
18	NULL	NULL	NULL	010100000053BAF4D7B7DE2341B4BE1BEC74DC0C41	0
10019	other	1.701, 1.702	Földtani olvasó és tárgyaló	0101000000D1DA51CFEDE23419781BEC6C5DC0C41	1
-9980	NULL	NULL	NULL	0101000000CD5479BBF8DE23417C51C8D9DBDC0C41	-1
10020	NULL	NULL	NULL	01010000006D58714402DF2341E0CD2DC9C8DC0C41	1
22	NULL	NULL	NULL	010100000098A4FBFAB6DE23417D0010D382DC0C41	0
-9978	NULL	NULL	NULL	01010000006E9DECDBF8DE2341E6B448F1DBDC0C41	-1
10022	NULL	NULL	NULL	0101000009174751104DF2341028D3800A1DC0C41	1
23	lecture room	0.821b	Bolyai János terem	0101000000B0283E21B9DE23415CDEB4F584DC0C41	0

Figure 11: nodes_all_del Table

As we see type refers to all node types, the null text refers to a node generated automatically by the tool that converted vertices to points, therefore there is no description for these nodes, but the non-empty fields refers to a door and the type is written also. Id field could not be a primary key because it has null values, and intersection nodes always are null. We can see also there are various levels in the new table because of merging.
The queries used to make these tables:

- *Create table nodes_all_del as ((select * from "nodes_-1_del") union (select * from "nodes_0_del") union (select * from "nodes_1_del"))*
- *Create table edges_all_del as ((select * from "edges_-1_del") union (select * from "edges_0_del") union (select * from "edges_1_del"))*

The Union operator is used to unify the output of 2 or more select statements, the statements must have the exact number of columns and must be within the same data types, also they must be within the same order.

While union selects only distinct values, the Union All operator selects also duplicated values in all statements. (W3Schools)

Then we need unique numbers in the field "gid", so we will add the level number times 1000 to it by using these statements:

- *update nodes_all_del set gid=gid+10000*level*
- *update edges_all_del set gid=gid+10000*level*

Since the id field was unable to be assigned as a primary key because it consists of null values, the gid became a primary key in the original (*nodes_*_del*, *edges_*_del*) tables, take a look at figure (12) to see the constraints of (*edges_1_del*) table for example.

Name	Definition	Actions	Comment
edges_1_del_pkey	PRIMARY KEY (gid)	Drop	

Figure 12: Constraints of (edges_1_del) Table

Adding indexes to nodes and edges table:

What are indexes?

Indexes are used to bring data very fast from databases, speeding up the search process and querying is the main task of indexes.

When creating an index we give a name of the index then “on” command followed by the target table and column name required to be indexed. (W3Schools)

Here is an example of index in our database:

Name	Definition	Constraints	Clustered?	Actions			Comment
geomi	CREATE INDEX geomi ON edges_all_del USING btree (the_geom)		No	Cluster	Reindex	Drop	

Figure 13: Index for (Edges_All_Del) Table

In (*Edges_All_Del*) table, index was created on (the_geomi) column were type is geometry (LineString) and the field is too long, it contains both characters and numbers, this leads increasing of storage size in the column and table, which requires to find a fast process to retrieve the data (index). This applies also on (*Nodes_All_Del*) table and the same column, so indexes are created everywhere these fields exists.

Creating routable edges table:

Here we will the create the table that contains all route segments, but only the routes within the same floor, later we will make other routes connecting floors together by using elevators or stairways, see figure (14) for (*routable_del*) table:

