Czech University of Life Science Prague

Faculty of Economics and Management

Diploma Thesis

Use of QR Codes in Mobile Software Platform

Author: **Martin Říha**

Supervisor: **assoc. prof. Vojtěch Merunka, Ph.D.**

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE
Department of Information Engineering
Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Říha Martin

Informatics

Thesis title
Use of QR codes in mobile software platform

Objectives of thesis
Usage of QR codes for navigation in the CZU campus and in the PEF building and help people to get to desired destination.

Methodology
Asigning QR to code to every building and room in the PEF faculty. After that user can scan in by barcode scanner in his mobile device (or directly access the webpages from browser), set destination and see the desired destination and path leading there.

Schedule for processing
Q1,Q2 2012 - Studying of Shortest path algorithms
Q3,Q4 2012 - JavaScript, JQuery preparations
Q1 2013 - Working on practical part

The proposed extent of the thesis

60-80 pages

Keywords

qr code, shortest path algorithm, javascript, html, css

Recommended information sources

MCFARLAND, David Sawyer. JavaScript. 2n ed. Sebastopol, Calif.: O'Reillly, 2011c2012, xvi, 518 p.. ISBN 14-493-9902-9.

FLANAGAN, David. JQuery pocket reference. 1. ed. Beijing: O'Reilly, 2010, xvi, 518 p. ISBN 14-493-9722-0.

CORMEN, Thomas H and David S JOHNSON. Introduction to algorithms: ninth DIMACS implementation challenge. 3rd ed. Cambridge: MIT Press, c2009, xix, 1292 s. DIMACS series in discrete mathematics and theoretical computer science, v. 74. ISBN 978-0-262-03384-8.

The Diploma Thesis Supervisor

Merunka Vojtěch, doc. Ing., Ph.D.

Last date for the submission

March 2013

Ing. Martin Pelikán, Ph.D.
Head of the Department

prof. Ing. Jan Hron, DrSc., dr.h.c.
Dean

Prague March 19. 2013

Declaration

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Prague, March 2013 _______________________________

Acknowledgement

I would like to use this opportunity to sincerely thank assoc. prof. Vojtěch Merunka, Ph.D. for his assistance in guiding me throughout this research and for providing me with the facilities that made this work possible.

Use of QR Codes in Mobile Software Platform

Summary

This diploma thesis deals with usage of QR Codes in Mobile Software Platform. Currently at CULS new students and visitors are usually wondering how they can get to certain campus location. The aim of this thesis is to develope a user-friendly web application to help users get to the desired destinations in time and by using the shortest possible route. Thanks to the fact that almost every smartphone has a built-in camera, a barcode scanner and a web browser installed, the author develops a web application based on barcode scanning. After a user scans a barcode it redirects his browser to his current location on a on-line map where he can type in the required destination and see the shortest route to it.

In the first part of the thesis the reader is introduced to important theoretical background about barcodes, graph theory, shortest path problem and used web technologies to solve the task. In the second part the author describes step by step the process of development of the application, the decisions he makes and explains the steps leading to his final solution.

Key words: barcode, QR code, graph, vertex, edge, node, route, breadth-depth search algorithm, JavaScript, associative array, recursion, function

Souhrn

Tato diplomová práce se zabývá využitím QR kódů na mobilní softwarové platformě. Současná situace v areálu ČZU je taková, že každý nový student nebo návštěvník přemýšlí, kde je místnost, kam má namířeno. Cílem této práce je vyvinutí uživatelsky přívětivé webové aplikace, která uživatelům pomůže dostat se do požadovaných destinací v čas a s využitím co nejkratší možné cesty. Díky skutečnosti, že téměř každý smartphone má zabudovaný fotografický přístoroj, nainstalovánu čtečku čárových kódů a webový prohlížeč, autor vyvíjí aplikaci založenu na skenování čárových kódů. Poté, co uživatel naskenuje čárový kód, url jej přesměruje na konkrétní lokaci na on-line mapě, kde si poté může zadat požadovanou destinaci a vidět nejkratší cestu k ní.

V první části práce je čtenáři představeno teoretické zázemí o čárových kodech, teorii grafů, hledání nejkratší cesty a využitých webových technologiích k vyřešení úlohy. V druhé části autor popisuje vývoj aplikace krok za krokem, rozhodnutí, která činí a vysvětluje kroky vedoucí k finálnímu řešení.

Klíčová slova: čárový kód, QR kód, graf, vrchol, hrana, uzel, cesta, prohledávání do šířky, JavaScript, asociativní pole, rekurze, funkce

Table of Contents

1. Introduction

Every visitor or a new student is not familiar with the room numbering used at the CULS campus. On one hand the first letter of most rooms is common for all rooms in a current building. On the other hand how can the user know in which building the room names start with the letter A. In some faculties , for example in the PEF faculty, some rooms start with the letter E and some rooms start with the letter C. Also some buildings on the campus do not respect the naming rule. For example buildings AFA, AFB, AFC. If the user has AFA in his timetable he could believe that the room is located in the building FAPPZ where all rooms start with letter A. The next problem is that the map of the faculty booklets does not contain a map of the campus. In the CULS campus a navigation system called ORIS is offered, which is located at www.oris.czu.cz website, but it's representation is not familiar with new age mobile phones and web browsers. The problem is that this system is not made having in mind modern mobile devices.

To facilitate orientation on the campus there is a need of a solution accessible to everyone. This solution has to cover all the visitor's and student's needs. To find a way from his current position to a desired destination and to show any location on the map. For these purposes the author develops a web-based application accessible by almost every modern smartphone's web browser. To identify the user's position the author uses QR codes, because not every smartphone has a GPS module.

2. Aim of work and Methodology

The aim of this diploma thesis is to develop user friendly navigation application which can run on as much smartphone devices as possible so that it helps users not to get lost in the CULS campus.

The first part describes barcode types and formats that are widely used. Next for every barcode is stated for which purposes it is used and which data it can store.Also In this part 1D and 2D barcodes are compared. After that it depicts detailed properties of QR codes. The next part is about Graph theory and shortest path problem. There will be a description of graph-theoretic data structures, because it's very important to decide how to represent data in the program. The shortest path problem section helps to decide which algorithm will be used in the final application by analyzing mainly used search graph search algorithms. Web technologies introduce the Document Object Model and other common web technologies like scripting languages, HTML, CSS and server side configuration.

In the practical part the process of development is described step by step where the author decides which solution and why he uses. The biggest part of this chapter is a description concerning the source code of the programmed application.

In the final chapter author depicts the pros and cons and summaries gained knowledge by a process of development of his application.

I. Literature Review

3. Background Analysis

3.1. Barcodes

3.1.1. Definition

Data which are related to an object can be stored in a barcode. In the beginning there were only linear and one-dimensional barcodes. These types of barcodes are known as 1D and data representation is done systematically by varying the widths and spacings of parallel lines. At these times barcodes are known as rectangles, dots, hexagons and other geometric patterns in two dimensions. In the beginning for reading data were used specialized barcode readers while today barcode readers are available in almost every smart device including smartphones, tablets and printers (IPS, 2013).

Barcode Type	**Information density**	**Information capacity**	**Information type**	**Error correction function**	**Dependence on database**	**Nature**
1D	Low	Small	Numbers, English	Yes	Must depend on database or communication network	Object label & index
2D	High	Big	Numbers, English, Chinese, pictures, voice and other binary information	No	No depend on database or communication network	Description on objects

Table 1 - Comparison between 2D and 1D barcode (SYSCAN, 2010)

3.1.2. 1D Barcodes

A single dimension barcode is a set of parallel lines, for best performance, on a contrasting background. Common symbologies for 1D barcodes include Code 128, Code 39, EAN, UPC and Interleaved 2 of 5.

Type	About
CODABAR A123456789B	Originally developed by Pitney Bowes, the Codabar Barcode is used primarily by US Blood Banks, Photo Labs and Overnight Delivery Services. Codabar can encode the numbers 0 through 9, the start/stop characters A, B, C, D, E, *, N or T. and the six symbols (-:.$/+).
CODE 11 Barcode 	Code 11 is a barcode symbology developed by Intermec in 1977. It is used primarily in telecommunications. The symbol can encode any length string consisting of the digits 0-9 and the dash character (-). One or more modulo-11 check digit(s) can be included.
CODE 128 Barcode NATIONWIDE BARCODE	Code 128 is a very high-density barcode symbology. (A special version of it called GS1-128 is used extensively world wide in shipping and packaging industries.) It is used for alphanumeric or numeric-only barcodes. It can encode all 128 characters of ASCII and, by use of an extension character (FNC4), the Latin-1 characters defined in ISO/IEC 8859-1.
CODE 2 OF 5 – Interleaved 1234567895	Interleaved 2 of 5 is a continuous two-width barcode symbology encoding digits. It is used commercially on 135 film and on cartons of some products, while the products inside are labeled with UPC or EAN.
CODE 39 Barcode 	Code 39 (also known as Alpha39, Code 3 of 9, Code 3/9, Type 39, USS Code 39, or USD-3) is a variable length, discrete barcode symbology. The Code 39 specification defines 43 characters, consisting of uppercase letters (A through Z), numeric digits (0 through 9) and a number of special characters (-, ., $, /, +, %, and space). An additional character (denoted '*') is used for both start and stop delimiters. One advantage of Code 39 is that since there is no need to generate a check digit, it can easily be integrated into existing printing system by adding a barcode font to the system or printer and then printing the raw data in that font.

Table 2 - 1D barcode formats CODE (*NATIONWIDEBARCODE*, 2010)

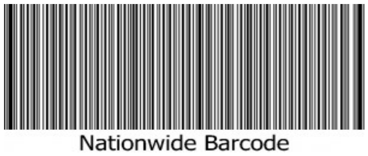
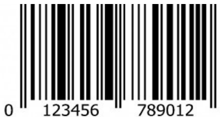
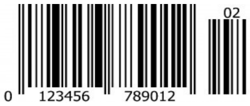
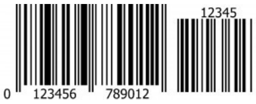
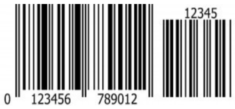
Code 93 Barcode 	Code 93 is a barcode symbology designed in 1982 by Intermec to provide a higher density and data security enhancement to Code 39. It is an alphanumeric, variable length symbology. Code 93 is used primarily by Canada Post to encode supplementary delivery information. Every symbol includes two check characters. Code 93 is designed to encode 26 upper case letters, 10 digits and 7 special characters: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 -, ., $, /, +, %, SPACE. Each Code 93 character is divided into nine modules and always has three bars and three spaces, thus the name. Each bar and space is from 1 to 4 modules wide. In addition to 43 characters, Code 93 defines 5 special characters (including a start/stop character), which can be combined with other characters to unambiguously represent all 128 ASCII characters.
EAN 13 Barcode 	An EAN-13 barcode (originally European Article Number, but now renamed International Article Number) is a 13 digit (12 data and 1 check-digit) barcoding standard which is a superset of the original 12-digit Universal Product Code (UPC) system developed in the United States. The EAN-13 barcode is defined by the standards organization GS1.
EAN+ 2 Digit Barcode 	This is a variant of the EAN barcode. This can be used for magazines where the last two numbers represent the volume or month of issue.
EAN 13 + 5 Digit Barcode 	This is a variant of the EAN barcode. Often used for Greeting Cards and other items when the manufacturer or distributor is responsible for inventory tracking
EAN 8 Barcode 	An EAN-8 is a barcode and is derived from the longer European Article Number (EAN-13) code. It was introduced for use on small packages where an EAN-13 barcode would be too large; for example on cigarettes, pencils (though it is rarely used for pencils), and chewing gum packets. EAN-8 barcodes may be used to encode GTIN-8s which are another set of product identifiers from the GS1 System. It begins with a 2- or 3-digit GS1 prefix (which is assigned to each national GS1 authority) 5- or 4-digit item reference element depending on the length of the GS1 prefix), and a checksum digit.

Table 3 - 1D barcode formats CODE and EAN (*NATIONWIDEBARCODE*, 2010)

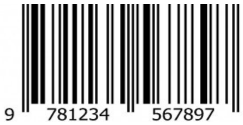
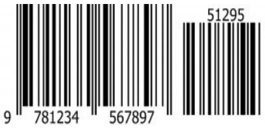
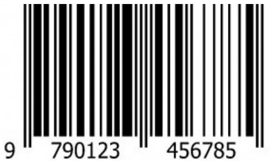
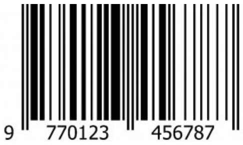
ISBN-13 	The International Standard Book Number (ISBN) is a unique numeric commercial book identifier based upon the 9-digit Standard Book Numbering (SBN) code created by Gordon Foster, Emeritus Professor of Statistics at Trinity College, Dublin,for the booksellers and stationers W.H. Smith and others in 1966. An ISBN barcode uses an EAN symbology.
ISBN with 5 digit Add On 	Most ISBN graphics also include a 5 digit add on. The first barcode is the ISBN barcode (using an EAN symbology) and the second barcode to the right indicates the price of the book. The first number, 5, represents a dollar sign ($). The next four numbers are the price of the book. In this exanple 51295 means that the book sells for $12.95. Since there are only 4 numbers after the 5, books that sell for more than $99.99 will be represented by 5999.
ISMN Barcode 	The International Standard Music Number (ISMN) is a unique number for the identification of all notated music publications from all over the world, whether available for sale, hire or gratis–whether a part, a score, or an element in a multi-media kit. The ISMN is designed to rationalize the processing and handling of notated music and the respective bibliographical data for publishing houses, the music trade and libraries. As of 1 January 2008 the ISMN consists of 13 digits starting with 979-0 Existing 10-digit ISMNs are prefixed by 979- The leading M- of the 10-digit ISMNs will be replaced by 0- (zero) The resulting 13-digit number will be identical with the EAN-13 number that is currently encoded in the bar code ISO Standard 10957 gives the basic rules of the ISMN system. The thirteen-digit number allows a billion items each to carry a different number.
ISSN Barcode 	Used mainly throughout Europe, The ISSN (International Standard Serial Number) identifies periodical publications as such, including electronic serials. The ISSN takes the form of the acronym ISSN followed by two groups of four digits, separated by a hyphen. The eighth character is a control digit calculated according to a modulo 11 algorithm on the basis of the 7 preceding digits; this eighth control character may be an "X" if the result of the computing is equal to "10″, in order to avoid any ambiguity. The first 3 digits

Table 4 - 1D barcode formats IS (*NATIONWIDEBARCODE*, 2010)

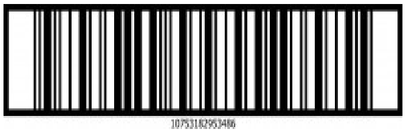
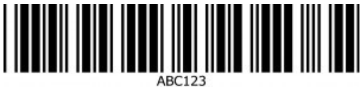
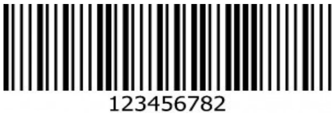
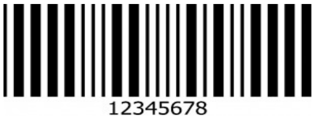
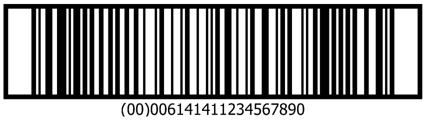
ITF 14 (GTIN 14)  	The ITF14 is a 14 digit barcode used to mark the master shipping containers of products with a UPC identifier. It is based on the I2of5 barcode. ITF14 barcodes usually contain a top and bottom bar (sometimes rectangle) called the Bearers bar. These bars make sure that the barcode is read completely. The first number is typically an arbitrary number between 0 and 7, the next number is a zero, the following 11 numbers are the first 11 numbers on a UPC barcode and the last number is a check digit.
LOGMARS Barcode  	LOGMARS (Logistics Applications of Automated Marking and Reading Symbols) is a special application of Code 39 used by the U.S. Department of Defense and is governed by Military Standard MIL-STD-1189B.
MSI Barcodes  	MSI (also known as Modified Plessey) is a barcode symbology developed by the MSI Data Corporation, based on the original Plessey Code symbology. It is a continuous symbology that is not self-checking. MSI is used primarily for inventory control, marking storage containers and shelves in warehouse environments.
Pharmacode Barcode  	Pharmacode, also known as Pharmaceutical Binary Code, is a barcode standard, used in the pharmaceutical industry as a packing control system. It is designed to be readable despite printing errors. It can be printed in multiple colors as a check to ensure that the remainder of the packaging (which the pharmaceutical company must print to protect itself from legal liability) is correctly printed.
SSCC18 Barcode also known as UCC 128  	This Symbology is also known as UPC-128 Shipping Container Code, Code 128 UPC Shipping Container Code, Serial Shipping Container Code, UCC-128, EAN-18, NVE (Nummer der Versandeinheit) This is a special version of UCC/EAN-128, with Application Identifier (AI) = 00. It is used to identify shipping containers by a serial number. SSCC18 Barcodes can be configured 3 different ways, no bearer bars, bearer bars top and bottom or bearer bars on all 4 sides.

Table 5 - 1D barcode formats OTHER (*NATIONWIDEBARCODE*, 2010)

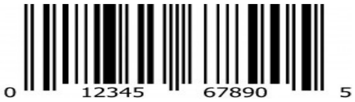
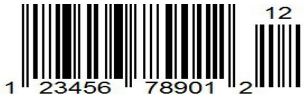
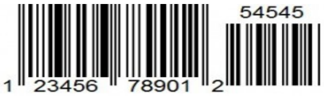
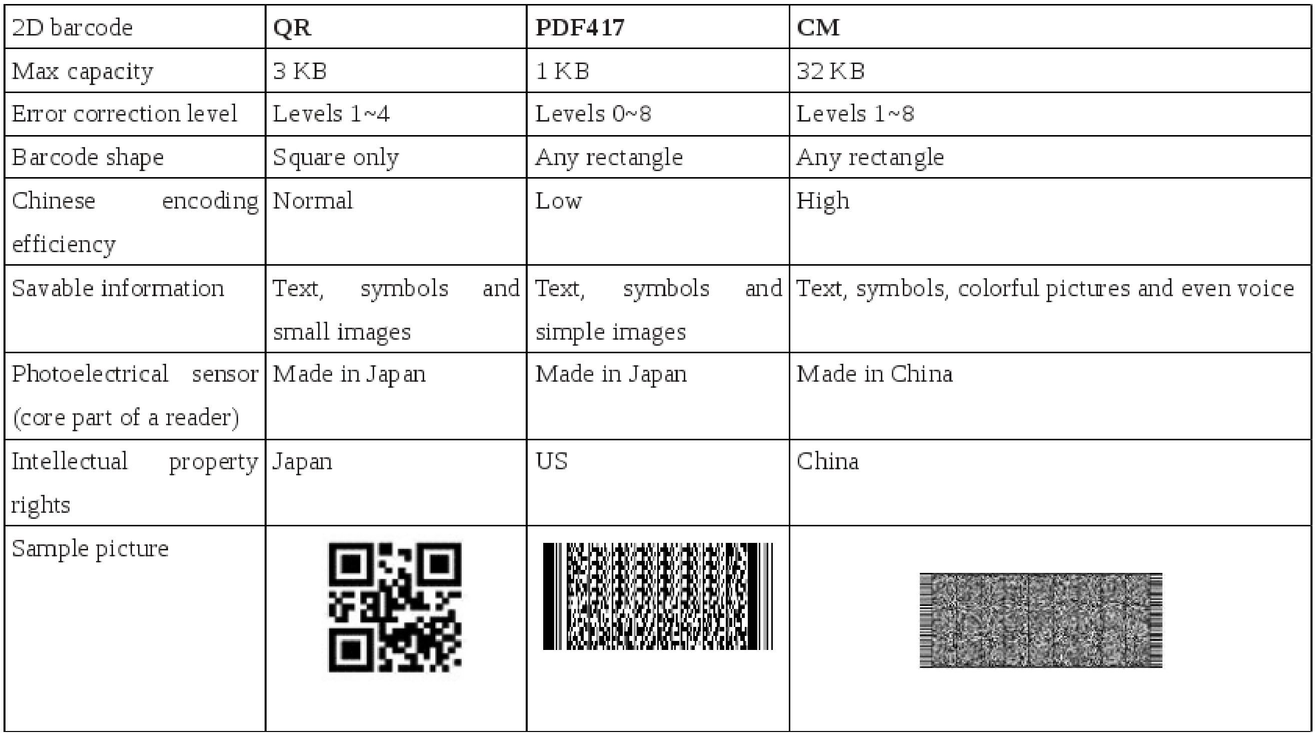
UPC A (UPC 12)  	UPC-A Barcodes are used for marking products sold in stores throughout the USA and Canada. The barcode is comprised of three groupongs of numbers, the manufacturer ID (the company who owns the barcode prefix) which is 6, 7, 8 or 9 digits in length, the balance of numbers totalling 11 digits and the final 12th number (Check Digit or Checksum).
UPC A with 2 Digit Extension  	Often used for Magazines – the second barcode indicates month or edition.
UPC-A Barcode with 5 Digit Extension  	Often used for Greeting Cards where the publisher takes responsibility for inventory management and stocking of product.

Table 6 - 1D barcode formats UPC (*NATIONWIDEBARCODE*, 2010)

3.1.3. 2D Barcodes

A 2D barcode is mainly represented as a square form of black and white squares where are stored data. As is mentioned in comparison between 1D and 2D barcodes, 2D barcodes offer high information density, big information capacity and are not dependent on database or communication network. Main used types of 2D barcodes are Compact Matrix Code (CM), Grid Matrix Code (GM), Portable Data File (PDF417), Data Matrix Code (DM) and Quick Response Code (QR). The comparison between these codes is in the table below.



2D barcode	**QR**	**PDF417**	**CM**
Max capacity	3 KB	1 KB	32 KB
Error correction level	Levels 1~4	Levels 0~8	Levels 1~8
Barcode shape	Square only	Any rectangle	Any rectangle
Chinese encoding efficiency	Normal	Low	High
Savable information	Text, symbols and small images	Text, symbols and simple images	Text, symbols, colorful pictures and even voice
Photoelectrical sensor (core part of a reader)	Made in Japan	Made in Japan	Made in China
Intellectual property rights	Japan	US	China
Sample picture			

Table 7 - Comparison between CM code and PDF417 & QR (SYSCAN, 2010)

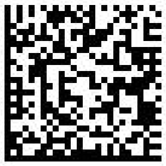
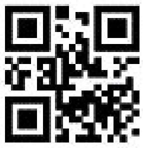
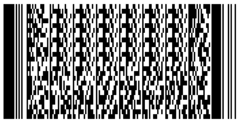
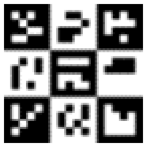
2D barcode	**Data Matrix**	**QR**	**PDF417**	**GM**
Layout	Matrix	Matrix	Layers (1D layout)	Matrix
Readable direction	Full direction	Full direction	Upward/downward	360o full direction
Image “dead point”	Yes (no tarnishing for positioning image)	Yes (no tarnishing for positioning image)	Yes (no tarnishing for positioning image)	No
Chinese encoding efficiency	Bad (16bit)	Bad (16bit)	Bad (16bit)	Good (13bit)
Photoelectrical sensor (core part of a reader)	Made in Japan or US	Made in Japan	Made in Japan or US	Made in China
Intellectual property rights	US	Japan	US	China
Sample picture				

Table 8 - Comparison between GM code and Data Matrix, PDF417 & QR (SYSCAN, 2010)

3.1.4. QR Codes

3.1.4.1. Properties

The conventional bar codes capability is a maximum of 20 digits. QR Code is capable of handling all types of data, not only digits. It can handle from numeric and alphabetic characters up to Kanji, Kana and Hiragana symbols. QR Code data capabilities are listed int the table 1. It is established as an ISO (ISO/IEC18004) standard.

QR Code Data Capacity	
Numeric only	Max. 7,089 characters
Alphanumeric	Max. 4,296 characters
Binary (8 bits)	Max. 2,953 bytes
Kanji, full-width Kana	Max. 1,817

Table 9 - QR Code Data Capacities (QRCode.com, 2010a)

Small printout size

QR Code of the same size as a traditional bar code is capable of encoding ten times higher amount of data. It is achieved thanks to the both horizontal and vertical information carriage.

Kanji and Kana Capability

Because of symbology used in Japan, QR Code can carry encoded JIS Level 1 and Level 2 kanji character set.

Dirt and Damage Resistant

Thanks to the correction capability Data can be restored even if the symbol is partially dirty or damaged. A maximum 30% of codewords can be restored (QRCode.com, 2010a).

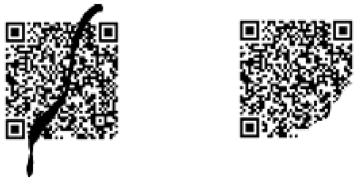


Image 1 - QR Code damage (QRCode.com, 2010a)

Readable from any direction in 360°

Capability of omni-directional and high speed reading is the next big advantage of QR Code. This is done through position detection patterns which are located at the three corners of the symbol and help to partially eliminate the negative effects of background interference (QRCode.com, 2010a).

Structured Append Feature

One data symbol can be reconstructed into up to 16 symbols and later reconstructed.

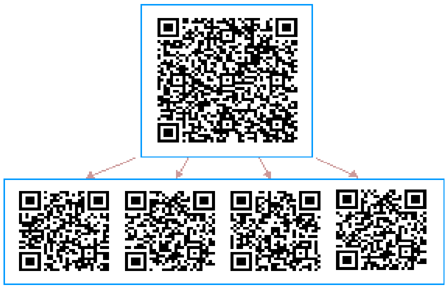


Image 2 - Structured QR Code (QR Code.com, 2010a)

A version number describes the density of the code. It is based on the number of columns of dots and number of rows. A Version 1 QR has 21 rows and 21 columns of dots and the version number increases by 1 for every 4 extra rows and columns. Most QR codes containing url end by being Version 2 QR codes – 25 columns of data dots and rows. The Magic number is the number which makes a QR code reliably scannable (QRStuff.com, 2013).

3.1.4.2. QR Code Size Decision Factor

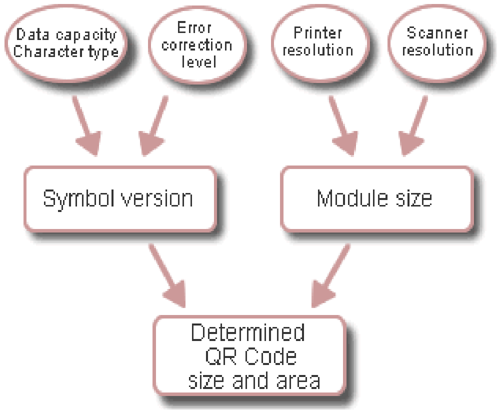
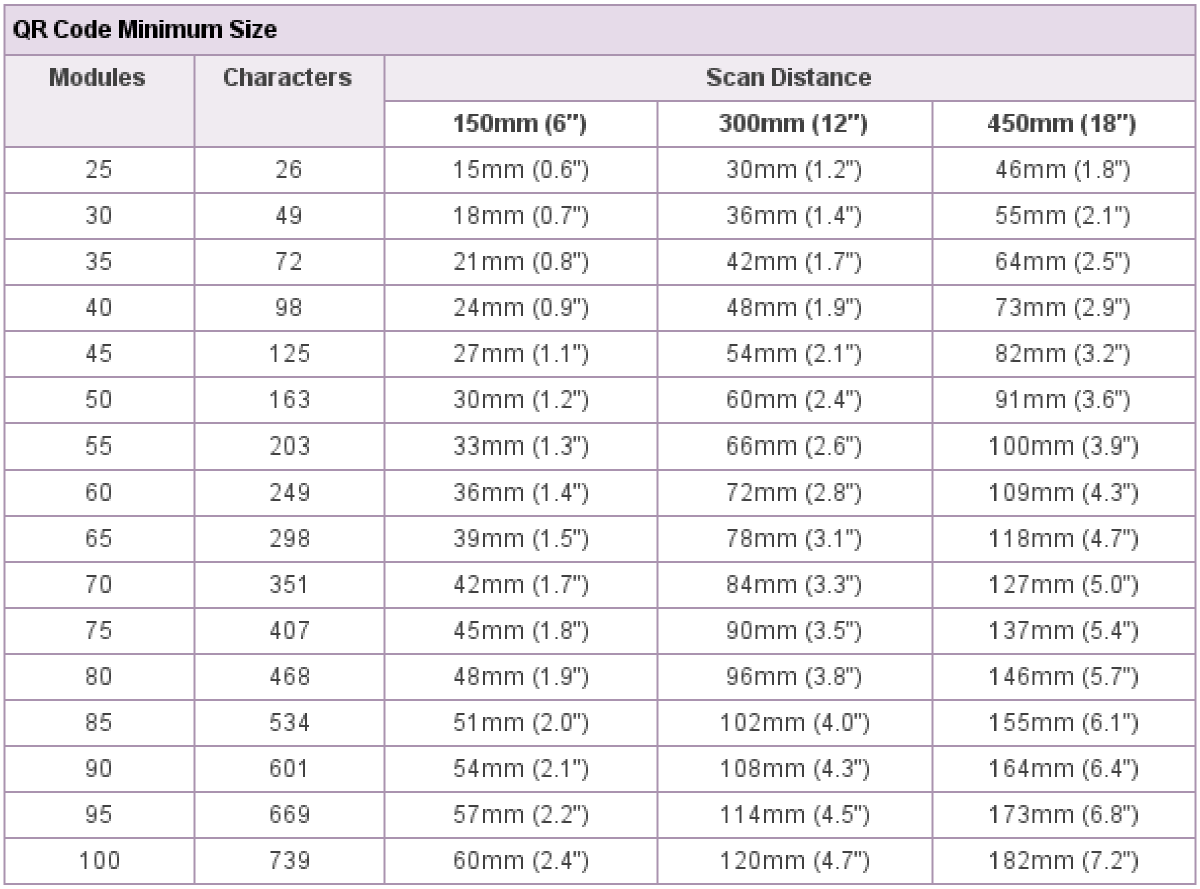


Image 3 - QR Code Size Decision Factor (QRCode.com, 2010b)

"The size of QR Code is decided by determining a symbol version, based on data capacity, character type and error correction level, and by setting a module size, based on the performance of the printer for printing or the scanner for reading" (QRCode.com, *2010b).*

A version number describes the density of the code. It is based on the number of columns of dots and number of rows. A Version 1 QR has 21 rows and 21 columns of dots and the version number increases by 1 for every 4 extra rows and columns. Most QR codes containing url end by being Version 2 QR codes – 25 columns of data dots and rows. The Magic number is the number which makes a QR code reliably scannable (QRStuff.com, 2013a).



QR Code Minimum Size				
Modules	Characters	Scan Distance		
		150mm (6")	300mm (12")	450mm (18")
25	26	15mm (0.6")	30mm (1.2")	46mm (1.8")
30	49	18mm (0.7")	36mm (1.4")	55mm (2.1")
35	72	21mm (0.8")	42mm (1.7")	64mm (2.5")
40	98	24mm (0.9")	48mm (1.9")	73mm (2.9")
45	125	27mm (1.1")	54mm (2.1")	82mm (3.2")
50	163	30mm (1.2")	60mm (2.4")	91mm (3.6")
55	203	33mm (1.3")	66mm (2.6")	100mm (3.9")
60	249	36mm (1.4")	72mm (2.8")	109mm (4.3")
65	298	39mm (1.5")	78mm (3.1")	118mm (4.7")
70	351	42mm (1.7")	84mm (3.3")	127mm (5.0")
75	407	45mm (1.8")	90mm (3.5")	137mm (5.4")
80	468	48mm (1.9")	96mm (3.8")	146mm (5.7")
85	534	51mm (2.0")	102mm (4.0")	155mm (6.1")
90	601	54mm (2.1")	108mm (4.3")	164mm (6.4")
95	669	57mm (2.2")	114mm (4.5")	173mm (6.8")
100	739	60mm (2.4")	120mm (4.7")	182mm (7.2")

Table 10 - QR Code Minimum Size (QRStuff.com, 2011b)

3.2. Graph theory

3.2.1. Definition

Study of pairwise relations between objects from a certain collection modelled by mathematical structures is called the Graph theory. A graph is represented by a collection of nodes or vertices which are connected by edges. When is one vertex directed to another the graph is directed. In case there is no distinction between the two vertices associated with each edge, the graph is undirected (WhatIs.com, 2005).

3.2.2. List structures

3.2.3. Incidence list

All the edges are stored in an array which contains pairs of nodes (in udirected graph). In directed graph edges contain tuples. Nodes which are connected by an edge are adjacent.

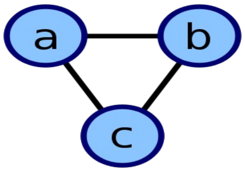
Vertice	**Connects**	
1	A,B	
2	A,C	
3	C,B	

Table 11 - Incidence List Structure

3.2.4. Adjacency list

To define adjacency list it is needed to say that it is a collection of unordered arrays representing each array for each vertex in the graph. Every array contains the set of neighbors of current node.

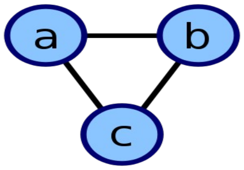
Vertex	**Is connected to**	
A	B,C	
B	A,C	
C	A,B	

Table 12 - Adjacency List Structure

3.3. Matrix structures

3.3.1. Incidence matrix

An incidence matrix shows the relationship between two classes of objects. In case that matrix has two elements – X and Y, one row of element X is for each element Y. If they are related, the entry in row x and column y is equal to 1. If not, the entry is equal to 0.

"In graph theory an undirected graph G has two kinds of incidence matrices: unoriented and oriented. The incidence matrix (or unoriented incidence matrix) of G is a p × q matrix (b_{ij}), where p and q are the numbers of vertices and edges respectively, such that b_{ij} = 1 if the vertex v_i and edge x_j are incident and 0 otherwise" (Wikipedia: the free encyclopedia, 2012a).

"The incidence matrix of a directed graph D is a p × q matrix [b$_{ij}$] where p and q are the number of vertices and edges respectively, such that b_{ij} = -1 if the edge x_j leaves vertex v_i, 1 if it enters vertex v_i and 0 otherwise (Note that many authors use the opposite sign convention.). Incidence matrix of a directed graph is a skew symmetric matrix" (Wikipedia: the free encyclopedia, 2012a).

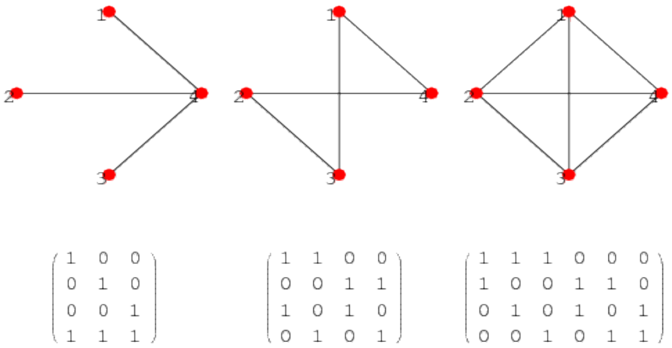
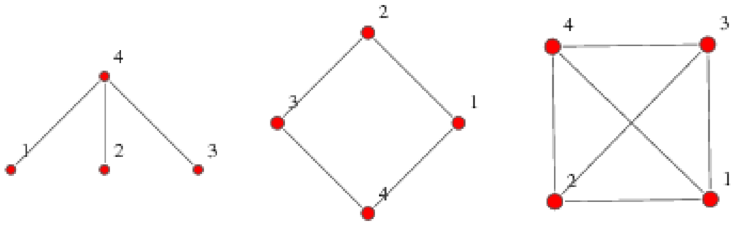


Image 4 - Incidence Matrix Structure (WolframMathWorld, 2012a)

3.3.2. Adjacency matrix

The connection matrix, as is sometimes called, has rows and colums labeled by graph vertices. The value is 0 or 1 (v_i,v_j) depending on whether v_i and v_j are adjacent or not. For example a simple graph without self-loops, the connection matrix has 0s on the diagonal. If a graph is undirected, the matrix is symmetric (WolframMathWorld, 2012).



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Image 5 - Adjacency Matrix Structure (WolframMathWorld, 2012b)

3.4. Search Algorithms

3.4.1. Definition

Search algorithms help to examine or visit each node exactly once in a tree structure in a travesal order where traversal order means to visit the vertices in some systematic order. It is a way to explore the structure and help to find the shortest path from the first node to destined node. In each step the algorithm compares the input key with an offered key from an array. If both keys match, algorithm terminates and outputs the path (University of California Irvine, 2004). There are known two types of tree traversal algorithms – Depth-first search algorithm and Breadth-first search algorithm.

3.4.2. Depth-first search

This algorithm starts it's search at the root and explores as far as possible along each branch before backtracking. The time to traverse an entire graph takse time O(|E|), linear in the size of the gaph. Space O(|V|) is in some cases used to store the stack of vertices on the current search path. Already visited vertices are stored in the same space but not in the same array. Some problems can occur when the searched graph is too large or infinite, so depth-first search may suffer from non-termination. In these cases the consumption of memory is very high. To solve this problem is in some cases set a limited depth which tells to algorithm in how many cycles to stop and return some value. Keeping already-visited data structures in memory is not in some cases used as well (Wikipedia: the free encyclopedia, 2013b). “*In this case, the time is linear in the number of expanded vertices and edges but the space complexity of this variant of DFS is only proportional to the depth limit, much smaller than the space needed for searching to the same depth using breadth-first search. For such applications, DFS also lends itself much better to heuristic methods of choosing a likely-looking branch”* (Wikipedia: the free encyclopedia, 2013b). If the depth limit is unknown depth-first search repeatedly searches with increasing number of depth limits. This algorithm can be used to collect samples of nodes too.

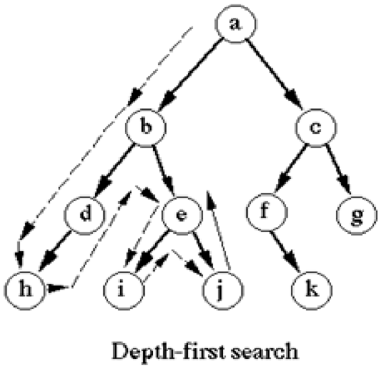


Image 6 - Depth-first Search Algorithm (Coding Lifestyle, 2011)

3.4.3. Breadth-first search

Breadth-first search algorithm is search algorithm which starts at the root node and explores all the neighboring nodes starting with the first. When it explores the first edge of the node, it starts exploring the next first edge of the node and so one. If it finds terminating node, algorithm step one node back and search the second possible edge of this node. Nodes that have not been examined are placed into some array where they stay until they are called. The problem of this algorithm is when it finds a destination, it can use more nodes to visit than the shortest path is and than the depth-first search algorithm finds (Princeton University, 2010).

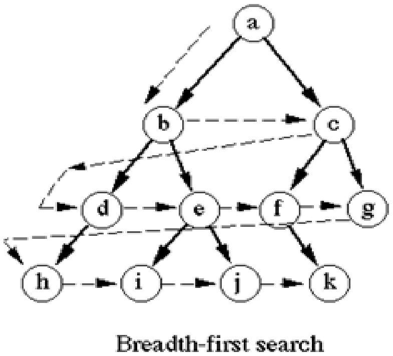
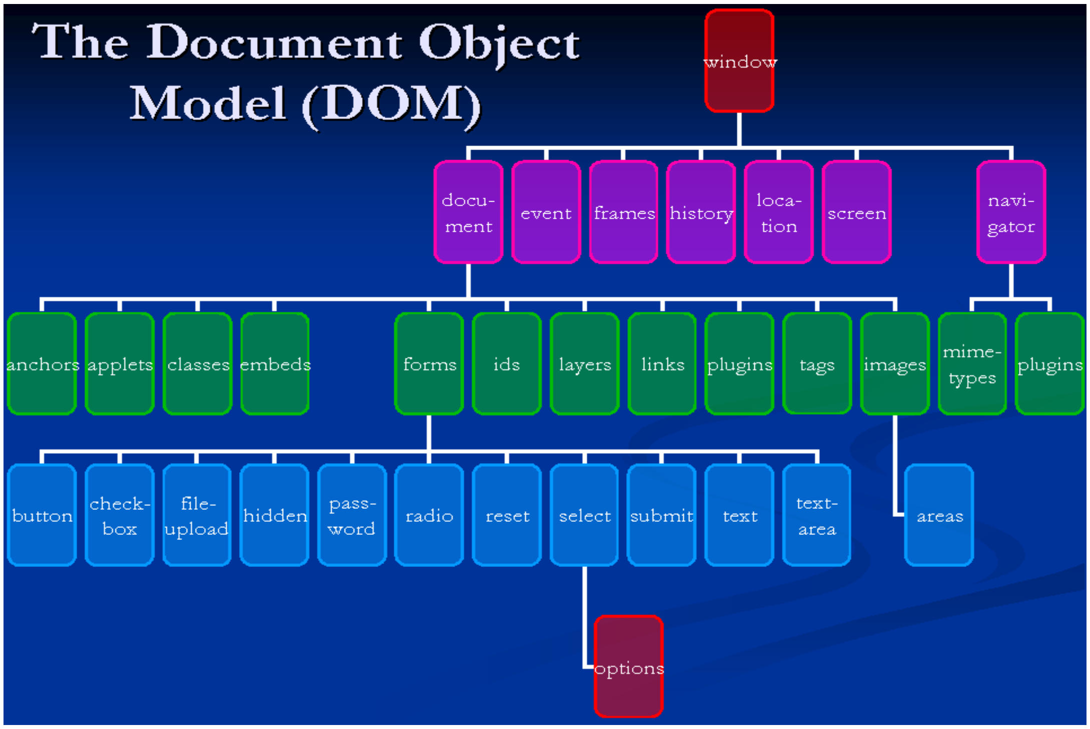


Image 7 - Breadth-first Search Algorithm **(**Coding Lifestyle, 2011)

3.5. Web Technologies

3.5.1. Document Object Model

The Document Object Model is interface which allows dynamical access by scripts and programs to update the content, structure and style of documents. It is platform and language-neutral interface. By processing the document the result can be incorporated into the presented page (W3C, 2005).



The Document Object Model (DOM)

Image 8 - The Document Object Model (Daniel Smith's Personal Blog, 2008)

3.5.2. HTML

HyperText Markup Language (HTML) is the main language used for coding web pages and all the web browser's displayable information. All the code is written in the form of HTML elements which are enclosed in angle brackets. These tags mainly come in pairs but there are some exceptions as well. In between these pairs can be added text, comments and other text-based content. After that a content can be opened in a web browser where all the tags are processed and displayed. The browser's work is to interpret the content. These elements form the building blocks of all websites. Images and objects are allowed to be used to create interactive forms. The HTML form can embed scripts written in languages as JavaScript to add dynamic content. To standardize the HTML it respects standards ISO/IEC 15445,

W3C HTML 4.01 and W3C HTML5 (W3C, 1999).

3.5.3. CSS

To describe the presentation semantics of document written in a markup language exists style called Cascading Style Sheets. Except styling HTML and XHTML pages it can be applied to any kind of XML document, including XML, SVG and XUL.

The primary purpose is the separation of document content from document presentation which can improve content accessibility, flexibility and control in the specifications. One of the biggest advantages is sharing same style within multiple pages. It is also possible to present page in different styles for different rendering methods, screen size or device which it is being viewed. The specifications are maintained by the W3C. Internet media type (MIME) is registered for use with RFC 2318 (W3C, 2012).

3.5.4. Scripting Languages

Scripting language is a high-level programming language interpreted by another program at runtime. It's not compiled by the computer's processor as other programming languages. Scripting languages, which can be embedded within HTML, are commonly used to add functionalities to a Web pages, such as menu styles, graphic displays or to serve dynamic advertisements. These languages affect the data that the end user sees in a browser. These languages are called client-side languages. Examples of scripting languages are JavaScript, ASP, JSP, PHP, Perl, Tcl and Python (Webopedia, 2013).

3.5.4.1. JavaScript

JavaScript is language influenced by the language C and many naming conventions come from the Java language. Self and Scheme languages influenced the key design principles (ECMAScript, 2007).

Javascript language is:

1. Open standard
2. A multi-paradigm
3. Imperative and functional
4. A prototype-based dynamic language
5. Weakly-typed scripting language with first class functions

From the C language it supports if, while loops, switch statements, etc. What it does not support is C-style block-level scoping. On the other hand JavaScript has its own function-level scoping. From the version 1.7, block-level scoping is supported with the let keyword. The semicolons terminating statements can be omitted, because in JavaScript is implemented automatic semicolons insertion. JavaScript does not work with classes, but works with constructors which behave very similar (FLANAGAN, 2006).