id	source	target	cost
-9999	-9999	-9998	1.21718038035247
-9998	-9999	-9996	3.74248856994774
-9997	-9996	-9994	0.756022729698282
-9996	-9993	-9992	2.54114251711144
-9995	-9992	-9996	1.93437323639941
-9994	-9989	-9988	1.23677837215429
-9993	-9987	-9989	0.101769626884379
-9992	-9985	-9987	1.58859046958858
-9991	-9983	-9987	1.35595008874146
-9990	-9983	-9980	1.20704607715311
-9989	-9980	-9978	0.0644100176099641
-9988	-9977	-9978	1.57886459038755
-9987	-9978	-9974	4.3775551861708
-9986	-9973	-9974	2.15176427997707
-9985	-9974	-9970	0.193880602361839
-9984	-9970	-9968	0.417604371785088
-9983	-9967	-9968	3.62343775994394
-9982	-9965	-9967	1.78628604109639
-9981	-9963	-9989	1.59001671807177
-9980	-9961	-9967	0.487468561816848
-9979	-9959	-9980	1.62816426620958
-9978	-9957	-9961	4.16658305544011
-9977	-9955	-9957	1.74911293898902
-9976	-9953	-9952	2.39917936494456
-9975	-9952	-9992	17.633595377953
-9974	-9949	-9957	3.87816228595219
-9973	-9947	-9961	1.78847641290592
-9972	-9945	-9949	3.65041345623696
-9971	-9943	-9945	1.68779983737676
-9970	-9941	-9940	1.1440635094502

Figure 14: (routable_del) Table

We created a source and a target fields which represents the start and end node, when connecting them together we will get the route (edge), the cost is a distance unit that represents length of the given edge, it is important and will help us later for computing the shortest way between two existing doors.

The SQL query used to make (*routable_del*) table is:

Create table routable_del as
(Select u.gid as id,n1.gid as source,n2.gid as target, st_length(u.the_geom) as cost
from nodes_all_del n1, nodes_all_del n2, edges_all_del u
where n1.the_geom=st_startpoint(u.the_geom) and n1.level=u.level
and n2.the_geom=st_endpoint(u.the_geom) and n2.level=u.level order by 1)

Data types of each column in (*routable_del*) table as following:

- ID (integer), as source table (*edges_all_del*) from the (gid) field.
- Source (integer), as source table (*nodes_all_del*) from the (gid) field.
- Target (integer), as source table (*nodes_all_del*) from the (gid) field.
- Cost (double precision), as source table (*edges_all_del*) from the field (the_geom).

Adding edges for stairs (where the coordinates are almost the same) and elevators:

In previous step, we've added routes (edges) within the same floor, for instance, a route between floor 0 and floor 1 doesn't exist in database yet, in this step we will add routes for all floors in the same table.

The idea returns from connecting floors between nodes of stairways and elevators that is equivalent to other stairways and elevators on the other floors, and creating routes (edges) between them.

Unlike the north building, stairways in south building is not named in a uniform where we can easily indicate and connect between pair stairways in different floors, so pairing them based on a name or a door is not possible.

Anyway, pairing stairways in south building based on distance is a logical solution, since we have a pre-measured coordinates it is logical that same stairways must not be far more than 1.5 meter, and same elevators not far more than 1 meter, this ensures using the same elevator and stairway to take the shortest route desired upon user request when using our web interface.

The following insert query is used for this purpose:

Insert into routable_del (select (row_number() over ())+100000 as id,n1.gid as source,n2.gid as target, 10 as cost from nodes_all_del n1, nodes_all_del n2 where n1.type='stairway' and st_distance(n1.the_geom,n2.the_geom)<1.5 and n2.level=n1.level+1)

Insert into routable_del (select (row_number() over ())+200000 as id,n1.gid as source,n2.gid as target, 20 as cost from nodes_all_del n1, nodes_all_del n2 where n1.type='elevator' and st_distance(n1.the_geom,n2.the_geom)<1 and n2.level>n1.level)

This is insert query, no columns (fields) are modified, but just rows are added to the same structure table, the (*routable_del*) table is enlarged by new routing records that connect all floors together. For example, drawing a route from door 0.413 that exits beside a stairway to a door in another floor requires drawing a route that goes through this stairway.

Only records in field "Type" filled as "*elevator*" or "*stairway*" considered to include a route that connects between various floors in the building.