3.5.5. HTML Canvas

The <canvas> element is part of HTML5 which allows scriptable rendering of 2D shapes and bitmap images and basically it is only a container for graphics. It consists of a drawable region in HTML code that can be accessed by JavaScript through a collection of drawing functions like drawing paths, boxes, circles, characters, adding images or typing text (W3C, 2011).

To create a Canvas element in an HTML page, it is necessary to insert the following code:

```
<canvas id="example" width="200" height="200">
This text is displayed if your browser does not support HTML5 Canvas.
</canvas>
```

For a drawing it is important to initialize canvas by selecting it by its ID and using function getContext('2d').

```
var example = document.getElementById('example');
var context = example.getContext('2d');
context.fillStyle = 'red';
context.fillRect(30, 30, 50, 50);
```

3.5.6. jQuery

jQuery is Javascript's lightweight library offering many useful functions. The motto of jQuery is “write less, do more”, because offered functions facilitate very common tasks. A lot of commont task requiring many lines of JavaScript code are wraped into methods that can be called with a single line of code. It does not simplify only working with JavaScript, but implented are things like AJAx calls and DOM manipulation (W3C, 2010).

“Using raw JavaScript can result in doezens of lines of code for each of these tasks. The creators of jQuery specifically created the library to make common tasks trivial” (BIBEAULT, 2010).

The jQuery features:

1. HTML/DOM manipulation
2. CSS manipulation
3. HTML event methods
4. Effects and animations
5. AJAX
6. Utilities

3.5.7. .htaccess

.htaccess is a directory configuration file used for decentralized management of web server configuration. This file is placed inside the directory tree and is able to override server's global configuration for a current directory (The Apache Software Foundation, 2013a*)*. Changing a content of this file provides a way to make changes on a per-directory basis. The file containing configuration directives is then placed in particular folder and the directives are applied on that current directory and its subdirectories (The Apache Software Foundation, 2013b*)*.

URL requests coming onto server based on a regular-expression parser are allowed by The Apache module mod_rewrite to be rewrited. For example direct request for one subdirectory to a different directory or adding www to every request (MT, 2012).

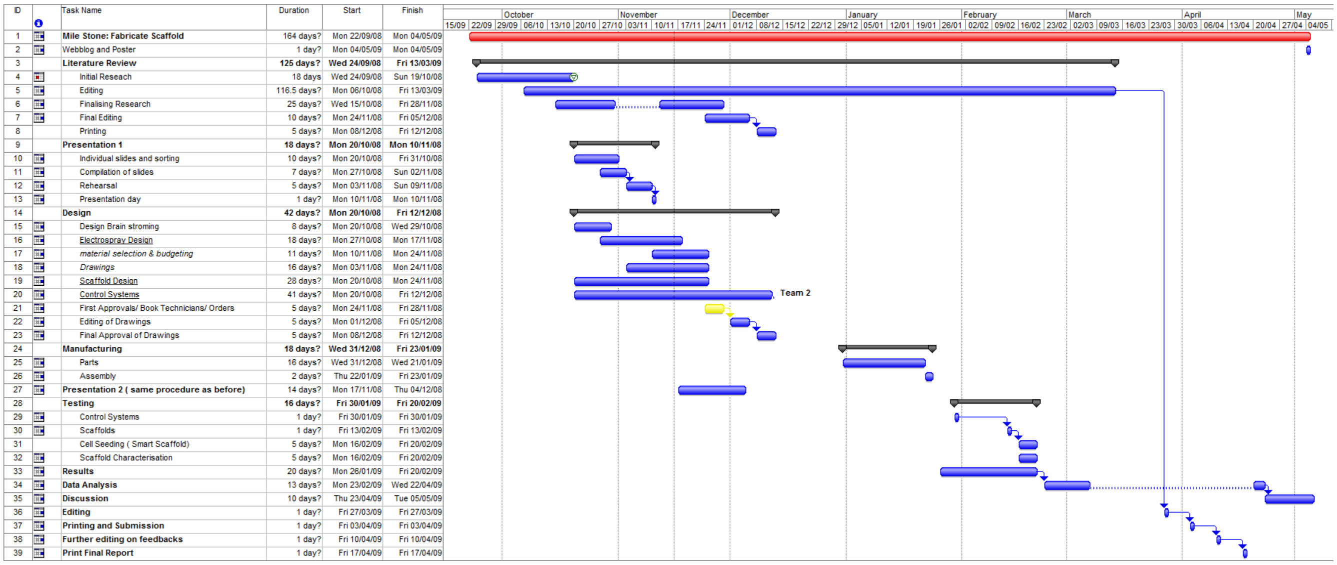
Example of configuration:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?somesite.com/.*$ [NC]
RewriteRule \.(gif|jpg|png)$ http://www.somesite.com/nasty.gif [R,L]
```

3.6. Scheduling Technologies

3.6.1. Gantt Chart

To illustrate a planning and progress of a project schedule in 1910s Henry Gantt developed a type of bar chart. This chart, called Gantt charts, displays start and finish dates of the terminal elements plus summary elements of a project. It helps to analyze and plan projects. Some Gantt charts display relationship between activities and their dependencies. Gantt charts are common techinque for representing the phases of a project and are understood all over the world. Gantt charts are appropriate for projects with about 30 activities, because in larger and more complex diagrams the readability decreses. It is because it communicates relatively little information per unit area of display (MindTools, 2013).



ID	ℹ	Task Name	Duration	Start	Finish
1	▦	**Mile Stone: Fabricate Scaffold**	164 days?	Mon 22/09/08	Mon 04/05/09
2	▦	Webblog and Poster	1 day?	Mon 04/05/09	Mon 04/05/09
3		**Literature Review**	**125 days?**	**Wed 24/09/08**	**Fri 13/03/09**
4	▦	Initial Reseach	18 days	Wed 24/09/08	Sun 19/10/08
5	▦	Editing	116.5 days?	Mon 06/10/08	Fri 13/03/09
6	▦	Finalising Research	25 days?	Wed 15/10/08	Fri 28/11/08
7	▦	Final Editing	10 days?	Mon 24/11/08	Fri 05/12/08
8		Printing	5 days?	Mon 08/12/08	Fri 12/12/08
9		**Presentation 1**	**18 days?**	**Mon 20/10/08**	**Mon 10/11/08**
10	▦	Individual slides and sorting	10 days?	Mon 20/10/08	Fri 31/10/08
11	▦	Compilation of slides	7 days?	Mon 27/10/08	Sun 02/11/08
12	▦	Rehearsal	5 days?	Mon 03/11/08	Sun 09/11/08
13	▦	Presentation day	1 day?	Mon 10/11/08	Mon 10/11/08
14		**Design**	**42 days?**	**Mon 20/10/08**	**Fri 12/12/08**
15	▦	Design Brain stroming	8 days?	Mon 20/10/08	Wed 29/10/08
16	▦	Electrospray Design	18 days?	Mon 27/10/08	Mon 17/11/08
17	▦	*material selection & budgeting*	11 days?	Mon 10/11/08	Mon 24/11/08
18	▦	*Drawings*	16 days?	Mon 03/11/08	Mon 24/11/08
19	▦	Scaffold Design	28 days?	Mon 20/10/08	Mon 24/11/08
20	▦	Control Systems	41 days?	Mon 20/10/08	Fri 12/12/08
21	▦	First Approvals/ Book Technicians/ Orders	5 days?	Mon 24/11/08	Fri 28/11/08
22	▦	Editing of Drawings	5 days?	Mon 01/12/08	Fri 05/12/08
23	▦	Final Approval of Drawings	5 days?	Mon 08/12/08	Fri 12/12/08
24		**Manufacturing**	**18 days?**	**Wed 31/12/08**	**Fri 23/01/09**
25	▦	Parts	16 days?	Wed 31/12/08	Wed 21/01/09
26	▦	Assembly	2 days?	Thu 22/01/09	Fri 23/01/09
27	▦	**Presentation 2 (same procedure as before)**	14 days?	Mon 17/11/08	Thu 04/12/08
28		**Testing**	**16 days?**	**Fri 30/01/09**	**Fri 20/02/09**
29	▦	Control Systems	1 day?	Fri 30/01/09	Fri 30/01/09
30	▦	Scaffolds	1 day?	Fri 13/02/09	Fri 13/02/09
31		Cell Seeding (Smart Scaffold)	5 days?	Mon 16/02/09	Fri 20/02/09
32	▦	Scaffold Characterisation	5 days?	Mon 16/02/09	Fri 20/02/09
33	▦	**Results**	20 days?	Mon 26/01/09	Fri 20/02/09
34	▦	**Data Analysis**	13 days?	Mon 23/02/09	Wed 22/04/09
35	▦	**Discussion**	10 days?	Thu 23/04/09	Tue 05/05/09
36	▦	**Editing**	1 day?	Fri 27/03/09	Fri 27/03/09
37	▦	**Printing and Submission**	1 day?	Fri 03/04/09	Fri 03/04/09
38	▦	**Further editing on feedbacks**	1 day?	Fri 10/04/09	Fri 10/04/09
39	▦	**Print Final Report**	1 day?	Fri 17/04/09	Fri 17/04/09

Image 9 - Gantt Chart (3D Electro Writers, 2008)

3.6.2. Component Diagram

To illustrate the pieces of software or some dependent parts of a system is used Unified Modelling Language Component Diagram. Comparing to a Class Diagram it offers a higher level of abstraction, because Class Diagrams are focused on one or more objects at runtime. The possibility of building blocks allows to encompass a large part of a system. Grouping elements into logical structures is done by components, which are very similar to package diagrams. Components can define boundaries as well. The only difference between component diagrams and package diagrams is that Component Diagrams has more semantically rich grouping mechanism and it offers the privacy for elements. Package diagrams display only public elements (Sparx Systems, 2012).

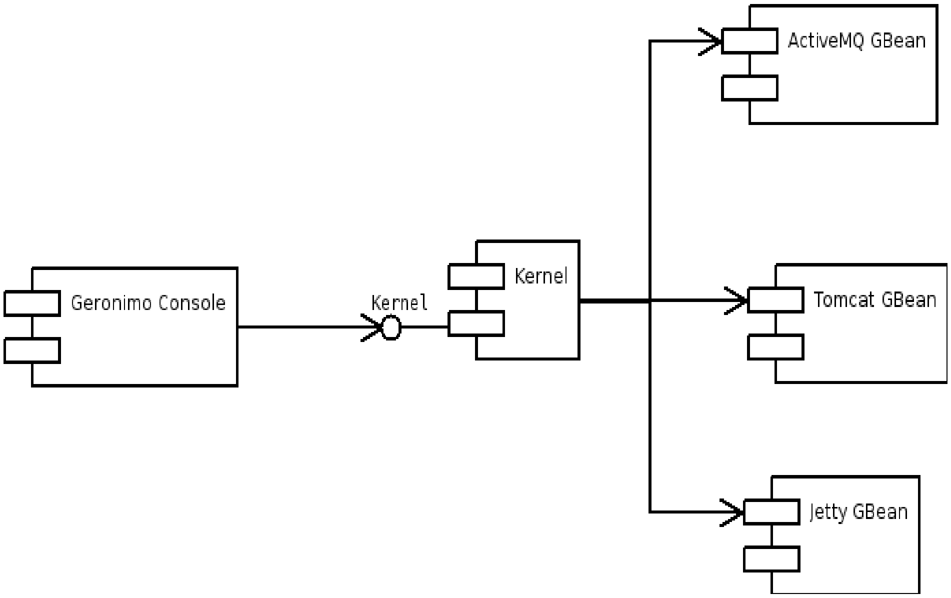


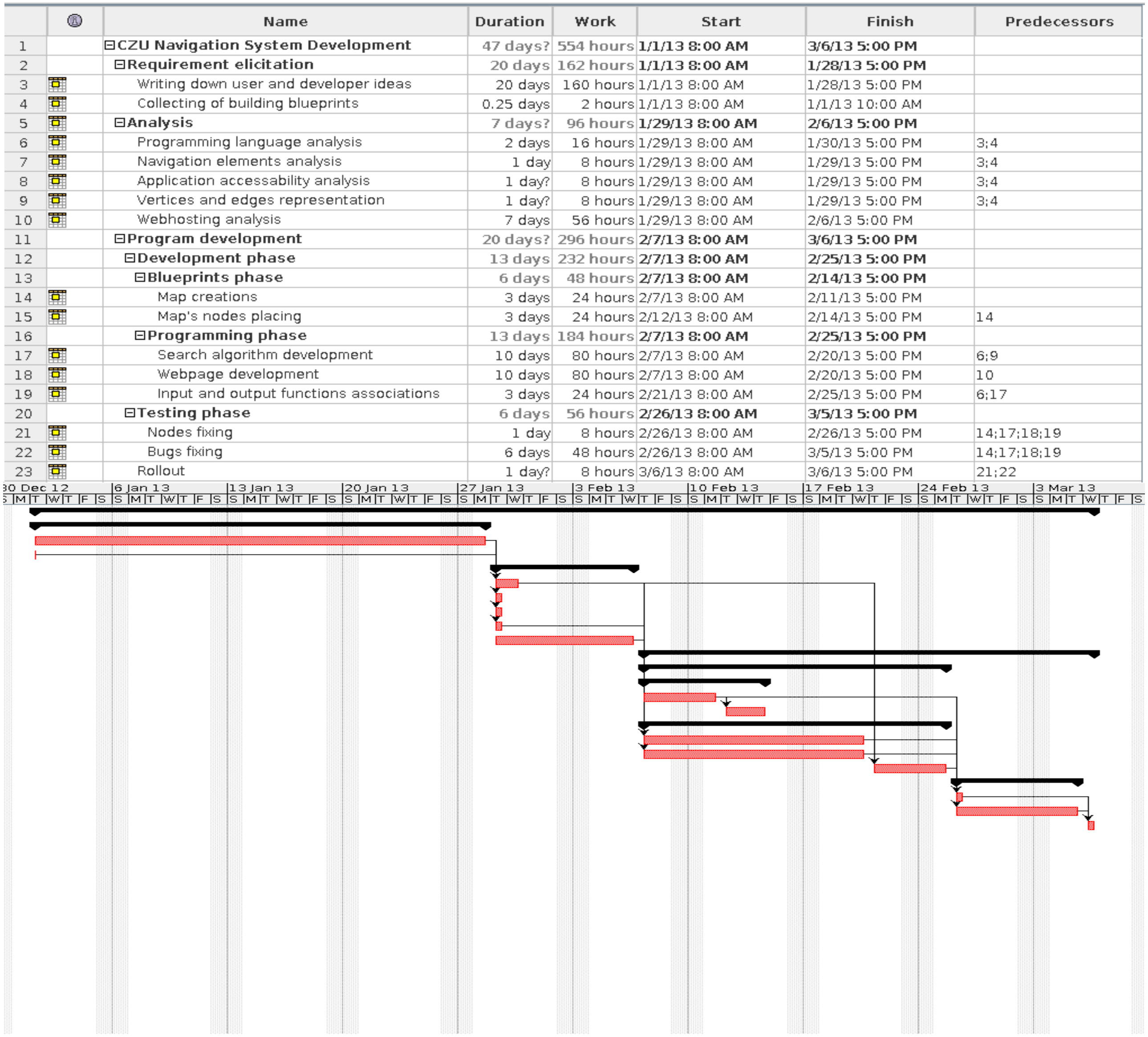
Image 10 - Component Diagram (Apache GERONIMO, 2009)

II. Project

4. Navigation Application

4.1. Project Schedule

For a project scheduling author uses OpenProj software. There he schedules all the tasks needed to accomplish the project. The result are displayed in a Gantt chart below.



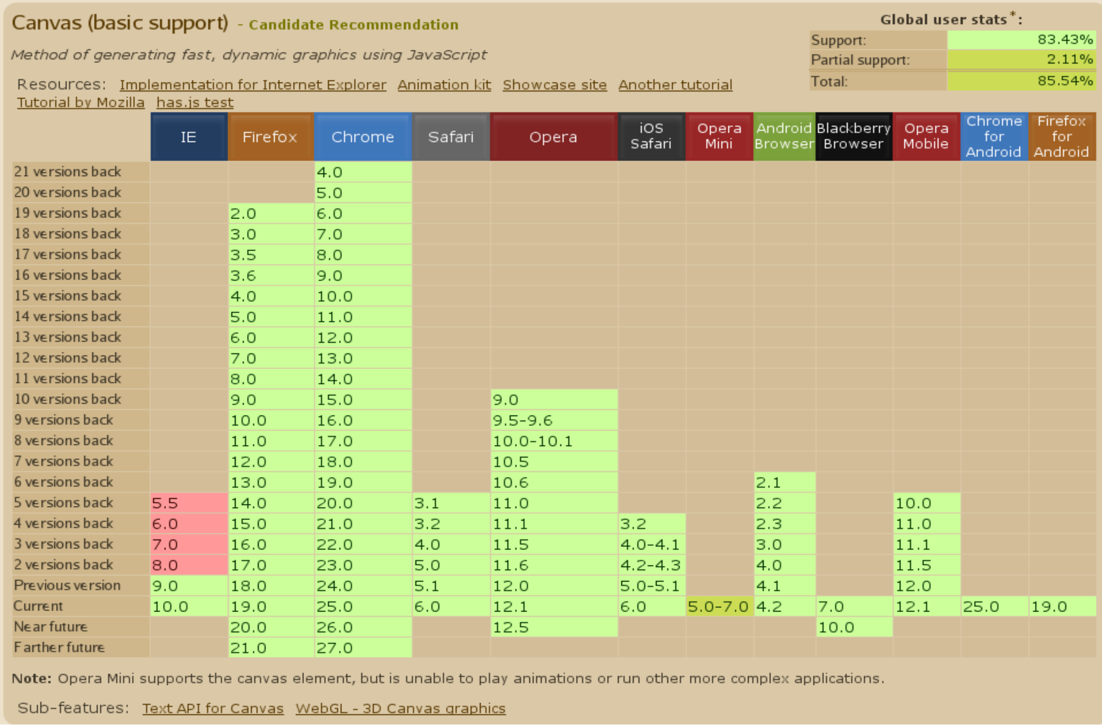
		Name	Duration	Work	Start	Finish	Predecessors
1		⊟CZU Navigation System Development	47 days?	554 hours	1/1/13 8:00 AM	3/6/13 5:00 PM	
2		⊟Requirement elicitation	20 days	162 hours	1/1/13 8:00 AM	1/28/13 5:00 PM	
3		Writing down user and developer ideas	20 days	160 hours	1/1/13 8:00 AM	1/28/13 5:00 PM	
4		Collecting of building blueprints	0.25 days	2 hours	1/1/13 8:00 AM	1/1/13 10:00 AM	
5		⊟Analysis	7 days?	96 hours	1/29/13 8:00 AM	2/6/13 5:00 PM	
6		Programming language analysis	2 days	16 hours	1/29/13 8:00 AM	1/30/13 5:00 PM	3;4
7		Navigation elements analysis	1 day	8 hours	1/29/13 8:00 AM	1/29/13 5:00 PM	3;4
8		Application accessability analysis	1 day?	8 hours	1/29/13 8:00 AM	1/29/13 5:00 PM	3;4
9		Vertices and edges representation	1 day?	8 hours	1/29/13 8:00 AM	1/29/13 5:00 PM	3;4
10		Webhosting analysis	7 days	56 hours	1/29/13 8:00 AM	2/6/13 5:00 PM	
11		⊟Program development	20 days?	296 hours	2/7/13 8:00 AM	3/6/13 5:00 PM	
12		⊟Development phase	13 days	232 hours	2/7/13 8:00 AM	2/25/13 5:00 PM	
13		⊟Blueprints phase	6 days	48 hours	2/7/13 8:00 AM	2/14/13 5:00 PM	
14		Map creations	3 days	24 hours	2/7/13 8:00 AM	2/11/13 5:00 PM	
15		Map's nodes placing	3 days	24 hours	2/12/13 8:00 AM	2/14/13 5:00 PM	14
16		⊟Programming phase	13 days	184 hours	2/7/13 8:00 AM	2/25/13 5:00 PM	
17		Search algorithm development	10 days	80 hours	2/7/13 8:00 AM	2/20/13 5:00 PM	6;9
18		Webpage development	10 days	80 hours	2/7/13 8:00 AM	2/20/13 5:00 PM	10
19		Input and output functions associations	3 days	24 hours	2/21/13 8:00 AM	2/25/13 5:00 PM	6;17
20		⊟Testing phase	6 days	56 hours	2/26/13 8:00 AM	3/5/13 5:00 PM	
21		Nodes fixing	1 day	8 hours	2/26/13 8:00 AM	2/26/13 5:00 PM	14;17;18;19
22		Bugs fixing	6 days	48 hours	2/26/13 8:00 AM	3/5/13 5:00 PM	14;17;18;19
23		Rollout	1 day?	8 hours	3/6/13 8:00 AM	3/6/13 5:00 PM	21;22

Image 11 - OpenProj Project Schedule

4.2. Application Requirements

4.2.1. Accessibility

Under the term accessibility is mainly meant accessibility of the application for smartphone devices. The application is web browser based, so user does not need to install the application separately depending on his smartphone's OS or architecture. For example to develop application for Android developer needs to use Java, for iPhones developer needs to use Objective C and for Windows Phone C#. More so to develop application for each OS used on smartphones is time consuming and inefficient in terms of software updates. The next advantage of this solution is guaranteed access from any browser on PC, tablet, etc. Finally to program web based application does need only one programming language to use in combination with HTML and CSS. The main disadvantage of web browser based application is the need of an on-line access. The next disadvantage is browser support which can be seen in the table below.



Canvas (basic support) - Candidate Recommendation

Method of generating fast, dynamic graphics using JavaScript

Resources: Implementation for Internet Explorer Animation kit Showcase site Another tutorial Tutorial by Mozilla has.js test

Global user stats*:

Support:	83.43%
Partial support:	2.11%
Total:	85.54%

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android
21 versions back			4.0									
20 versions back			5.0									
19 versions back		2.0	6.0									
18 versions back		3.0	7.0									
17 versions back		3.5	8.0									
16 versions back		3.6	9.0									
15 versions back		4.0	10.0									
14 versions back		5.0	11.0									
13 versions back		6.0	12.0									
12 versions back		7.0	13.0									
11 versions back		8.0	14.0									
10 versions back		9.0	15.0		9.0							
9 versions back		10.0	16.0		9.5-9.6							
8 versions back		11.0	17.0		10.0-10.1							
7 versions back		12.0	18.0		10.5							
6 versions back		13.0	19.0		10.6			2.1				
5 versions back	5.5	14.0	20.0	3.1	11.0			2.2		10.0		
4 versions back	6.0	15.0	21.0	3.2	11.1	3.2		2.3		11.0		
3 versions back	7.0	16.0	22.0	4.0	11.5	4.0-4.1		3.0		11.1		
2 versions back	8.0	17.0	23.0	5.0	11.6	4.2-4.3		4.0		11.5		
Previous version	9.0	18.0	24.0	5.1	12.0	5.0-5.1		4.1		12.0		
Current	10.0	19.0	25.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0	12.1	25.0	19.0
Near future		20.0	26.0		12.5				10.0			
Farther future		21.0	27.0									

Note: Opera Mini supports the canvas element, but is unable to play animations or run other more complex applications.

Sub-features: Text API for Canvas WebGL - 3D Canvas graphics

Image 12 - Canvas support (Can I use, 2013)

4.2.2. Usability

Under the term usability is meant user friendly GUI. The main requirement is to deliver as least as possible buttons to click and fields to fill. Every time when user hovers over a field, application automatically pop ups fields where can user select his preferred destination. Author uses only three fields and two buttons.

Fields:

1. From – User types in or selects the destination where he wants to get from
2. To - User types in or selects the destination where he wants to get to
3. Loc - User types in or selects the location which he wants to show on a map

Buttons:

1. Show - This button takes input from the field Loc and runs a function to show a desired location on a map
2. Find – This buttons takes input from fields From and To and runs a function to find a route

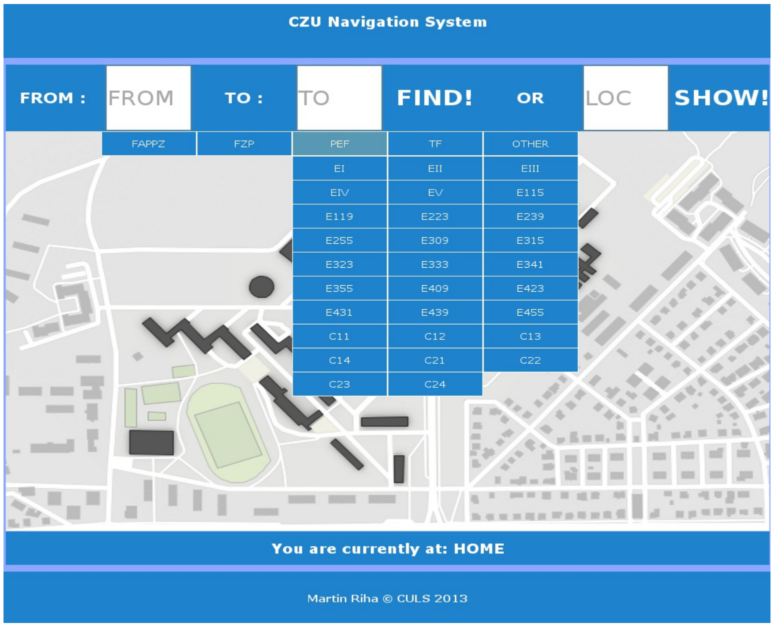


Image 13 - User Interface

4.3. Application Analysis

Because the application is web based, author decides to use JavaScript in combination with HTML and CSS. The reason for JavaScript is the universality (nativity to the web browser) and functionality with HTML5 elements, mainly with HTML5 Canvas, which, as stated before, is one of the best solutions to draw on-line.

The next thing is representation of nodes in the program. Author decides to use adjacency list data representation. On one hand it comes with redundancy of vertices going out from each node. On the other hand it comes with better readability and orientation in the array.

All nodes in the campus outside buildings are 5 digits numbers starting with number 1. The reason for this numbering is that JavaScript does not support numbering starting with number 0. The reason for 5 digit number is extendability in the future.

Nodes inside buildings have different numbering. It starts with character determining the current building and 4 digit number after that. For example the PEF building's first character is P. The first digit number determines the floor. The basement has number 0, first floor number 1, etc. The stairway's nodes are common for all floors. It means that node from the basement directly connects to all floors and opposite.

For instance the node in the fourth floor in the PEF building looks like the code below:

```
{node:"P4009",loc:"PEF4",to:["P4008","E455"],x:627,y:403}
```

And a node in the campuse looks like:

```
{node:10038,loc:"OUT",to:[10034,10037,"FARMA"],x:484,y:396}
```

4.4. Program Development

4.4.1. Introduction

Firstly it's important to insert into the project's HTML webpage link to used JavaScript applications by adding lines:

```
<script type="text/javascript" src="jquery-1.8.3.min.js"></script>
<script type='text/javascript' src='application.js'></script>
```

The first link links to some of used Jquery functions and the second one links to the application file called application.js.

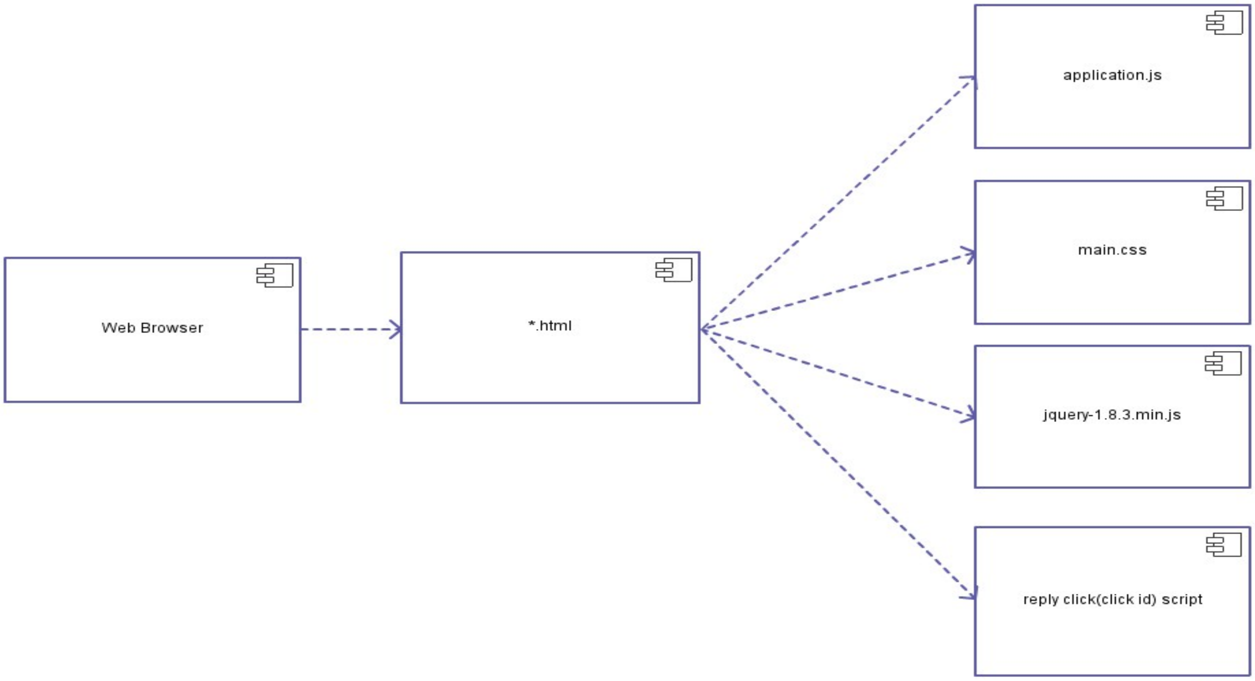


Image 14 - Application Scheme

On the first line is needed to add `$(document).ready(function(){` which means that everything after the bracket will load as soon as the DOM is loaded and before the page contents are loaded.

4.4.2. Canvas Elements

Because the application works with five Canvas elements, it's needed to initialize them in when the page is loaded.

```
var ctx0 = $('#canvas0')[0].getContext("2d");
var ctx1 = $('#canvas1')[0].getContext("2d");
var ctx2 = $('#canvas2')[0].getContext("2d");
var ctxG1 = $('#canvasG1')[0].getContext("2d");
var ctxG2 = $('#canvasG2')[0].getContext("2d");
```

The reason for the use of five Canvas elements is obvious. Canvas ctx0 and ctx2 are used for displaying interiors of buildings. The ctx1 elements is for outside view. The ctxG1 element is used when the route is found from any building from higher than the first floor. The same use is for ctxG2 element besides that it is displayed only, it the destination is located in any building in higher than first floor.

4.4.3. Nodes Array

As stated in the Application Analysis part, all nodes are in an associative array. The name of the associative array is nodes. The structure is shown below:

```
{node:<name of the node>,loc:<location>,to:[<edges>],x:<x coordinates>,y:<y coordinates>}
```

`<name of the node>` - is unique named node in the nodes matrix

`<location>` - is location of the node. If location is outside in campus, the loc is “OUT”, otherwise it marks the building like “PEF”

`<edges>` - in this array are located all edges of the current node

`<x coordinates>,<y coordinates>` - x and y coordinates of the current node

```
var nodes = [
//Faculty of Agrobiology, Food and Natural Renodes[from].nodes
{node:"FAPPZ",loc:"OUT",to:[10100],x:604,y:207},
…
318 lines
…
{node:"P4009",loc:"PEF4",to:["P4008","E455"],x:627,y:403},
];
```

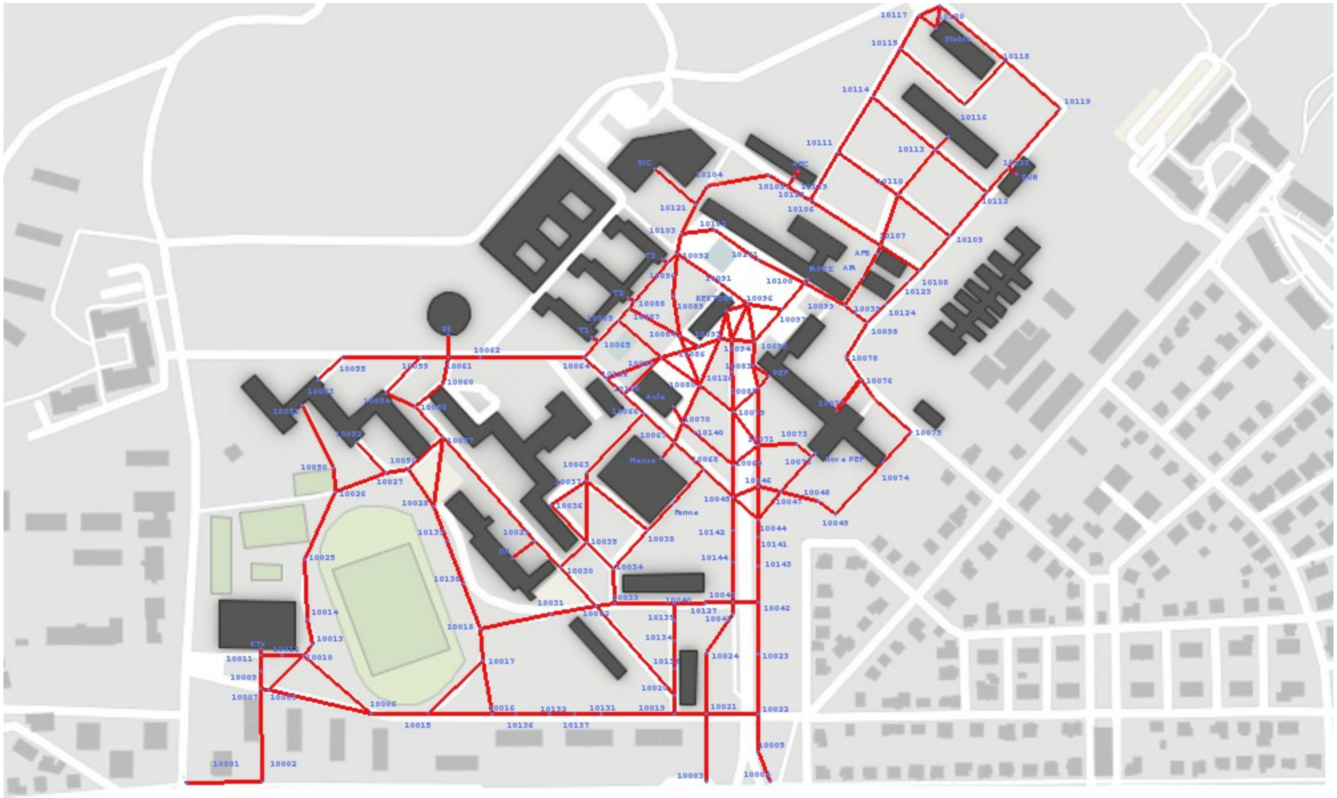


Image 15 - Campus Map

4.4.4. Mistake Finder

When are all the nodes inserted into an array, it can happen that there are some mistakes (insufficiency or redundancy of node's links). For these purposes author develops a Mistake finder function which finds these errors and outputs them as an alert windows in the HTML web page. The Find mistake button is hidden on the web page and can be displayed by changing behavior of an initialization function.

```
$('#mistake').click(function () {
    for (i in nodes)
  {
    numberOfFindings=0;
    for (z in nodes)
    {
      for (y in nodes[z].to)
      {
        if (nodes[i].node == nodes[z].to[y])
        {
          numberOfFindings++;
        }
      }
    }
    if (numberOfFindings != nodes[i].to.length)
    {
          alert (nodes[i].node+" leads to "+nodes[i].to+" has been found
"+numberOfFindings+ " times ");
    }
  }
  alert("Looks ok!");
});
```

4.4.5. Initialization

When the page is loaded an initialization function is runned. The purpose of this function is to hide all unnecessary elements on the page. It hides all Canvas elements except the canvas1 which is used as a initial background. It hides Mistake finder button and loading progress as well. More so this function empties all arrays and number of hops sets to 0. In the beginning it calls the clearCanvas function which erases all the drawn routes on canvases.

```
function init() {
  $('#position').show();
  $('#newsearch').hide();
  $('#mistake').hide();
  $('#canvas0').hide();
  $('#canvas1').show();
  $('#canvas2').hide();
  $('#canvasG1').hide();
  $('#canvasG2').hide();
  $('#loading').hide();
  $('#main').show();
  $('#text').show();
  path = [];
  child = [];
  parent = [];
  length = [];
  hops = 0;
  document.getElementById("canvas1").style.background = "url(map/mapa1.jpg)";
  clearCanvas();  };
function clearCanvas() {
ctx0.clearRect(0, 0, 1000, 600);
ctx1.clearRect(0, 0, 1000, 600);
ctx2.clearRect(0, 0, 1000, 600);
ctxG1.clearRect(0, 0, 1000, 600);
ctxG2.clearRect(0, 0, 1000, 600);
};
```

4.4.6. Supplementary Functions

The application uses two supplementary functions used in main functions. The first one is inArray function which has parameters object and array. Its purpose is to return true statement if desired object is in array.

```
function inArray(object,array)
{
  var flag = false;
  for (i in array)
  {
    if (array[i]==object)
    {
      flag = true;
    }
  }
  return flag;
};
```

The getNodeNumber function's purpose is to return the i-th position of desired element in desired array. If an element does not exist in array, it alerts message containing warning that element does not exist. This warning is for example displayed when user types in not existing node and clicks find or show, depending on a field.

```
function getNodeNumber(name,array) {
  for (i in array)
  {
    if (array[i].node==name)
    {
     return i;
    }
  }
 alert(name+" does not exist!");
 return false;
};
```

4.4.7. Find Route Function

This function is the main function of the application. It is associated with the Find button on the HTML web page. As inputs it takes text from the fields from and to. The scheme of the function is on the figure below

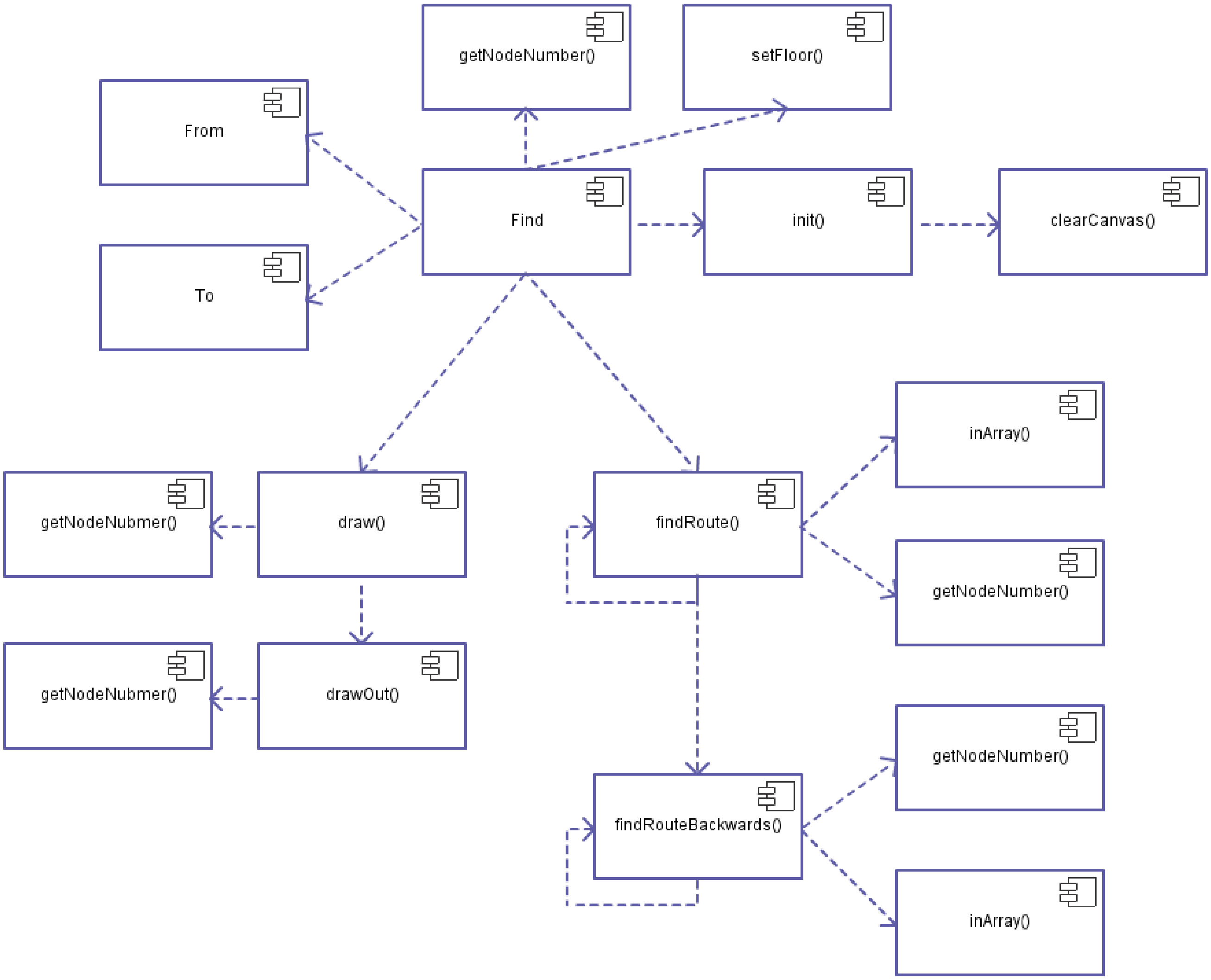


Image 16 - Find Component Diagram

Firstly this function calls init function which initializes arrays needed for the process of route finding and sets variable hops to zero. These arrays are child, parent, length and path. After getting the arguments from and to and in case it's fields are not empty, it sets from destination as a root vertex and calls function findRoute with from and to arguments.

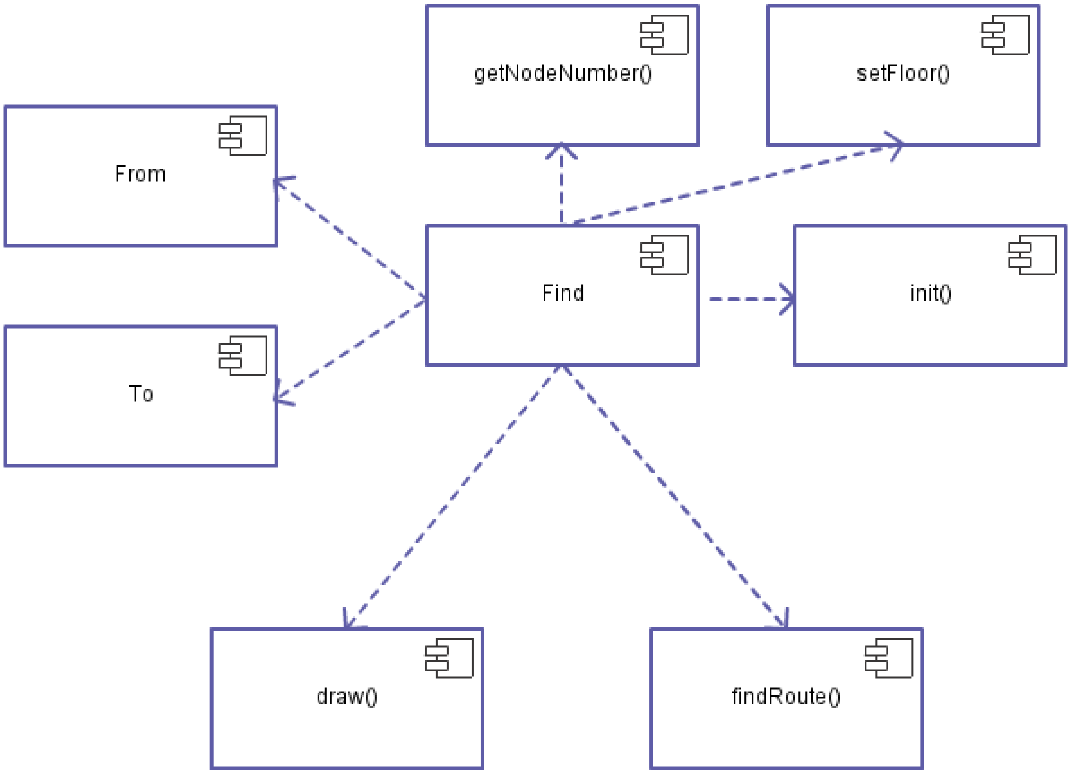


Image 17 - Find Component Diagram

```
$('#route').click(function() {
  init();
  $('#position').hide();
  $('#canvas1').hide();
  $('#main').hide();
  $('#text').hide();
  $('#loading').show();
  $('#greenDivD').hide();
  var from = [document.getElementById('from').value.toUpperCase()];
  var to = [document.getElementById('to').value.toUpperCase()];

  fromNode = getNodeNumber(from,nodes);
  toNode = getNodeNumber(to,nodes);
  if (!fromNode || !toNode)
  {
    init();
    return;
  }
  parent.push(from);
  path.push(from);
  findRoute(from,to);
  setFloor(nodes[fromNode].loc,"F");
  setFloor(nodes[toNode].loc,"T");