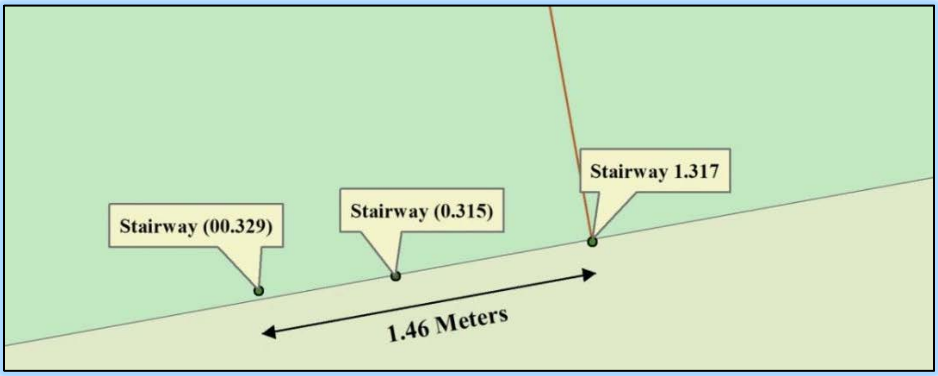

Figure 15: Doors (00.329, 0.315, 1.317) are beside the same stairway

Web Development (Web-Based Route Planner):

Routing Website:

First let’s look for the routing website address:

http://terkeptar.elte.hu/~campusrouting/south/

The file contents of the website stored in folder named (south), regarding to the south building where the project is located.

The web site itself is running in HTML code and the file running name is “*index.html*”, but the scripting language and functions are stored in a JavaScript file named “*south.js*”, all is inside folder “*south*”.

What is HTML and JavaScript language?

It is important to make a look and explain these languages we are typing in, Hypertext markup language (HTML) is a computer language its role is to make web pages, JavaScript has the same role but more advanced, but both of these languages integrates their efforts to get better web pages, for example, an HTML code can call a JavaScript code by referring to a JavaScript file by using HTML <Script> tag.

Later we will illustrate all contents of the web site, and describe the role, language written in for a file if needed and exists, however we have more than these files but it is simpler to explain each file when we look at the web-site viewers, contents and connections whenever it is needed.

Web-Site Viewers:

- OpenLayers: it is the layers on map on the back of the web-site, they are grouped like a panel that consists of a list, we can see the planned route after we input desired points, and the current point, and the building plan, it is the shapefile that we imported in order to view it over the map to see the route clearly, however it is not compulsory to use existing shapefile, we can use any other shapefile for the building in order to enhance the view.
- A panel for entering room numbers: on the bottom-left side of the web-site we can see a panel that consists of two input textboxes with their labels “from” and “to”, in these textboxes we should input the beginning point or room (door) of the route then the end of the route room (door), and also we can see a button labeled “go” which receives the order when clicking on mouse and calls the function that draws the route.
- A panel showing directions: on the right side of the page, there is a panel to show directions that will be written, before we input data it will be empty, when we input data we will see directions written literally and is harmonic with the route, each direction is separated in a new line, when we click on each direction it shows us the point also on the map that is related to the direction written, these written directions are associated with the position of each node in the route, it gets them from the database, anyway we will see later in this chapter how we will call the data in database

and also the task that is done to translate directions from the Hungarian language to the English version language.

Open-Layers Map:

There are four layers on the OpenLayers Map:

- The OpenStreetMap layer (it uses the map titles, icons, streets, names and other data of OpenStreetMap).
- The building outline layer (it is a vector layer, fed by a GeoJSON file),
  What is GeoJSON?
  It is a type of formatting to encode different geographic structures, and it supports points, lines and polygons, the name of GeoJSON file we will use is (building.geojson). (GEOJSON)
- The layer for the route plan, when giving the order for drawing route and drawing the graph the layer is separated in a layer, we can also make it visible or not.
- The layer when we move the mouse above one of the directions instructions and the node belonging to that instruction is displayed on this layer.

Route planning:

Managing PostgreSQL databases by using PHP frontend is the process that phpPgAdmin prepared to do. (Aarchlinux, 2018)

Consequently, a request is send from our web-site when executing to a PHP script file which name is (route_south.php).

The PHP script file steps to create a route will be as following:

- It gets the “from” and “to” information which is the “id” or door (room) from the user input.
- It creates a connection to access our database “gis”, if it couldn’t connect a message “Could not connect” appears.
- If the connection succeeded, it gets the “gid” field value of the “from” point and “to” point, which is the starting and ending point of the route, the “gid” is the primary key, hence it is the field required to call.
- Then it executes a route planning query between these points. And this query results list of nodes containing the following fields:
    1) X, Y coordinates for both points, the reference of these points is got from the field “the_geom” in the table “nodes_all_del” where it includes the coordinates that appears to us as a binary code.
    2) Level (it is the floor number).
    3) Type indicating type of the node, door, elevator, etc.
    4) ID, it includes the door number.
    5) Name, description written on the label of the door to show the room name.