  $('#loading').hide();
  $('#main').show();
  draw();
  document.getElementById("main").style.maxHeight = "100%";

});
```

The findRoute function is a recursive function. The base case is when any of the current root vertex's edges is equal to a final destination stored in the variable to. The recursive step searches the edges of all current root vertices and pushes them into the parent variable. Finally the function calls itself with the parent array argument as a first parameter.

When the base case condition is met the path array contains the number of arrays as is number of vertices between from and to destination, plus from and to destination. In each of these arrays are stored all edges of all skimmed vertices. Finally it starts backward search by calling the findRouteBackwards function.

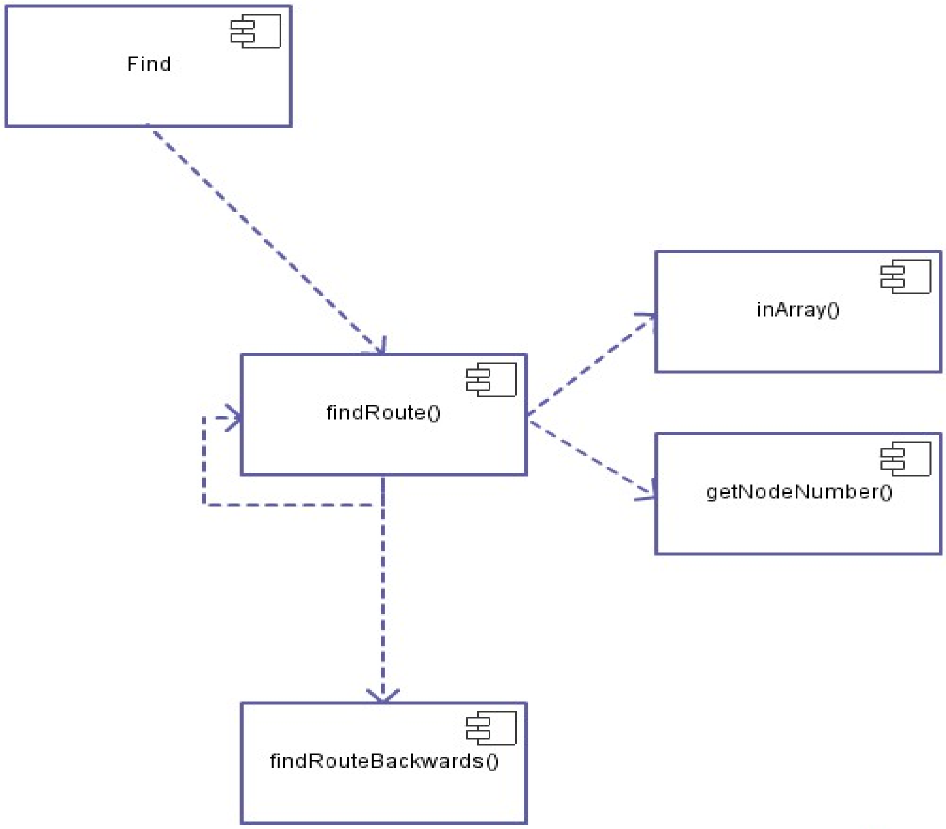


Image 18 - findRoute() Component Diagram

```
function findRoute(from,to) {

  if (inArray(to,from))
  {
    length.reverse();
    path.reverse();
    path[0]=to;
    findRouteBackwards(to,1);

  }
  else
  {
    $('#loading').append(".");
    for (i in parent)
    {
     arr1 = nodes[getNodeNumber(parent[i],nodes)].to;

     for (y in arr1)
     {
       if (inArray(arr1[y],path)==false && (inArray(arr1[y],child))==false) {
       child.push(arr1[y]);
       }
      }
    }
    length.push(child.length);
    path.push(child.slice());
    parent = child.slice();
    parent = child;
    child=[];
    hops++;
    findRoute(parent,to);
  }
};
```

The findRouteBackwards is a recursive function as well. It skims all the arrays in the path array and if it finds that from the current element in array leads any edge to any element of the next array, this element replaces the current array where does the element comes from. Finally the path array has only one element in each array which is the final route.

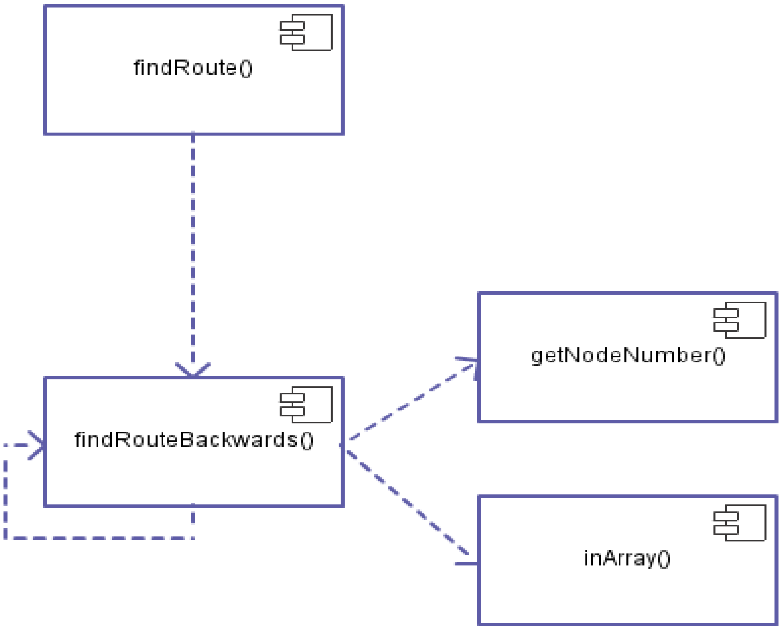


Image 19 - findRouteBackwards() Component Diagram

```
function findRouteBackwards(to,y){
  knik = nodes[getNodeNumber(to,nodes)].to;
  for (z in knik)
  {
      if (inArray(knik[z],path[y]))
      {
      path[y]=knik[z];
      }
  }
  to=path[y];
  if (y > hops)
  {
    path.reverse();
    return true;
  };
  y=y+1;
  findRouteBackwards(to,y);};
```

Because background of a canvas depends on a current floor and building, it's necessary to set background images according to a location. In the function setFloor is a switch statement which sets background style of particular canvas.

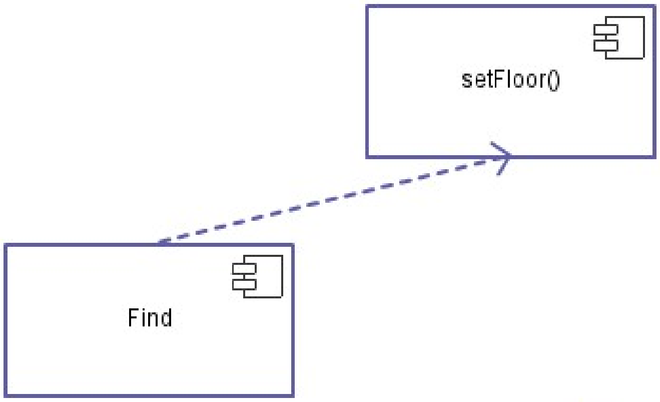


Image 20 - setFloor() function Component Diagram

```
function setFloor(node,x)
{
 if (x == "F") {
  a="canvas0";
 }
 else if (x == "T") {
  a="canvas2";
 }

 switch (node)
  {
  case "PEF1":
    document.getElementById(a).style.background = "url(map/1np.jpg)";
    break;
  case "PEF2":
    document.getElementById(a).style.background = "url(map/2np.jpg)";
    break;
  case "PEF3":
    document.getElementById(a).style.background = "url(map/3np.jpg)";
    break;
  case "PEF4":
    document.getElementById(a).style.background = "url(map/4np.jpg)";
    break;
 };
};
```

In the end the findRoute function calls the draw function which is responsible for drawing a route onto canvases.

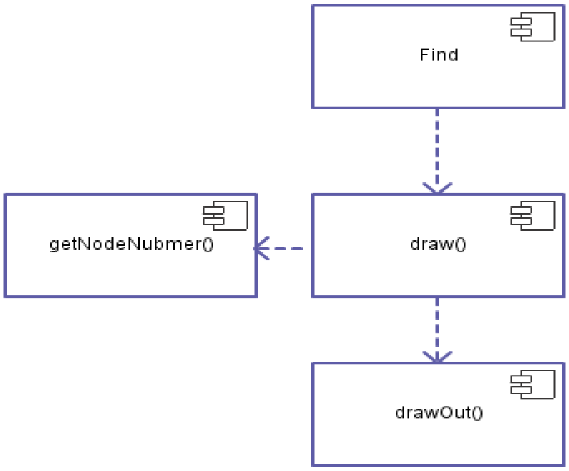


Image 21 - draw() function Component Diagram

```
function draw() {

  var path0 = [];
  var path1 = [];
  var path2 = [];
  var pathG1 = [];
  var pathG2 = [];
  firstDest = nodes[getNodeNumber(path[0],nodes)];
  finalDest = nodes[getNodeNumber(path.slice(-1)[0],nodes)];
```

The first task is to split a route onto parts according to a location. If nodes belong to same location, they are pushed into particular array depending on their position in the path array. The reason is that each part of split route has is binded with particular canvas element.

```
for (i in path)
 {
   u=path[i];

   kvik = nodes[getNodeNumber(path[i],nodes)].loc;
   if (kvik == "OUT")
   {
     path1.push(u);
   }
   else if (kvik == firstDest.loc)
   {
     path0.push(u);
   }
   else if (kvik == finalDest.loc) {
     path2.push(u);
   }
 }
```

The next important thing is setting the ground floor if a destination is in a higher than in a first floor.

```
for (i in path)
{
  u=path[i];

  kvik = nodes[getNodeNumber(path[i],nodes)].loc;
  if (kvik == "PEF1")
    {
    if(path0.length == 0) {
      pathG2.push(u);
      document.getElementById("canvasG2").style.background = "url(map/1np.jpg)";
    }
    else
    {
      pathG1.push(u);
      document.getElementById("canvasG1").style.background = "url(map/1np.jpg)";
    }
    }
}
```

The sequence of if statements initializes canvas elements depending if the particular array for current canvas is empty or not. If not it displays canvas on HTML web page and calls the drawOut function which is responsible for drawing routes on particular canvases.

```
if(path0.length != 0)
{
node1=nodes[getNodeNumber(path0[0],nodes)];
node2=nodes[getNodeNumber(path0.slice(-1)[0],nodes)];
$('#canvas0').show();
drawOut(ctx0,node1,node2,path0);
}
if(path1.length != 0)
{
node3=nodes[getNodeNumber(path1[0],nodes)];
node4=nodes[getNodeNumber(path1.slice(-1)[0],nodes)];
$('#canvas1').show();
drawOut(ctx1,node3,node4,path1);
}
if(path2.length != 0)
{
node5=nodes[getNodeNumber(path2[0],nodes)];
node6=nodes[getNodeNumber(path2.slice(-1)[0],nodes)];
$('#canvas2').show();
drawOut(ctx2,node5,node6,path2);
}
```

```
if(pathG2.length != 0)
{
node7=nodes[getNodeNumber(pathG2[0],nodes)];
node8=nodes[getNodeNumber(pathG2.slice(-1)[0],nodes)];
$('#canvasG2').show();
drawOut(ctxG2,node7,node8,pathG2);
}
if(pathG1.length != 0)
{
node9=nodes[getNodeNumber(pathG1[0],nodes)];
node10=nodes[getNodeNumber(pathG1.slice(-1)[0],nodes)];
$('#canvasG1').show();
drawOut(ctxG1,node9,node10,pathG1);
}
```

For a good orientation is marked the from and to node on the map by its name. It is done by conditionals listed in the code below.

```
  if (firstDest.loc == "OUT")
{
  ctx1.strokeStyle = 'blue';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.strokeText(firstDest.node, firstDest.x-20, firstDest.y-15);

  ctx1.fillStyle = '#E0F574';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.fillText(firstDest.node, firstDest.x-20, firstDest.y-15);
 }
else
{
  ctx0.strokeStyle = 'blue';
  ctx0.font = 'bold 18px sans-serif';
  ctx0.textBaseline = 'bottom';
  ctx0.strokeText(firstDest.node, firstDest.x-20, firstDest.y-15);

  ctx0.fillStyle = '#E0F574';
  ctx0.font = 'bold 18px sans-serif';
  ctx0.textBaseline = 'bottom';
  ctx0.fillText(firstDest.node, firstDest.x-20, firstDest.y-15);
}

if (finalDest.loc == "OUT")
{
  ctx1.strokeStyle = 'blue';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.strokeText(finalDest.node, finalDest.x-20, finalDest.y-15);


  ctx1.fillStyle = '#E0F574';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.fillText(finalDest.node, finalDest.x-20, finalDest.y-15);

}
else
{

  ctx2.strokeStyle = 'blue';
  ctx2.font = 'bold 18px sans-serif';
  ctx2.textBaseline = 'bottom';
  ctx2.strokeText(finalDest.node, finalDest.x-20, finalDest.y-15);

  ctx2.fillStyle = '#E0F574';
```

```
    ctx2.font = 'bold 18px sans-serif';
    ctx2.textBaseline = 'bottom';
    ctx2.fillText(finalDest.node, finalDest.x-20, finalDest.y-15);
  }
```

And finally this condition hides ground floor canvases if a destination from or to is located in a ground floor.

```
         if    (firstDest.loc.charAt(firstDest.loc.length-1    )    ==    1    ||
finalDest.loc.charAt(finalDest.loc.length-1 )==1)
  {
    $('#canvasG1').hide();
    $('#canvasG2').hide();
  }
};
```

The drawOut function skims all the nodes of its path parameter and draws line from point to point. The first node is marked by a blue colored circle and termination one by yellow colored.

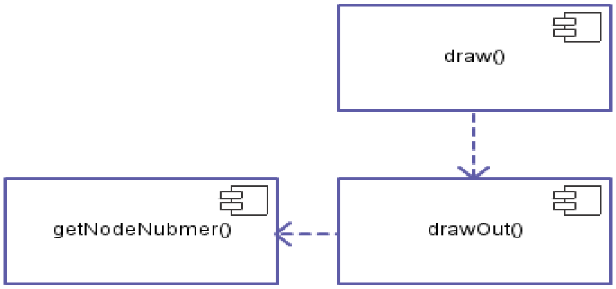


Image 22 - drawOut() function Component Diagram

```
function drawOut(where,nodea,nodeb,path) {

    where.fillStyle = '#5B69EC';
    where.beginPath();
    where.arc(nodea.x, nodea.y, 7, 0, Math.PI*2, true);

    where.closePath();
    where.fill();

    where.fillStyle = '#E0F574';
    where.beginPath();
    where.arc(nodeb.x, nodeb.y, 9, 0, Math.PI*2, true);

    where.closePath();
    where.fill();

    where.lineWidth = 4;
    where.strokeStyle = '#5B69EC';
    where.beginPath();
    node1=nodes[getNodeNumber(path[0],nodes)];
    where.moveTo(nodea.x,nodea.y);
    for (i in path) {

      node=nodes[getNodeNumber(path[i],nodes)];
      where.lineTo(node.x,node.y);
      where.lineTo(node.x,node.y);
      where.stroke();
    }
  };
```

For example the result of this function if user enters 'C11' into from field and 'AULA' into to field can be seen below.

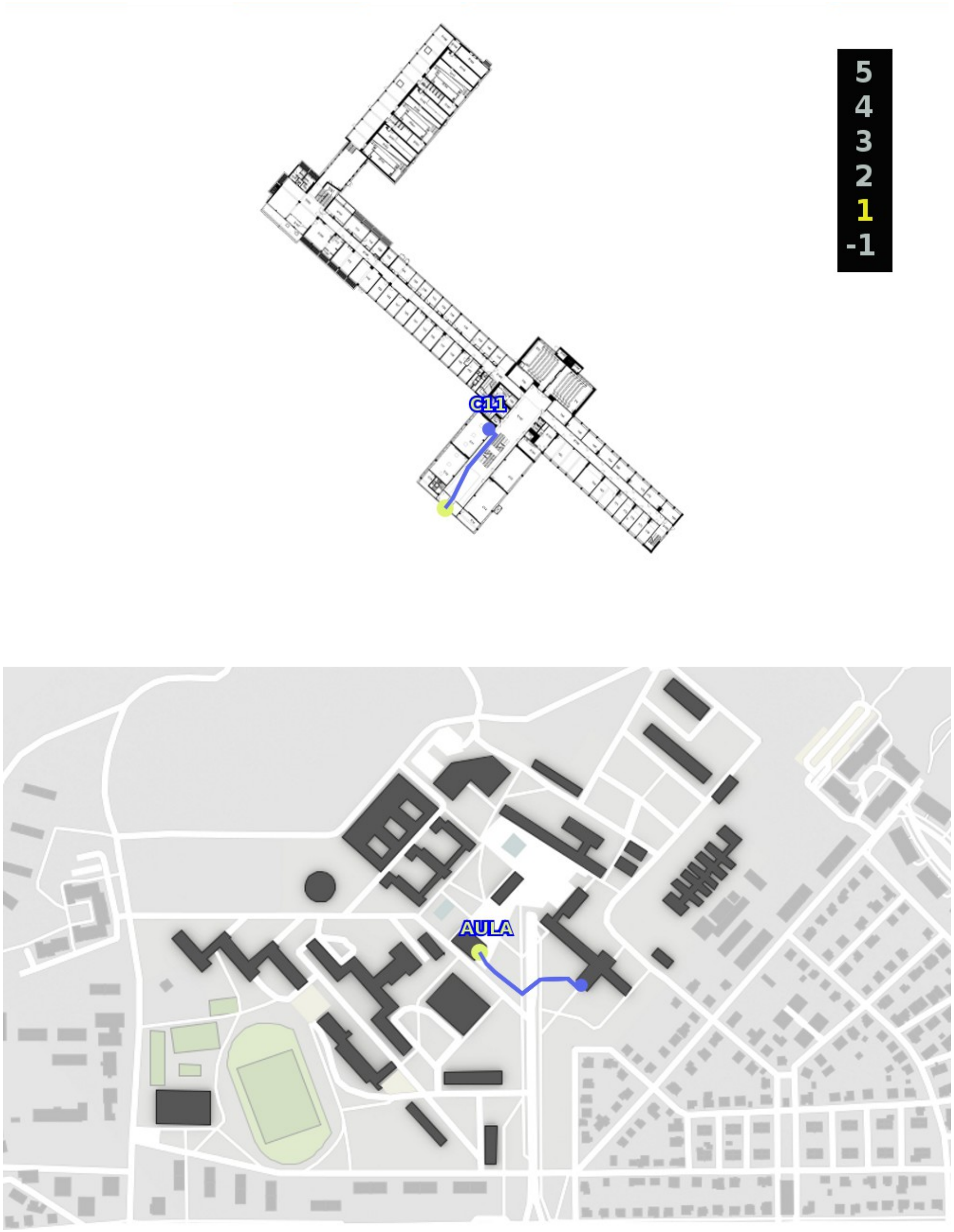


Image 23 - Find Route from C11 to Aula

4.4.8. Show Location Function

This function is binded with Loc field and Show button. The Loc field is an input parameter for this function and the button is binded with the showLoc function. If a destined position is inside a building it displays current floor and campus map with highlighted node number and building's entrance. This is a key function for QR code scannning, because it is called when user visits appropriate url.

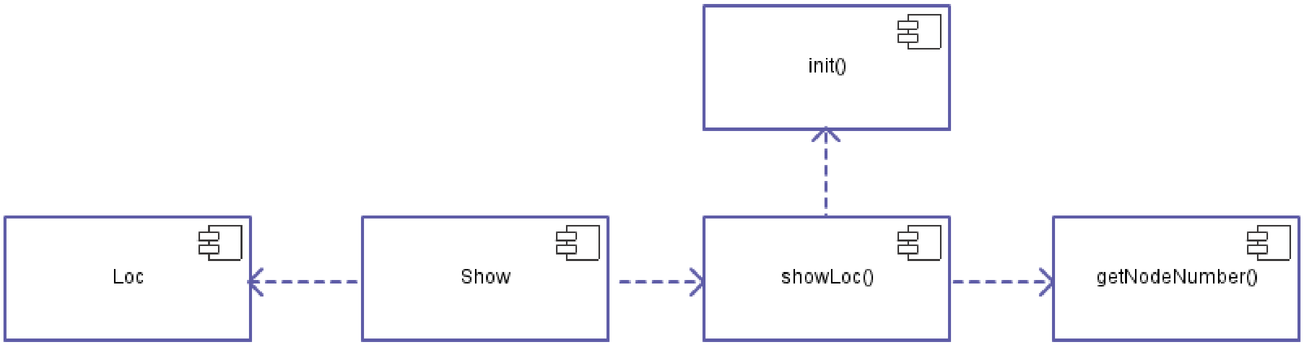


Image 24 - Show Component Diagram

```
function showLoc(input){
  init();
  $('#canvas1').hide();
  node=nodes[getNodeNumber(input,nodes)];

  if(node.loc == "OUT")
  {
    $('#canvas1').show();

    ctx1.fillStyle = '#5B69EC';
    ctx1.beginPath();
    ctx1.arc(node.x, node.y, 7, 0, Math.PI*2, true);

    ctx1.closePath();
    ctx1.fill();

    ctx1.strokeStyle = 'blue';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.strokeText(node.node, node.x-20, node.y-15);

    ctx1.fillStyle = '#E0F574';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.fillText(node.node, node.x-20, node.y-15);


  }
  else
  {
    $('#canvas0').show();
    setFloor(node.loc,"F");

    ctx0.fillStyle = '#5B69EC';
    ctx0.beginPath();
    ctx0.arc(node.x, node.y, 9, 0, Math.PI*2, true);

    ctx0.closePath();
    ctx0.fill();

    ctx0.strokeStyle = 'blue';
    ctx0.font = 'bold 18px sans-serif';
    ctx0.textBaseline = 'bottom';
    ctx0.strokeText(node.node, node.x-20, node.y-15);

    ctx0.fillStyle = '#E0F574';
    ctx0.font = 'bold 18px sans-serif';
    ctx0.textBaseline = 'bottom';
    ctx0.fillText(node.node, node.x-20, node.y-15);

    $('#canvas1').show();
```

```
    node=nodes[getNodeNumber(node.loc.slice(0,-1),nodes)];

    ctx1.fillStyle = '#5B69EC';
    ctx1.beginPath();
    ctx1.arc(node.x, node.y, 9, 0, Math.PI*2, true);

    ctx1.closePath();
    ctx1.fill();

    ctx1.strokeStyle = 'blue';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.strokeText(node.node, node.x-20, node.y-15);

    ctx1.fillStyle = '#E0F574';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.fillText(node.node, node.x-20, node.y-15);
  }

};
```

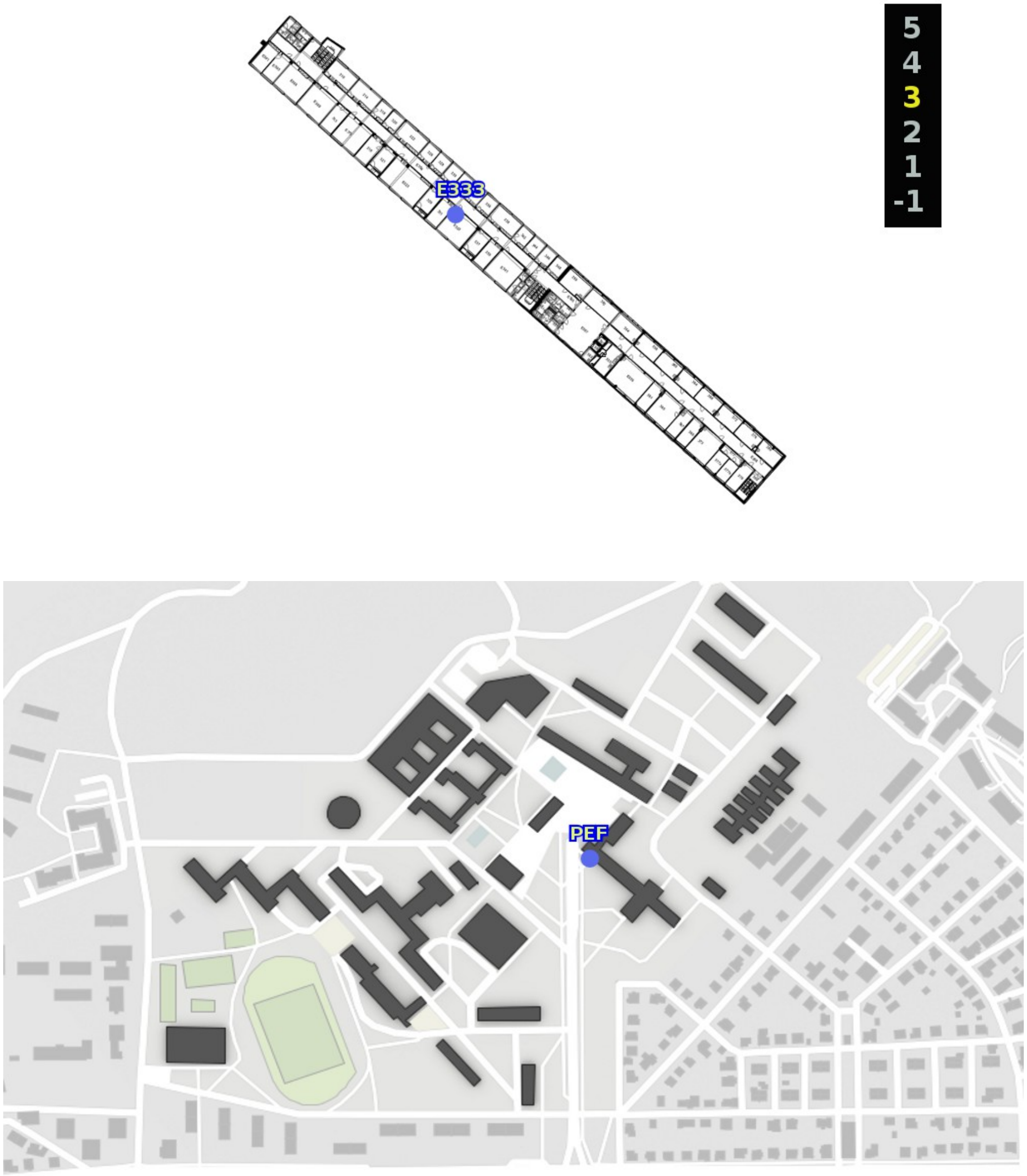


Image 25 - Show E333 location

4.4.9. Web Page Initialization

Because each url contains same code, it is necessary, when a QR Code is scanned, to dynamically show current position on a map. It is done by taking the rest of url just after z/ . It does not remove .html part, because it is done by .htaccess file on a web server. After that the function showLoc is called with parameter equal to the rest of the url. In case that user is on a homepage, the application just displays an empty map and notification that user is at home.

```
title = document.location.href;
ind = title.indexOf('z/');

title = title.slice(ind+2).toUpperCase();

if (title == "INDEX.HTML" || title == "INDEX" || title == "")
{
  title = "HOME";

}
else {

showLoc(title);
}
  document.title = "CZUNAV - " + title;

$('#pos').append(title);
```

4.4.10. Input Fields Auto-fill

When user visits the web site, he is in most of cases confused what to fill in. For these cases is implemented menu which is displayed when user hovers over the field. This is done in combination with CSS, HTML and JavaScript. The JavaScript function is called reply-click and returns the id of clicked element into a particular field.

```
function reply_click(clicked_id)
{

  switch(clicked_id.charAt(0))
{
    case "F":
      $('#from').val("");
      $('#from').val(clicked_id.substring(1));
      break;
    case "T":
      $('#to').val("");
        $('#to').val(clicked_id.substring(1));
      break;
    case "L":
      $('#loc').val("");
            $('#loc').val(clicked_id.substring(1));
      break;
  }
}
```

The code recognizes particular field by a first character of selected id. For example destination JIH in a location field has as a first character L.

```
<li><a id="LJIH" onClick="reply_click(this.id)">JIH</a></li>
<li><a id="LKTV" onClick="reply_click(this.id)">KTV</a></li>
```

4.5. Web Server setting

4.5.1. Domain Settings

For the purposes of this work author registred domain `czunav.cz`. To forward all queries for this domain it is needed to add provider's primary and secondary name servers or adding an A record pointing on the IP address of the server. By visiting this address anyone can download application's source codes.

4.5.2. Configuration of .htaccess

To ease up the process of location displaying author configures Apache .htaccess file. Thanks to these changes if user wants to display location of E333, he can just type `www.czunav.cz/e333` without appending `.html`. To configure this it is needed to edit the RewriteBase section by adding these rules below.

```
# RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteRule ^ index.php [L]
RewriteCond %{REQUEST_FILENAME}\.html -f
RewriteRule ^(.*)$ $1.html
```

4.6. QR Code URL Linking

To bind an url with QR Code author uses http://qrcode.kaywa.com/ website. The inserted url is in format http://www.czunav.cz/<location>. The example of url binded with the E333 room is shown below on the figure. Because it is very important to have a QR Code visible and scannable, author sets the size 60mm x 60mm.

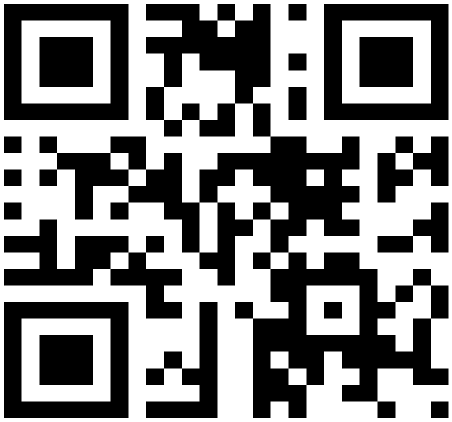


Image 26 - E333 QR Code

5. Conclusion

To recap the goals of this thesis are 1) Find a best way to locate user's position in a campus and inside buildings, 2) Find a most appropriate search algorithm and nodes representation of the campus and building's interiors, 3) To deliver user friendly application which can offer functions to display user's location and to find a route to the desired destination; and 4) To ensure accessibility from most of the modern smartphones.

All the goals of this thesis are met. In Chapter I author analyzes most of the available barcodes. After this the author analyzes search algorithms, lists and matrix representation needed for the nodes representation on the map and in the application. In Chapter I author analyzes web technologies like scripting languages where he decided to use JavaScript as a main language in combination with HTML, CSS. Author examines best server configuration for displaying the website's urls in browsers. In Chapter II the author comments on the programmed application where he states the decisions he made. In this chapter author shows the design of the application and connection between QR Codes and urls. All the steps needed to accomplish the project are displayed in a Gantt's chart at the beginning of the Chapter II and every part of the application is additionally displayed in a Component Diagram of particular function.

The developed application is written in the JavaScript scripting language and is developed with an emphasis on future extendability. Because JavaScript is open standard language, anyone can modify the code. The code is easy and understandable so anyone with basic knowledge of this scripting language can modify it, add nodes, buildings and platforms. The code is written in a way that to add new parts of the CULS campus it needs only minor changes.

Author decided to use QR Codes rather than 1D barcodes because of their storage capabilities. To find the shortest path depth-first search algorithm was used which the author finds to be most suitable for the application’s purposes. Many routes were tested by testers. Thanks to this heuristic process if testers find any route, which was not shortest, additionally nodes were inserted into the appropriate graph. As stated before, in the practical part of this work, the author decided to use JavaScript in combination with HTML , CSS and configuration of .htaccess file.

Comparing to Oris CULS navigation application, the author's application is more user friendly and accessible by modern devices like tablets and smartphones. Because of the fact that most of these devices has an in built camera, which is in most cases possible to use as a barcode reader, scanning a QR Code is the fastest way to determine the user's location. It's faster than GPS and works in buildings as well. The only disadvantage of this application is that it is client-side only. If the user searches a difficult route and if he has a slow processor in his device, it could last few minutes to find it. On the other hand the processor speed increases and prices go down every year. Next good thing is multi-platform availability. It works in every web browser supporting JavaScript so operating system is not a key thing. More so it can be used not only in portable devices, but even in every machine where a modern web browser is installed.

To conclude the developed application satisfies all requirements. As tested on few users, it is very intuitive and user friendly. They do not struggle what to fill in and for them it is very easy to use it. The author hopes that this application will make life in the CULS campus easier for people with a poor sense of direction, new students and visitors.

6. References

1D Barcode Formats. PERETZ, Phil. *NATIONWIDEBARCODE* [online]. 2013 [cited 2012-10-15]. Available from: http://www.nationwidebarcode.com/1d-barcode-format/

Adjacency Matrix. WEISSTEIN, Eric W. *WolframMathWorld* [online]. 2012b [cited 2012-11-28]. Available from: http://mathworld.wolfram.com/AdjacencyMatrix.html

Apache Core Features: AllowOverride Directive. *The Apache Software Foundation* [online]. 2013a [cited 2013-02-27]. Available from: http://httpd.apache.org/docs/current/mod/core.html#allowoverride

Apache HTTP Server Tutorial: .htaccess files. *The Apache Software Foundation* [online]. 2013b [cited 2013-02-27]. Available from: http://httpd.apache.org/docs/2.2/howto/htaccess.html

Barcode. *Wikipedia* [online]. 2012-09-30 [cited 2012-09-30]. Available from: http://en.wikipedia.org/wiki/Barcode

BIBEAULT, Bear a Yehuda KATZ. *JQuery in action*. 2nd ed. Stamford, CT: Manning Publications Co., c2010, p. 35. ISBN 978-1935182320.

Breadth first search and depth first search. *University of California Irvine* [online]. 2004, 2004-05-07 [cited 2012-12-10]. Available from: http://www.ics.uci.edu/~eppstein/161/960215.html

Breadth-first search. *Princeton University* [online]. 2010 [cited 2013-01-20]. Available from: http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Breadth-first_search.html

Comparison between 2D barcode and 1D barcode, CM code and PDF417, QR Codes, date matrix. *SYSCAN* [online]. 2010 [cited 2012-10-02]. Available from: http://www.syscantech.com/en/SyscanCode/Syscan_2dcode.asp

Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers. Can I use [online]. 2013 [cited 2013-03-10]. Available from: http://caniuse.com/#feat=canvas

CORMEN, Thomas H a David Sawyer MCFARLAND. Introduction to algorithms. 3rd ed. Cambridge: MIT Press, c2009, xix, 1292 s. Missing manual. ISBN 978-0-262-03384-8.

COXETER, H. Regular Polytopes. 3rd ed. New York: Dover Publications, 1973, 321 s. Graduate texts in mathematics, 173. ISBN 04-866-1480-8.

Depth-first search. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012b [cited 2013-03-12]. Available from: http://en.wikipedia.org/wiki/Depth-first_search

DIESTEL, Reinhard. *Graph theory*. 3rd ed. Berlin: Springer, c2006, xvi, 410s. Graduate texts in mathematics, 173. ISBN 35-402-6183-4.

Document Object Model (DOM). *W3C* [online]. 2005, 2005-01-19 [cited 2013-01-22]. Available from: http://www.w3.org/DOM/#what

ECMAScript Language Overview. ECMAScript [online]. 2007 [cited 2013-03-27]. Available from: http://www.ecmascript.org/es4/spec/overview.pdf

FindControl: Recursive DFS, BFS, and Leaf to Root Search with Pruning. *Coding Lifestyle* [online]. 2011, 2011-10-24 [cited 2012-12-20]. Available from: http://codinglifestyle.wordpress.com/category/jquery/

FLANAGAN, David a David Sawyer MCFARLAND. JQuery pocket reference. 1. ed. Beijing: O'Reilly, 2010, xvi, 518 p. Missing manual. ISBN 14-493-9722-0.

FLANAGAN, David. JavaScript: the definitive guide. Fifth edition. Sebastopol: O´Reilly, 2006, 994 s. ISBN 05-961-0199-6.

Gantt Charts: Planning and Scheduling More Complex Projects. *MindTools* [online]. 2013 [cited 2013-02-27]. Available from: http://www.mindtools.com/pages/article/newPPM_03.htm

Gantt Charts. *3D Electro Writers* [online]. 2008 [cited 2013-02-27]. Available from: http://nanoelectrospray.wordpress.com/managerial-reports/gantt-charts/

Geronimo Kernel: Overview. Apache GERONIMO [online]. 2009 [cited 2013-02-27]. Available from: https://cwiki.apache.org/GMOxDEV/3-geronimo-kernel.html

GROSS, Jonathan L. *Graph theory and its applications*. 2nd ed. New York: Chapman and Hall, 2006, 779 s. Graduate texts in mathematics, 173. ISBN 15-848-8505-X.

HTML & CSS: What is CSS?. W3C [online]. 2012 [cited 2013-01-28]. Available from: http://www.w3.org/standards/webdesign/htmlcss#whatcss

HTML5 Canvas. *W3C* [online]. 2011 [cited 2013-02-10]. Available from: http://www.w3schools.com/html/html5_canvas.asp

Imaging Glossary. *IPS* [online]. 2012 [cited 2012-09-26]. Available from: http://www.ipsservices.com/resources/glossaries-and-guides/imaging-glossary

Incidence Matrix. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012a, 2012-11-20 [cited 2013-03-26]. Available from: http://en.wikipedia.org/wiki/Incidence_matrix

Incidence Matrix. WEISSTEIN, Eric W. *WolframMathWorld* [online]. 2012a [cited 2012-11-28]. Available from: http://mathworld.wolfram.com/IncidenceMatrix.html

Introduction to HTML 4. *W3C* [online]. 1999 [cited 2013-01-22]. Available from: http://www.w3.org/TR/html401/intro/intro.html

JQuery Introduction. *W3C* [online]. 2010 [cited 2013-02-10]. Available from: http://www.w3schools.com/jquery/jquery_intro.asp

MCFARLAND, David Sawyer a David Sawyer MCFARLAND. JavaScript. 2nd ed. Sebastopol, Calif.: O'Reilly, 2011c2012, xvi, 518 p. Missing manual. ISBN 14-493-9902-9.

QR Code Features. *QRCode.com* [online]. 2010a [cited 2012-09-30]. Available from: http://www.qrcode.com/en/qrfeature.html

QR Code Introduction: Overview. *QRCode.com* [online]. 2010b [cited 2012-09-30]. Available from: http://www.qrcode.com/en/qrgene1.html

QR Code Minimum Size. *QRStuff.com* [online]. 2011b, 2011-01-18 [cited 2012-10-19]. Available from: http://www.qrstuff.com/blog/2011/11/23/qr-code-minimum-size

QR Code Minimum Size. ROUSE, Margaret. *WhatIs.com* [online]. 2005 [cited 2012-11-01]. Available from: http://whatis.techtarget.com/definition/graph-theory

Scripting Language. Webopedia [online]. 2013 [cited 2013-01-28]. Available from: http://www.webopedia.com/TERM/S/scripting_language.html

Should I use 1D or 2D Barcode Format?. *SAGEDATA* [online]. 2012 [cited 2012-10-02]. Available from: http://www.sagedata.com/learning_centre/Should-I-use-1D-or-2D-barcodes.html

The Document Object Model (DOM). SMITH, Daniel. *Daniel Smith's Personal Blog* [online]. 2008, 2008-07-20 [cited 2013-01-22]. Available from: http://dsmith77.wordpress.com/2008/07/20/the-document-object-model-dom/

UML 2 Component Diagram. *Sparx Systems* [online]. 2012 [cited 2013-02-27]. Available from: http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_componentdiagram.html

Using .htaccess rewrite rules. *MT* [online]. 2012 [cited 2013-02-27]. Available from: http://kb.mediatemple.net/questions/85/Using+.htaccess+rewrite+rules#gs

What Size Should A Printed QR Code Be?. *QRStuff.com* [online]. 2011a, 2011-11-23 [cited 2012-10-19]. Available from: http://www.qrstuff.com/blog/2011/01/18/what-size-should-a-qr-code-be

III. Appendix

7. Supplements

7.1. List of tables

7.2. List of images

7.3. .htaccess configuration file

```
# Apache/PHP/Drupal settings:
#

# Protect files and directories from prying eyes.
<FilesMatch "\.(engine|inc|info|install|make|module|profile|test|po|sh|.*sql|theme|
tpl(\.php)?|xtmpl)$|^(\..*|Entries.*|Repository|Root|Tag|Template)$">
  Order allow,deny
</FilesMatch>

# Don't show directory listings for URLs which map to a directory.
Options -Indexes

# Follow symbolic links in this directory.
Options +FollowSymLinks

# Make Drupal handle any 404 errors.
ErrorDocument 404 /index.php

# Force simple error message for requests for non-existent favicon.ico.
<Files favicon.ico>
  # There is no end quote below, for compatibility with Apache 1.3.
  ErrorDocument 404 "The requested file favicon.ico was not found.
</Files>

# Set the default handler.
DirectoryIndex index.php index.html index.htm

# PHP 5, Apache 1 and 2.
```

```
<IfModule mod_php5.c>
  php_flag magic_quotes_gpc                 off
  php_flag magic_quotes_sybase              off
  php_flag register_globals                 off
  php_flag session.auto_start               off
  php_value mbstring.http_input             pass
  php_value mbstring.http_output            pass
  php_flag mbstring.encoding_translation    off
</IfModule>

# Requires mod_expires to be enabled.
<IfModule mod_expires.c>
  # Enable expirations.
  ExpiresActive On

  # Cache all files for 2 weeks after access (A).
  ExpiresDefault A1209600

  <FilesMatch \.php$>

    ExpiresActive Off
  </FilesMatch>
</IfModule>

# Various rewrite rules.
<IfModule mod_rewrite.c>
  RewriteEngine on

  RewriteRule "(^|/)\." - [F]

  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_URI} !=/favicon.ico
  RewriteCond %{REQUEST_FILENAME}\.html -f
  RewriteRule ^(.*)$ $1.html


  <IfModule mod_headers.c>

    RewriteCond %{HTTP:Accept-encoding} gzip
    RewriteCond %{REQUEST_FILENAME}\.gz -s
    RewriteRule ^(.*)\.css $1\.css\.gz [QSA]

    RewriteCond %{HTTP:Accept-encoding} gzip
    RewriteCond %{REQUEST_FILENAME}\.gz -s
    RewriteRule ^(.*)\.js $1\.js\.gz [QSA]

    # Serve correct content types, and prevent mod_deflate double gzip.
    RewriteRule \.css\.gz$ - [T=text/css,E=no-gzip:1]
    RewriteRule \.js\.gz$ - [T=text/javascript,E=no-gzip:1]

    <FilesMatch "(\.js\.gz|\.css\.gz)$">
      # Serve correct encoding type.
      Header append Content-Encoding gzip
      # Force proxies to cache gzipped & non-gzipped css/js files separately.
      Header append Vary Accept-Encoding
    </FilesMatch>
  </IfModule>
</IfModule>
```

7.4. Application.js

```
$(document).ready(function(){



var ctx0 = $('#canvas0')[0].getContext("2d");
var ctx1 = $('#canvas1')[0].getContext("2d");
var ctx2 = $('#canvas2')[0].getContext("2d");
var ctxG1 = $('#canvasG1')[0].getContext("2d");
var ctxG2 = $('#canvasG2')[0].getContext("2d");

function clearCanvas() {
ctx0.clearRect(0, 0, 1000, 600);
ctx1.clearRect(0, 0, 1000, 600);
ctx2.clearRect(0, 0, 1000, 600);
ctxG1.clearRect(0, 0, 1000, 600);
ctxG2.clearRect(0, 0, 1000, 600);
};


$('#route').click(function() {
  init();
  $('#position').hide();
  $('#canvas1').hide();
  $('#main').hide();
  $('#text').hide();
  $('#loading').show();
  $('#greenDivD').hide();
  //clearCanvas();

  var from = [document.getElementById('from').value.toUpperCase()];

  var to = [document.getElementById('to').value.toUpperCase()];


  fromNode = getNodeNumber(from,nodes);
  toNode = getNodeNumber(to,nodes);
 if (!fromNode || !toNode)
  {
    init();
    return;
  }
  parent.push(from);

  path.push(from);

  findRoute(from,to);


 setFloor(nodes[fromNode].loc,"F");
 setFloor(nodes[toNode].loc,"T");



   $('#loading').hide();
   $('#main').show();
  draw();
document.getElementById("main").style.maxHeight = "100%";