For extra explanation look at figure (11) where the table “nodes_all_del” describes the values of those fields and shows many examples.

Routing Web-site Owner & Modifications:

It is necessary to indicate the first owners and publishers of the route planner web application, because I really adopted it from the same web application done for the north building which were done in participants between *Dr. Gede Mátyás* and *Eszényi Krisztián*, the last who is working on a similar project for north building.

The modifications I have done are listed as following:

- The PHP script uses same database but I have my own tables, views, sequences and my own structure.
- The viewer shows the ELTE southern building because we use different centering views, which is specified in file "south.js".
- I have translated the directions from Hungarian language to an English version, while the Hungarian version still remains available too.

Translations could be explained in the following table:

Original expression in Hungarian	Translated expression to English
Menj	Go
Menj előre	Go Ahead
métert	meter
Fordíts hátat az indulási pontnak	Turn your back to the starting point
fel a	up to
le a	down to
emeletre	upstairs
fel	up
fel az	up to
a földszintre	to downstairs
Hiba! Nincs ilyen objektum az adatbázisban	Fault! There is no such object in the database
Menj a lépcsőhöz	Go to the stairs
Menj a lifthez	Go to the elevator
Odaértél szerintem	I think you got it
Fordulj vissza	Turn back
Fordulj enyhén balra	Turn slightly to the left
Fordulj enyhén jobbra	Turn slightly to the right
Fordulj balra	Turn left
Fordulj jobbra	Turn right

Table 1: Translations of expressions and directions

Summary:

An indoor positioning system (IPS) is a method to determine place of objects, day by day the importance of IPS is increasing especially in rural areas.

GPS is the most usable device in outdoor places but it’s not appropriate for inside places.

One of the most challenges of IPS is finding accurate locations and the errors of radio signals.

This thesis (which is part of a larger project on routing within ELTE buildings) discusses several possible ways of indoor positioning, and introducing a new one: determining location based on the unique door numbers of a building.

This project covers the creation of an indoor route planner to south building of ELTE University. The project started with a practical mission: surveying all the doors of the building on the lower three levels. (The remaining levels are planned to be processed in the future.)

Then, the data of the survey was completed into a routable graph of nodes (including doors) and edges (possible ways between nodes), realized in ArcGIS and stored in PostGreSQL. The graph is realized as a table of edges in which every single record represents a connection between two nodes. The thesis discusses the creation steps of this graph.

Route planning is realized using the PGRouting extension of PostGres which can find the shortest (or lowest cost in any sense) route between two nodes of the route graph.

The web-based interface (developed by other participants of the project for the northern building) was modified and customized for the current dataset and the southern building. The navigation instructions were translated from Hungarian to English.

References

(dátum nélk.). Forrás: WhataFuture.com: http://www.whatafuture.com/google-maps-navigate-inside-building/

The future of indoor positioning. (2016. February 4). Letöltés dátuma: 2018. 04 27, forrás: Future Lab: https://futurelab.assaabloy.com/en/the-future-of-indoor-positioning/

Aarchlinux. (2018. March 18). *phpPgAdmin*. Forrás: Aarchlinux: https://wiki.archlinux.org/index.php/PhpPgAdmin

COPPELIS. (dátum nélk.). *Indoor Navigation, Indoor Location & Beacons Technology*. Forrás: COPPELIS Digital Innovation: http://coppelis.com/indoor-navigation/

Creative Commons. (2013. 04 15). *phpPgAdmin*. Forrás: phpPgAdmin: http://phppgadmin.sourceforge.net/doku.php

DEMPSEY, C. (2013. January 18). Letöltés dátuma: 2018. April 15, forrás: gislounge.com: https://www.gislounge.com/attribute-data-types-gis/

ESRI. (2016). Letöltés dátuma: 2018. April 14, forrás: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/creating-new-features/creating-a-segment-using-an-angle-and-a-length.htm