});


function setFloor(node,x)
{
 if (x == "F") {
  a="canvas0";
 }
 else if (x == "T") {
  a="canvas2";
 }

 switch (node)
```

```
  {

  case "PEF1":
    document.getElementById(a).style.background = "url(map/1np.jpg)";
    break;
  case "PEF2":
    document.getElementById(a).style.background = "url(map/2np.jpg)";
    break;
  case "PEF3":
    document.getElementById(a).style.background = "url(map/3np.jpg)";
    break;
  case "PEF4":
    document.getElementById(a).style.background = "url(map/4np.jpg)";
    break;
 };


};


function draw() {

  var path0 = [];
  var path1 = [];
  var path2 = [];
  var pathG1 = [];
  var pathG2 = [];

  firstDest = nodes[getNodeNumber(path[0],nodes)];
  finalDest = nodes[getNodeNumber(path.slice(-1)[0],nodes)];

  for (i in path)
  {
    u=path[i];
    kvik = nodes[getNodeNumber(path[i],nodes)].loc;
    if (kvik == "OUT")
    {
      path1.push(u);
    }
    else if (kvik == firstDest.loc)
    {
      path0.push(u);
    }
    else if (kvik == finalDest.loc) {
      path2.push(u);
    }
  }

  for (i in path)
  {
    u=path[i];

    kvik = nodes[getNodeNumber(path[i],nodes)].loc;
    if (kvik == "PEF1")
      {
      if(path0.length == 0) {
        pathG2.push(u);
        document.getElementById("canvasG2").style.background = "url(map/1np.jpg)";


      }
      else
      {
        pathG1.push(u);
        document.getElementById("canvasG1").style.background = "url(map/1np.jpg)";
```

```
        }
        }
    }

    if(path0.length != 0)
    {
    node1=nodes[getNodeNumber(path0[0],nodes)];
    node2=nodes[getNodeNumber(path0.slice(-1)[0],nodes)];
    $('#canvas0').show();
    drawOut(ctx0,node1,node2,path0);
    }

    if(path1.length != 0)
    {
    node3=nodes[getNodeNumber(path1[0],nodes)];
    node4=nodes[getNodeNumber(path1.slice(-1)[0],nodes)];
    $('#canvas1').show();
    drawOut(ctx1,node3,node4,path1);
    }
    if(path2.length != 0)
    {
    node5=nodes[getNodeNumber(path2[0],nodes)];
    node6=nodes[getNodeNumber(path2.slice(-1)[0],nodes)];
    $('#canvas2').show();
    drawOut(ctx2,node5,node6,path2);
    }
    if(pathG2.length != 0)
    {
    node7=nodes[getNodeNumber(pathG2[0],nodes)];
    node8=nodes[getNodeNumber(pathG2.slice(-1)[0],nodes)];
    $('#canvasG2').show();
    drawOut(ctxG2,node7,node8,pathG2);
    }
    if(pathG1.length != 0)
```

```
{
node9=nodes[getNodeNumber(pathG1[0],nodes)];
node10=nodes[getNodeNumber(pathG1.slice(-1)[0],nodes)];
$('#canvasG1').show();
drawOut(ctxG1,node9,node10,pathG1);
}
  if (firstDest.loc == "OUT")
{
  ctx1.strokeStyle = 'blue';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.strokeText(firstDest.node, firstDest.x-20, firstDest.y-15);

  ctx1.fillStyle = '#E0F574';
  ctx1.font = 'bold 18px sans-serif';
  ctx1.textBaseline = 'bottom';
  ctx1.fillText(firstDest.node, firstDest.x-20, firstDest.y-15);
}
else
{
  ctx0.strokeStyle = 'blue';
  ctx0.font = 'bold 18px sans-serif';
  ctx0.textBaseline = 'bottom';
  ctx0.strokeText(firstDest.node, firstDest.x-20, firstDest.y-15);

  ctx0.fillStyle = '#E0F574';
  ctx0.font = 'bold 18px sans-serif';
  ctx0.textBaseline = 'bottom';
  ctx0.fillText(firstDest.node, firstDest.x-20, firstDest.y-15);
}

if (finalDest.loc == "OUT")
{
    ctx1.strokeStyle = 'blue';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.strokeText(finalDest.node, finalDest.x-20, finalDest.y-15);


    ctx1.fillStyle = '#E0F574';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.fillText(finalDest.node, finalDest.x-20, finalDest.y-15);


  }
  else
  {

    ctx2.strokeStyle = 'blue';
    ctx2.font = 'bold 18px sans-serif';
    ctx2.textBaseline = 'bottom';
    ctx2.strokeText(finalDest.node, finalDest.x-20, finalDest.y-15);

    ctx2.fillStyle = '#E0F574';
    ctx2.font = 'bold 18px sans-serif';
    ctx2.textBaseline = 'bottom';
    ctx2.fillText(finalDest.node, finalDest.x-20, finalDest.y-15);



  }
  if (firstDest.loc.charAt(firstDest.loc.length-1 ) == 1 ||
finalDest.loc.charAt(finalDest.loc.length-1 )==1)
  {
    $('#canvasG1').hide();
    $('#canvasG2').hide();
  }

};


  function drawOut(where,nodea,nodeb,path) {


    where.fillStyle = '#5B69EC';
    where.beginPath();
    where.arc(nodea.x, nodea.y, 7, 0, Math.PI*2, true);

    where.closePath();
    where.fill();

    where.fillStyle = '#E0F574';
    where.beginPath();
    where.arc(nodeb.x, nodeb.y, 9, 0, Math.PI*2, true);

    where.closePath();
    where.fill();

    where.lineWidth = 4;
    where.strokeStyle = '#5B69EC';
    where.beginPath();
    node1=nodes[getNodeNumber(path[0],nodes)];
    where.moveTo(nodea.x,nodea.y);
    for (i in path) {

      node=nodes[getNodeNumber(path[i],nodes)];
      where.lineTo(node.x,node.y);
      where.lineTo(node.x,node.y);
      where.stroke();
    }

  }

function inArray(object,array)
{
  var flag = false;
  for (i in array)
  {
    if (array[i]==object)
    {
      flag = true;
    }
  }
  return flag;
};

function findRouteBackwards(to,y){

  knik = nodes[getNodeNumber(to,nodes)].to;

  for (z in knik)
  {

      if (inArray(knik[z],path[y]))
      {
      path[y]=knik[z];

      }
  }


  to=path[y];
  if (y > hops)

  {
    path.reverse();

    return true;
  };
  y=y+1;

  findRouteBackwards(to,y);


};


function findRoute(from,to) {

  if (inArray(to,from))
  {

    length.reverse();

    path.reverse();
    path[0]=to;
          // alert(path);
    findRouteBackwards(to,1);

  }
  else
  {
    $('#loading').append(".");
    for (i in parent)
    {
     arr1 = nodes[getNodeNumber(parent[i],nodes)].to;
     for (y in arr1)
     {
       if (inArray(arr1[y],path)==false && (inArray(arr1[y],child))==false) {

       child.push(arr1[y]);
       }

     }
    }
    length.push(child.length);
    path.push(child.slice());


    parent = child.slice();

    parent = child;

    child=[];

    hops++;

    findRoute(parent,to);

  }

};


$('#location').click(function() {
  showLoc(document.getElementById('loc').value.toUpperCase());
});
```

```
function showLoc(input){
  init();

  $('#canvas1').hide();


  node=nodes[getNodeNumber(input,nodes)];


  if(node.loc == "OUT")
  {
    $('#canvas1').show();

    ctx1.fillStyle = '#5B69EC';
    ctx1.beginPath();
    ctx1.arc(node.x, node.y, 7, 0, Math.PI*2, true);

    ctx1.closePath();
    ctx1.fill();

    ctx1.strokeStyle = 'blue';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.strokeText(node.node, node.x-20, node.y-15);

    ctx1.fillStyle = '#E0F574';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.fillText(node.node, node.x-20, node.y-15);


  }
  else
  {
```

```
$('#canvas0').show();
setFloor(node.loc,"F");

ctx0.fillStyle = '#5B69EC';
ctx0.beginPath();
ctx0.arc(node.x, node.y, 9, 0, Math.PI*2, true);

ctx0.closePath();
ctx0.fill();

ctx0.strokeStyle = 'blue';
ctx0.font = 'bold 18px sans-serif';
ctx0.textBaseline = 'bottom';
ctx0.strokeText(node.node, node.x-20, node.y-15);

ctx0.fillStyle = '#E0F574';
ctx0.font = 'bold 18px sans-serif';
ctx0.textBaseline = 'bottom';
ctx0.fillText(node.node, node.x-20, node.y-15);


$('#canvas1').show();
node=nodes[getNodeNumber(node.loc.slice(0,-1),nodes)];

ctx1.fillStyle = '#5B69EC';
ctx1.beginPath();
ctx1.arc(node.x, node.y, 9, 0, Math.PI*2, true);


ctx1.closePath();
ctx1.fill();

ctx1.strokeStyle = 'blue';
ctx1.font = 'bold 18px sans-serif';
```

```
    ctx1.textBaseline = 'bottom';
    ctx1.strokeText(node.node, node.x-20, node.y-15);

    ctx1.fillStyle = '#E0F574';
    ctx1.font = 'bold 18px sans-serif';
    ctx1.textBaseline = 'bottom';
    ctx1.fillText(node.node, node.x-20, node.y-15);
  }

};


function init() {
  $('#position').show();
  $('#newsearch').hide();
  $('#mistake').hide(); //HIDE MISTAKE FINDER
  $('#canvas0').hide();
  $('#canvas1').show();
  $('#canvas2').hide();
  $('#canvasG1').hide();
  $('#canvasG2').hide();
  $('#loading').hide();
  $('#main').show();
  $('#text').show();

  path = [];
  child = [];
  parent = [];
  length = [];
  hops = 0;
  document.getElementById("canvas1").style.background = "url(map/mapa1.jpg)";
  clearCanvas();

  };
```

```
init();

$('#mistake').click(function () {


  for (i in nodes)
  {

    numberOfFindings=0;
    for (z in nodes)
    {
      for (y in nodes[z].to)
      {


        if (nodes[i].node == nodes[z].to[y])
        {

          numberOfFindings++;
        }
      }
    }

    if (numberOfFindings != nodes[i].to.length)
    {
    alert (nodes[i].node+" leads to "+nodes[i].to+" has been found
"+numberOfFindings+ " times ");
    }
    else
    {

    }
  }
  alert("Looks ok!");
});




function getNodeNumber(name,array) {
  for (i in array)
  {
    if (array[i].node==name)
    {
     return i;
    }


  }
 alert(name+" does not exist!");
 return false;

};

var nodes = [
//Faculty of Agrobiology, Food and Natural Renodes[from].nodes
{node:"FAPPZ",loc:"OUT",to:[10100],x:604,y:207},
{node:"AFA",loc:"OUT",to:[10099,"AFB"],x:646,y:206},
{node:"AFB",loc:"OUT",to:[10107,"AFA"],x:656,y:188},
{node:"AFC",loc:"OUT",to:[10129],x:597,y:126},

//Faculty of Engineering
{node:"MI",loc:"OUT",to:[10065],x:442,y:251},
{node:"MII",loc:"OUT",to:[10088],x:469,y:221},
{node:"MIII",loc:"OUT",to:[10090],x:495,y:192},

//Kruhova hala
```

```
{node:"RK",loc:"OUT",to:[10061],x:334,y:249},

//Katedra Jazyku
{node:"KJ",loc:"OUT",to:[10145],x:705,y:317},




//Faculty of Environmental Science
{node:"FZP",loc:"OUT",to:[10113],x:710,y:100},

//KTV
{node:"KTV",loc:"OUT",to:[10011],x:193,y:485},


//REKTORAt
{node:"REKTORAT",loc:"OUT",to:[10094,10093,10096],x:543,y:231},

//SIC
{node:"SIC",loc:"OUT",to:[10121],x:489,y:124},

//Koleje JIH
{node:"JIH",loc:"OUT",to:[10029],x:382,y:417},

//BUNKOVISTE
{node:"BUN",loc:"OUT",to:[10125],x:761,y:128},

//OTHER
{node:"FARMA",loc:"OUT",to:[10038,10068],x:503,y:376},
{node:"STABLE",loc:"OUT",to:[10117,10120],x:701,y:18},
{node:"AULA",loc:"OUT",to:[10070],x:503,y:303},
{node:"MENSA",loc:"OUT",to:[10067],x:494,y:342},
//Routes
{node:10001,loc:"OUT",to:[10002],x:137,y:586},
{node:10002,loc:"OUT",to:[10001,10007],x:194,y:585},
{node:10003,loc:"OUT",to:[10021],x:528,y:585},
{node:10004,loc:"OUT",to:[10005],x:576,y:585},
{node:10005,loc:"OUT",to:[10004,10022],x:567,y:562},
{node:10006,loc:"OUT",to:[10008,10010,10015],x:276,y:534},
{node:10007,loc:"OUT",to:[10002,10008,10009],x:193,y:515},
{node:10008,loc:"OUT",to:[10006,10007,10010],x:199,y:516},
{node:10009,loc:"OUT",to:[10007,10011],x:193,y:502},
{node:10010,loc:"OUT",to:[10012,10008,10006,10013],x:225,y:490},
{node:10011,loc:"OUT",to:[10009,"KTV",10012],x:193,y:490},
{node:10012,loc:"OUT",to:[10010,10011],x:209,y:490},
{node:10013,loc:"OUT",to:[10010,10014],x:232,y:482},
{node:10014,loc:"OUT",to:[10013,10025],x:228,y:463},
{node:10015,loc:"OUT",to:[10006,10016,10017],x:319,y:534},
{node:10016,loc:"OUT",to:[10015,10017,10136],x:367,y:534},
{node:10017,loc:"OUT",to:[10015,10016,10018],x:360,y:494},
{node:10018,loc:"OUT",to:[10017,10138,10031],x:358,y:470},
{node:10019,loc:"OUT",to:[10131,10020,10021],x:504,y:534},
{node:10020,loc:"OUT",to:[10019,10032,10133],x:504,y:520},
{node:10021,loc:"OUT",to:[10003,10019,10022,10024],x:528,y:534},
{node:10022,loc:"OUT",to:[10005,10021,10023],x:567,y:534},
{node:10023,loc:"OUT",to:[10022,10042],x:567,y:489},
{node:10024,loc:"OUT",to:[10021,10041],x:528,y:488},
{node:10025,loc:"OUT",to:[10014,10026],x:226,y:418},
{node:10026,loc:"OUT",to:[10025,10027,10050],x:249,y:367},
{node:10027,loc:"OUT",to:[10052,10056,10026],x:288,y:353},
{node:10028,loc:"OUT",to:[10139,10056,10057],x:323,y:378},
{node:10029,loc:"OUT",to:[10147,10030,"JIH"],x:398,y:404},
{node:10030,loc:"OUT",to:[10029,10032,10035],x:418,y:424},
{node:10031,loc:"OUT",to:[10018,10032],x:411,y:459},
{node:10032,loc:"OUT",to:[10031,10033,10020,10030],x:445,y:453},
{node:10033,loc:"OUT",to:[10032,10040,10034],x:459,y:451},
{node:10034,loc:"OUT",to:[10033,10035,10038],x:458,y:425},
{node:10035,loc:"OUT",to:[10030,10034,10037,10036],x:438,y:405},
{node:10036,loc:"OUT",to:[10037,10035],x:411,y:377},
{node:10037,loc:"OUT",to:[10035,10036,10038,10063],x:438,y:359},
{node:10038,loc:"OUT",to:[10034,10037,"FARMA"],x:484,y:396},
{node:10039,loc:"OUT",to:[10099,10098],x:640,y:233},
{node:10040,loc:"OUT",to:[10033,10127,10135],x:504,y:451},
{node:10041,loc:"OUT",to:[10024,10043],x:548,y:459},
{node:10042,loc:"OUT",to:[10023,10043,10143],x:567,y:450},
{node:10043,loc:"OUT",to:[10127,10041,10042,10144],x:548,y:450},
{node:10044,loc:"OUT",to:[10141,10045,10046,10047],x:567,y:388},
{node:10045,loc:"OUT",to:[10142,10044,10046,10068,10069],x:548,y:371},
{node:10046,loc:"OUT",to:[10044,10045,10047,10071],x:567,y:363},
{node:10047,loc:"OUT",to:[10044,10046,10048,10072],x:584,y:367},
{node:10048,loc:"OUT",to:[10047,10049],x:612,y:374},
{node:10049,loc:"OUT",to:[10048,10074],x:625,y:384},
{node:10050,loc:"OUT",to:[10026,10051],x:248,y:349},
{node:10051,loc:"OUT",to:[10050],x:224,y:302},
{node:10052,loc:"OUT",to:[10027],x:265,y:330},
{node:10053,loc:"OUT",to:[10055],x:236,y:283},
{node:10054,loc:"OUT",to:[10059,10058],x:287,y:293},
{node:10055,loc:"OUT",to:[10053,10059],x:254,y:266},
{node:10056,loc:"OUT",to:[10027,10028,10057],x:304,y:350},
{node:10057,loc:"OUT",to:[10056,10028,10146,10058],x:330,y:326},
{node:10058,loc:"OUT",to:[10054,10057,10060],x:309,y:303},
{node:10059,loc:"OUT",to:[10055,10054,10061],x:313,y:266},
{node:10060,loc:"OUT",to:[10058,10061],x:330,y:286},
{node:10061,loc:"OUT",to:[10060,10059,10062,"RK"],x:334,y:266},
{node:10062,loc:"OUT",to:[10061,10064],x:358,y:266},
{node:10063,loc:"OUT",to:[10037,10066],x:438,y:353},
{node:10064,loc:"OUT",to:[10062,10065,10122],x:436,y:266},
{node:10065,loc:"OUT",to:[10064,10085,"MI"],x:447,y:254},
{node:10066,loc:"OUT",to:[10130,10063,10067],x:481,y:308},
{node:10067,loc:"OUT",to:[10066,10068,10070,"MENSA"],x:504,y:330},
{node:10068,loc:"OUT",to:[10067,10045,"FARMA"],x:526,y:350},
{node:10069,loc:"OUT",to:[10045,10140,10071,10079],x:548,y:347},
{node:10070,loc:"OUT",to:[10067,10140,10080,"AULA"],x:510,y:315},
{node:10071,loc:"OUT",to:[10046,10069,10073,10079,10082],x:567,y:332},
{node:10072,loc:"OUT",to:[10073,10047,"NEW PEF"],x:607,y:341},
{node:10073,loc:"OUT",to:[10071,10072],x:596,y:331},
{node:10074,loc:"OUT",to:[10049,10075,"P1037"],x:661,y:345},
{node:10075,loc:"OUT",to:[10074,10076,10145],x:682,y:322},
{node:10076,loc:"OUT",to:[10077,10078,10075],x:644,y:283},
{node:10077,loc:"OUT",to:[10076,"P1035"],x:623,y:306},
{node:10078,loc:"OUT",to:[10098,10076],x:633,y:266},
{node:10079,loc:"OUT",to:[10069,10071,10082,10126,10080],x:548,y:307},
{node:10080,loc:"OUT",to:[10070,10081,10084,10093,10079],x:523,y:287},
{node:10081,loc:"OUT",to:[10086,10080,10085,10122,10130],x:494,y:266},
{node:10082,loc:"OUT",to:[10071,10079,10083,"PEF"],x:567,y:289},
{node:10083,loc:"OUT",to:[10082,10095,"PEF"],x:563,y:272},
{node:10084,loc:"OUT",to:[10087,10080,10089,10086],x:508,y:250},
{node:10085,loc:"OUT",to:[10087,10065,10081],x:462,y:238},
{node:10086,loc:"OUT",to:[10081,10084,10093],x:522,y:258},
{node:10087,loc:"OUT",to:[10084,10085,10088],x:470,y:229},
{node:10088,loc:"OUT",to:[10087,10090,"MII"],x:474,y:224},
{node:10089,loc:"OUT",to:[10084,10092],x:503,y:220},
{node:10090,loc:"OUT",to:[10088,10092,"MIII"],x:499,y:195},
{node:10091,loc:"OUT",to:[10092,10096],x:538,y:210},
{node:10092,loc:"OUT",to:[10089,10090,10091,10103],x:506,y:188},
{node:10093,loc:"OUT",to:[10080,10086,10094,"REKTORAT"],x:538,y:252},
{node:10094,loc:"OUT",to:[10126,10093,10095,"REKTORAT",10096],x:547,y:253},
{node:10095,loc:"OUT",to:[10083,10094,10096,10097],x:564,y:254},
{node:10096,loc:"OUT",to:[10091,10097,10095,10094,"REKTORAT"],x:559,y:226},
{node:10097,loc:"OUT",to:[10095,10096,10100],x:585,y:230},
{node:10098,loc:"OUT",to:[10078,10039,10124],x:649,y:240},
{node:10099,loc:"OUT",to:[10039,10100,"AFA"],x:632,y:227},
{node:10100,loc:"OUT",to:[10097,10099,10101,"FAPPZ"],x:602,y:210},
{node:10101,loc:"OUT",to:[10100,10102],x:567,y:193},
{node:10102,loc:"OUT",to:[10101,10103],x:534,y:170},
{node:10103,loc:"OUT",to:[10092,10102,10121],x:509,y:173},
{node:10104,loc:"OUT",to:[10105,10121],x:529,y:137},
{node:10105,loc:"OUT",to:[10104,10128],x:575,y:129},
{node:10106,loc:"OUT",to:[10128,10107,10111],x:606,y:148},
{node:10107,loc:"OUT",to:[10106,10110,10108,"AFB"],x:659,y:182},
{node:10108,loc:"OUT",to:[10123,10107,10109],x:688,y:201},
{node:10109,loc:"OUT",to:[10108,10110,10112],x:711,y:175},
{node:10110,loc:"OUT",to:[10107,10109,10111,10113],x:672,y:144},
{node:10111,loc:"OUT",to:[10106,10110,10114],x:628,y:112},
{node:10112,loc:"OUT",to:[10109,10113,10125],x:739,y:142},
{node:10113,loc:"OUT",to:[10110,10112,10114,"FZP"],x:700,y:110},
{node:10114,loc:"OUT",to:[10111,10113,10115],x:653,y:070},
{node:10115,loc:"OUT",to:[10114,10116,10117],x:674,y:34},
{node:10116,loc:"OUT",to:[10115,10118],x:722,y:078},
{node:10117,loc:"OUT",to:[10115,10120,"STABLE"],x:688,y:009},
{node:10118,loc:"OUT",to:[10116,10119,10120],x:753,y:043},
{node:10119,loc:"OUT",to:[10118,10125],x:794,y:079},
{node:10120,loc:"OUT",to:[10117,10118,"STABLE"],x:704,y:002},
{node:10121,loc:"OUT",to:[10104,10103,"SIC"],x:520,y:151},
{node:10122,loc:"OUT",to:[10081,10130,10064],x:453,y:282},
{node:10123,loc:"OUT",to:[10108,10124],x:680,y:209},
{node:10124,loc:"OUT",to:[10098,10123],x:664,y:226},
{node:10125,loc:"OUT",to:[10112,10119,"BUN"],x:755,y:124},
{node:10126,loc:"OUT",to:[10094,10079],x:547,y:277},
{node:10127,loc:"OUT",to:[10040,10043],x:527,y:451},
{node:10128,loc:"OUT",to:[10105,10106,10129],x:589,y:138},
{node:10129,loc:"OUT",to:[10128,"AFC"],x:594,y:130},
{node:10130,loc:"OUT",to:[10081,10122,10066],x:466,y:295},
{node:10131,loc:"OUT",to:[10019,10137],x:449,y:534},
{node:10132,loc:"OUT",to:[10136,10137],x:410,y:534},
{node:10133,loc:"OUT",to:[10020,10134],x:504,y:500},
{node:10134,loc:"OUT",to:[10133,10135],x:504,y:480},
{node:10135,loc:"OUT",to:[10040,10134],x:504,y:464},
{node:10136,loc:"OUT",to:[10016,10132],x:385,y:534},
{node:10137,loc:"OUT",to:[10132,10131],x:429,y:534},
{node:10138,loc:"OUT",to:[10139,10018],x:346,y:436},
{node:10139,loc:"OUT",to:[10138,10028],x:333,y:404},
{node:10140,loc:"OUT",to:[10069,10070],x:519,y:324},
{node:10141,loc:"OUT",to:[10143,10044],x:567,y:402},
{node:10142,loc:"OUT",to:[10045,10144],x:548,y:397},
{node:10143,loc:"OUT",to:[10042,10141],x:567,y:423},
{node:10144,loc:"OUT",to:[10142,10043],x:548,y:420},
{node:10145,loc:"OUT",to:[10075,"KJ"],x:694,y:331},
{node:10146,loc:"OUT",to:[10057,10147],x:356,y:356},
{node:10147,loc:"OUT",to:[10029,10146],x:375,y:377},
//Faculty of Economics and Management
{node:"PEF",loc:"OUT",to:[10082,10083,"P1034"],x:575,y:280},
{node:"NEW PEF",loc:"OUT",to:[10072,"P1036"],x:610,y:338},
{node:"INFOCENTRUM",loc:"PEF1",to:["P1007"],x:345,y:212},
{node:"EI",loc:"PEF1",to:["P1003"],x:384,y:151},
{node:"EII",loc:"PEF1",to:["P1004"],x:418,y:113},
{node:"EIII",loc:"PEF1",to:["P1005"],x:451,y:74},
{node:"EIV",loc:"PEF1",to:["P1013"],x:568,y:389},
{node:"EV",loc:"PEF1",to:["P1014"],x:576,y:394},
{node:"E115",loc:"PEF1",to:["P1008"],x:359,y:251},
{node:"E119",loc:"PEF1",to:["P1009"],x:380,y:269},
{node:"E223",loc:"PEF2",to:["P2009"],x:314,y:203},
{node:"E239",loc:"PEF2",to:["P2010"],x:410,y:286},
{node:"E255",loc:"PEF2",to:["P2016"],x:549,y:404},
{node:"E309",loc:"PEF3",to:["P3007"],x:302,y:140},
{node:"E315",loc:"PEF3",to:["P3008"],x:334,y:166},
{node:"E323",loc:"PEF3",to:["P3009"],x:382,y:208},
{node:"E333",loc:"PEF3",to:["P3010"],x:442,y:261},
{node:"E341",loc:"PEF3",to:["P3011"],x:489,y:304},
{node:"E355",loc:"PEF3",to:["P3014"],x:616,y:411},
{node:"E409",loc:"PEF4",to:["P4004"],x:284,y:115},
{node:"E423",loc:"PEF4",to:["P4005"],x:368,y:190},
{node:"E431",loc:"PEF4",to:["P4006"],x:419,y:235},
{node:"E439",loc:"PEF4",to:["P4007"],x:470,y:282},
{node:"E455",loc:"PEF4",to:["P4009"],x:619,y:413},
{node:"C11",loc:"PEF1",to:["P1025"],x:511,y:434},
{node:"C12",loc:"PEF1",to:["P1029"],x:520,y:478},
{node:"C13",loc:"PEF1",to:["P1030"],x:482,y:466},
{node:"C14",loc:"PEF1",to:["P1032"],x:509,y:490},
{node:"C21",loc:"PEF2",to:["P2019"],x:455,y:390},
{node:"C22",loc:"PEF2",to:["P2020"],x:467,y:446},
{node:"C23",loc:"PEF2",to:["P2023"],x:420,y:430},
{node:"C24",loc:"PEF2",to:["P2021"],x:454,y:461},