ESRI. (2016). Forrás: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/creating-new-features/creating-a-segment-perpendicular-to-another-segment.htm

ESRI. (2016). *An overview of the Features toolset*. Forrás: ESRI: http://desktop.arcgis.com/en/arcmap/10.4/tools/data-management-toolbox/an-overview-of-the-features-toolset.htm

ESRI. (2016). *COGO Tools.* Letöltés dátuma: 2018. April 14, forrás: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/editing-parcels/parallel-perpendicular-and-deflection.htm

ESRI. (2016). *Using Snapping*. Letöltés dátuma: 2018. April 12, forrás: ArcGIS: http://desktop.arcgis.com/en/arcmap/10.4/manage-data/editing-fundamentals/about-snapping.htm

ESRI. (2017). *Turn snapping on or off*. Letöltés dátuma: 2018. April 12, forrás: www.arcgis.com: http://pro.arcgis.com/en/pro-app/help/editing/enable-snapping.htm

ESRI. (2018). *ArcGIS field data types*. Letöltés dátuma: 2018. April 12, forrás: www.arcgis.com: http://desktop.arcgis.com/en/arcmap/10.4/manage-data/geodatabases/arcgis-field-data-types.htm

ESRI. (2018). *DBMS data types supported in ArcGIS*. Letöltés dátuma: 2018, forrás: arcgis.com: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/databases/dbms-data-types-supported.htm

ESRI. (2018). *Feature To Line*. Forrás: ESRI: http://pro.arcgis.com/en/pro-app/tool-reference/data-management/feature-to-line.htm

Farina, N. (2011. 10 11). *Why indoor navigation is so hard*. Forrás: O'Reilly: http://radar.oreilly.com/2011/10/indoor-navigation.html

Gartner, H. H. (2010). *A Survey of Mobile Indoor Navigation Systems.* Institute of Geoinformation and Cartography, Vienna University of Technology, Austria: Haosheng Huang and Georg Gartner.

GEOJSON. (dátum nélk.). *GEOJSON*. Letöltés dátuma: 2018. May 10, forrás: GEOJSON: http://geojson.org/

Holcik, M. (2012). Master's Thesis. *Indoor Navigation for Android*. Brno, Masaryk University - Faculty of Informatics, Czech Republic: Michal Holcik.

LinkedIn Corporation. (2017. May 25). *The Importance of Indoor Navigation Systems*. Forrás: https://www.slideshare.net/indoornavigationsystems/the-importance-of-indoor-navigation-systems

Mann, K. (2006. April - June). *More ArcMap Editing Tips and Tricks.* Forrás: ESRI Industry Solutions on the Government.
Schmidt, A. (dátum nélk.). *Advanced ArcMap Editing.* Letöltés dátuma: 2018. April 13, forrás: United Services Group: http://www.mweuug.com/presentations/2006/Advanced%20ArcMap%20Editing.pdf
SenionLab. (dátum nélk.). *Senion*. Forrás: SenionLab: https://senion.com/indoor-positioning-system/
SPREO development team. (2014. 7 15). *Indoor Navigation Challenges Series*. Forrás: SPREO: http://spreo.co/indoor-navigation-challenges-series-chapter-1-finding-an-accurate-location-indoors/
W3Schools. (dátum nélk.). *SQL CREATE INDEX Statement*. Letöltés dátuma: 2018. May 5, forrás: W3Schools: https://www.w3schools.com/sql/sql_create_index.asp
W3Schools. (dátum nélk.). *SQL UNION Operator*. Letöltés dátuma: 2018. May 5, forrás: W3Schools: https://www.w3schools.com/sql/sql_union.asp
Wikipedia. (2018. 03 15). *Indoor positioning system*. Forrás: Wikipedia: https://en.wikipedia.org/wiki/Indoor_positioning_system
Young, J. (2015. November). *ArcWatch: GIS News, Views, and Insights*. Letöltés dátuma: 2018. 04 21, forrás: Mapping Indoor Building Space is a Smart Business Move: http://www.esri.com/esri-news/arcwatch/1115/mapping-indoor-building-space-is-a-smart-business-move