//PEF1
{node:"P1001",loc:"PEF1",to:["P2001","P3001","P4001","P1006"],x:340,y:190},
{node:"P1002",loc:"PEF1",to:["P1010","P2002","P3002","P4002"],x:510,y:393},

{node:"P1003",loc:"PEF1",to:["P1011","P1004","EI"],x:374,y:140},
{node:"P1004",loc:"PEF1",to:["P1003","P1005","EII"],x:406,y:102},
{node:"P1005",loc:"PEF1",to:["P1004","EIII"],x:439,y:64},
{node:"P1006",loc:"PEF1",to:["P1034","P1011","P1007","P1001"],x:324,y:207},
{node:"P1007",loc:"PEF1",to:["P1006","P1008","INFOCENTRUM"],x:337,y:218},
{node:"P1008",loc:"PEF1",to:["P1007","P1009","E115"],x:366,y:243},
{node:"P1009",loc:"PEF1",to:["P1008","P1010","E119"],x:386,y:261},
{node:"P1010",loc:"PEF1",to:["P1035","P1002","P1012","P1009"],x:523,y:379},
{node:"P1011",loc:"PEF1",to:["P1006","P1003"],x:324,y:196},
{node:"P1012",loc:"PEF1",to:["P1013","P1010","P1024","P1030"],x:550,y:404},
{node:"P1013",loc:"PEF1",to:["P1012","P1014","EIV"],x:554,y:406},
{node:"P1014",loc:"PEF1",to:["P1013","P1015","EV"],x:560,y:412},
{node:"P1015",loc:"PEF1",to:["P1014","P1016","P1022"],x:564,y:414},
{node:"P1016",loc:"PEF1",to:["P1015","P1017"],x:587,y:435},
{node:"P1017",loc:"PEF1",to:["P1016","P1018"],x:607,y:452},
{node:"P1018",loc:"PEF1",to:["P1017","P1019"],x:629,y:471},
{node:"P1019",loc:"PEF1",to:["P1018","P1020"],x:650,y:490},
{node:"P1020",loc:"PEF1",to:["P1019","P1021"],x:674,y:509},
{node:"P1021",loc:"PEF1",to:["P1020","P1037"],x:691,y:524},
{node:"P1022",loc:"PEF1",to:["P1015","P1023","P1029"],x:537,y:444},
{node:"P1023",loc:"PEF1",to:["P1022","P1024","P1026"],x:528,y:436},
{node:"P1024",loc:"PEF1",to:["P1012","P1023","P1025"],x:524,y:432},
{node:"P1025",loc:"PEF1",to:["P1024","P1030","C11"],x:518,y:440},
{node:"P1026",loc:"PEF1",to:["P1027","P1023","P2003"],x:514,y:454},
{node:"P1027",loc:"PEF1",to:["P1026","P1028","P2004"],x:516,y:456},
{node:"P1028",loc:"PEF1",to:["P1027","P1031","P2005"],x:518,y:458},
{node:"P1029",loc:"PEF1",to:["P1032","P1022","C12"],x:513,y:472},
{node:"P1030",loc:"PEF1",to:["P1031","P1025","C13","P1012"],x:489,y:473},
{node:"P1031",loc:"PEF1",to:["P1030","P1032","P1028","P1033"],x:495,y:480},
{node:"P1032",loc:"PEF1",to:["P1029","P1031","C14"],x:502,y:484},
{node:"P1033",loc:"PEF1",to:["P1031","P1036"],x:473,y:504},
{node:"P1034",loc:"PEF1",to:["P1006","PEF"],x:304,y:230},
{node:"P1035",loc:"PEF1",to:["P1010",10077],x:536,y:364},
{node:"P1036",loc:"PEF1",to:["P1033","NEW PEF"],x:465,y:514},
{node:"P1037",loc:"PEF1",to:["P1021",10074],x:702,y:534},

//PEF2
{node:"P2001",loc:"PEF2",to:["P2008","P1001","P3001","P4001"],x:245,y:092},
{node:"P2002",loc:"PEF2",to:["P1002","P2011","P3002","P4002"],x:452,y:333},
{node:"P2003",loc:"PEF2",to:["P1026","P2004","P2022"],x:460,y:414},
{node:"P2004",loc:"PEF2",to:["P1027","P2003","P2005"],x:463,y:417},
{node:"P2005",loc:"PEF2",to:["P1028","P2004","P2018"],x:466,y:420},
//{node:"P2006",loc:"PEF2",to:["P2007","P3001"],x:245,y:084},
//{node:"P2007",loc:"PEF2",to:["P2006","P2008"],x:225,y:106},
{node:"P2008",loc:"PEF2",to:["P2009","P2001"],x:229,y:108},
{node:"P2009",loc:"PEF2",to:["P2010","P2008","E223"],x:324,y:192},
{node:"P2010",loc:"PEF2",to:["P2009","P2011","E239"],x:418,y:276},
{node:"P2011",loc:"PEF2",to:["P2010","P2002","P2013"],x:472,y:324},
//{node:"P2012",loc:"PEF2",to:["P2024","P3002"],x:450,y:346},
{node:"P2013",loc:"PEF2",to:["P2011","P2014","P2015"],x:503,y:351},
{node:"P2014",loc:"PEF2",to:["P2013","P2016","P2017"],x:524,y:369},
{node:"P2015",loc:"PEF2",to:["P2013","P2018","P2019"],x:475,y:384},
{node:"P2016",loc:"PEF2",to:["P2014","E255"],x:556,y:396},
{node:"P2017",loc:"PEF2",to:["P2014","P2018","P2020"],x:495,y:399},
{node:"P2018",loc:"PEF2",to:["P2005","P2017","P2015"],x:488,y:394},
{node:"P2019",loc:"PEF2",to:["P2023","P2015","C21"],x:462,y:396},
{node:"P2020",loc:"PEF2",to:["P2017","P2021","C22"],x:459,y:439},
{node:"P2021",loc:"PEF2",to:["P2022","P2020","C24"],x:446,y:454},
{node:"P2022",loc:"PEF2",to:["P2023","P2021","P2003"],x:433,y:442},
{node:"P2023",loc:"PEF2",to:["P2022","P2019","C23"],x:427,y:436},
//{node:"P2024",loc:"PEF2",to:["P2011","P2013","P2012"],x:471,y:324},

//PEF3
{node:"P3001",loc:"PEF3",to:["P2001","P1001","P4001","P3006"],x:312,y:096},
{node:"P3002",loc:"PEF3",to:["P1002","P2002","P4002","P3013"],x:518,y:345},
//{node:"P3003",loc:"PEF3",to:["P4001","P3005"],x:310,y:090},
//{node:"P3004",loc:"PEF3",to:["P4002","P3013"],x:518,y:351},
//{node:"P3005",loc:"PEF3",to:["P3003","P3006"],x:292,y:110},
{node:"P3006",loc:"PEF3",to:["P3007","P3001"],x:295,y:114},
{node:"P3007",loc:"PEF3",to:["E309","P3008","P3006"],x:312,y:129},
{node:"P3008",loc:"PEF3",to:["P3009","P3007","E315"],x:343,y:156},
{node:"P3009",loc:"PEF3",to:["P3008","P3010","E323"],x:390,y:198},
{node:"P3010",loc:"PEF3",to:["P3009","P3011","E333"],x:450,y:251},
{node:"P3011",loc:"PEF3",to:["P3010","P3013","E341"],x:498,y:294},
//{node:"P3012",loc:"PEF3",to:["P3002","P3011","P3013"],x:532,y:323},
{node:"P3013",loc:"PEF3",to:["P3011","P3002","P3014"],x:536,y:326},
{node:"P3014",loc:"PEF3",to:["P3013","E355"],x:623,y:403},

//PEF4
{node:"P4001",loc:"PEF4",to:["P3001","P2001","P1001","P4003"],x:299,y:071},
{node:"P4002",loc:"PEF4",to:["P3002","P2002","P1002","P4008"],x:507,y:343},
{node:"P4003",loc:"PEF4",to:["P4001","P4004"],x:278,y:094},
{node:"P4004",loc:"PEF4",to:["P4003","P4005","E409"],x:291,y:106},
{node:"P4005",loc:"PEF4",to:["P4004","P4006","E423"],x:376,y:181},
{node:"P4006",loc:"PEF4",to:["P4005","P4007","E431"],x:427,y:226},
{node:"P4007",loc:"PEF4",to:["P4006","P4008","E439"],x:479,y:272},
{node:"P4008",loc:"PEF4",to:["P4007","P4009","P4002"],x:530,y:317},
{node:"P4009",loc:"PEF4",to:["P4008","E455"],x:627,y:403},







];


title = document.location.href;
ind = title.indexOf('z/');

title = title.slice(ind+2).toUpperCase();

if (title == "INDEX.HTML" || title == "INDEX" || title == "")
{
  title = "HOME";

}
else {
showLoc(title);
}
  document.title = "CZUNAV - " + title;

$('#pos').append(title);

});
```

7.5. HTML Code

```
<!DOCTYPE html>
<head>
<script type="text/javascript" src="jquery-1.8.3.min.js"></script>
<link rel='stylesheet' type='text/css' href='main.css'/>
<script type='text/javascript' src='application.js'></script>
</head>
<body>
<script>
function reply_click(clicked_id)
{

  switch(clicked_id.charAt(0))
{
    case "F":
      $('#from').val("");
      $('#from').val(clicked_id.substring(1));
      break;
    case "T":
      $('#to').val("");
        $('#to').val(clicked_id.substring(1));
      break;
    case "L":
      $('#loc').val("");
            $('#loc').val(clicked_id.substring(1));
      break;

  }
}
</script>
```

```
<div id="header">
<a href="index.html">CZU Navigation System</a>

</div>
<div id="main">

<ul id="menu">
      <li><span class="menu">FROM :</span></li>
      <li><input id="from" class="input" type="text" name="from"
placeholder="FROM">
      <ul>

            <li><a id="FFAPPZ" onClick="reply_click(this.id)">FAPPZ</a>
                  <ul>
                  <li><a id="FAFA" onClick="reply_click(this.id)">AFA</a></li>
                  <li><a id="FAFB" onClick="reply_click(this.id)">AFB</a></li>
                  <li><a id="FAFC" onClick="reply_click(this.id)">AFC</a></li>
                  </ul>
            </li>
            <li><a id="FFZP" onClick="reply_click(this.id)">FZP</a></li>


            <li><a id="FPEF" onClick="reply_click(this.id)">PEF</a>
                  <ul>
                  <li><a id="FEI" onClick="reply_click(this.id)">EI</a></li>
                  <li><a id="FEII" onClick="reply_click(this.id)">EII</a></li>
                  <li><a id="FEIII" onClick="reply_click(this.id)">EIII</a></li>
                  <li><a id="FEIV" onClick="reply_click(this.id)">EIV</a></li>
                  <li><a id="FEV" onClick="reply_click(this.id)">EV</a></li>
                  <li><a id="FEI" onClick="reply_click(this.id)">E115</a></li>
                  <li><a id="FE119" onClick="reply_click(this.id)">E119</a></li>
                  <li><a id="FE223" onClick="reply_click(this.id)">E223</a></li>
                  <li><a id="FE239" onClick="reply_click(this.id)">E239</a></li>
                  <li><a id="FE255" onClick="reply_click(this.id)">E255</a></li>
                  <li><a id="FE309" onClick="reply_click(this.id)">E309</a></li>
                  <li><a id="FE315" onClick="reply_click(this.id)">E315</a></li>
                  <li><a id="FE323" onClick="reply_click(this.id)">E323</a></li>
                  <li><a id="FE333" onClick="reply_click(this.id)">E333</a></li>
                  <li><a id="FE341" onClick="reply_click(this.id)">E341</a></li>
                  <li><a id="FE355" onClick="reply_click(this.id)">E355</a></li>
                  <li><a id="FE409" onClick="reply_click(this.id)">E409</a></li>
                  <li><a id="FE423" onClick="reply_click(this.id)">E423</a></li>
                  <li><a id="FE431" onClick="reply_click(this.id)">E431</a></li>
                  <li><a id="FE439" onClick="reply_click(this.id)">E439</a></li>
                  <li><a id="FE455" onClick="reply_click(this.id)">E455</a></li>
                  <li><a id="FC11" onClick="reply_click(this.id)">C11</a></li>
                  <li><a id="FC12" onClick="reply_click(this.id)">C12</a></li>
                  <li><a id="FC13" onClick="reply_click(this.id)">C13</a></li>
                  <li><a id="FC14" onClick="reply_click(this.id)">C14</a></li>
                  <li><a id="FC21" onClick="reply_click(this.id)">C21</a></li>
                  <li><a id="FC22" onClick="reply_click(this.id)">C22</a></li>
                  <li><a id="FC23" onClick="reply_click(this.id)">C23</a></li>
                  <li><a id="FC24" onClick="reply_click(this.id)">C24</a></li>
                  </ul>
            </li>
            <li><a id="FMI" onClick="reply_click(this.id)">TF</a>
                  <ul>
                  <li><a id="FMI" onClick="reply_click(this.id)">MI</a></li>
                  <li><a id="FMII" onClick="reply_click(this.id)">MII</a></li>
                  <li><a id="FMIII" onClick="reply_click(this.id)">MIII</a></li>
                  </ul>
            </li>
            <li><a>OTHER</a>
                  <ul>
                  <li><a id="FAULA" onClick="reply_click(this.id)">AULA</a></li>
                  <li><a id="FBUN" onClick="reply_click(this.id)">BUN</a></li>
                  <li><a id="FFARMA" onClick="reply_click(this.id)">FARMA</a></li>
```

```
                  <li><a id="FJIH" onClick="reply_click(this.id)">JIH</a></li>
                  <li><a id="FKTV" onClick="reply_click(this.id)">KTV</a></li>
                  <li><a id="FMENSA" onClick="reply_click(this.id)">MENSA</a></li>
                  <li><a id="FREKTORAT"
onClick="reply_click(this.id)">REKTORAT</a></li>
                  <li><a id="FSTABLE"
onClick="reply_click(this.id)">STABLE</a></li>
                  <li><a id="FSIC" onClick="reply_click(this.id)">SIC</a></li>
                  <li><a id="FRK" onClick="reply_click(this.id)">RK</a></li>
                  </ul>
            </li>



      </ul>
      </li>
      <li><span class="menu">TO : </span></li>
      <li><input id="to" class="input" type="text" name="to" placeholder="TO">
      <ul>

            <li><a id="TFAPPZ" onClick="reply_click(this.id)">FAPPZ</a>
                  <ul>
                  <li><a id="TAFA" onClick="reply_click(this.id)">AFA</a></li>
                  <li><a id="TAFB" onClick="reply_click(this.id)">AFB</a></li>
                  <li><a id="TAFC" onClick="reply_click(this.id)">AFC</a></li>
                  </ul>
            </li>
            <li><a id="TFZP" onClick="reply_click(this.id)">FZP</a></li>


            <li><a id="TPEF" onClick="reply_click(this.id)">PEF</a>
                  <ul>
                  <li><a id="TEI" onClick="reply_click(this.id)">EI</a></li>
                  <li><a id="TEII" onClick="reply_click(this.id)">EII</a></li>
                  <li><a id="TEIII" onClick="reply_click(this.id)">EIII</a></li>
                  <li><a id="TEIV" onClick="reply_click(this.id)">EIV</a></li>
                  <li><a id="TEV" onClick="reply_click(this.id)">EV</a></li>
                  <li><a id="TEI" onClick="reply_click(this.id)">E115</a></li>
                  <li><a id="TE119" onClick="reply_click(this.id)">E119</a></li>
                  <li><a id="TE223" onClick="reply_click(this.id)">E223</a></li>
                  <li><a id="TE239" onClick="reply_click(this.id)">E239</a></li>
                  <li><a id="TE255" onClick="reply_click(this.id)">E255</a></li>
                  <li><a id="TE309" onClick="reply_click(this.id)">E309</a></li>
                  <li><a id="TE315" onClick="reply_click(this.id)">E315</a></li>
                  <li><a id="TE323" onClick="reply_click(this.id)">E323</a></li>
                  <li><a id="TE333" onClick="reply_click(this.id)">E333</a></li>
                  <li><a id="TE341" onClick="reply_click(this.id)">E341</a></li>
                  <li><a id="TE355" onClick="reply_click(this.id)">E355</a></li>
                  <li><a id="TE409" onClick="reply_click(this.id)">E409</a></li>
                  <li><a id="TE423" onClick="reply_click(this.id)">E423</a></li>
                  <li><a id="TE431" onClick="reply_click(this.id)">E431</a></li>
                  <li><a id="TE439" onClick="reply_click(this.id)">E439</a></li>
                  <li><a id="TE455" onClick="reply_click(this.id)">E455</a></li>
                  <li><a id="TC11" onClick="reply_click(this.id)">C11</a></li>
                  <li><a id="TC12" onClick="reply_click(this.id)">C12</a></li>
                  <li><a id="TC13" onClick="reply_click(this.id)">C13</a></li>
                  <li><a id="TC14" onClick="reply_click(this.id)">C14</a></li>
                  <li><a id="TC21" onClick="reply_click(this.id)">C21</a></li>
                  <li><a id="TC22" onClick="reply_click(this.id)">C22</a></li>
                  <li><a id="TC23" onClick="reply_click(this.id)">C23</a></li>
                  <li><a id="TC24" onClick="reply_click(this.id)">C24</a></li>
                  </ul>
            </li>
            <li><a id="TMI" onClick="reply_click(this.id)">TF</a>
                  <ul>
                  <li><a id="TMI" onClick="reply_click(this.id)">MI</a></li>
                  <li><a id="TMII" onClick="reply_click(this.id)">MII</a></li>
                  <li><a id="TMIII" onClick="reply_click(this.id)">MIII</a></li>
                  </ul>
```

```
            </li>
            <li><a>OTHER</a>
                  <ul>
                  <li><a id="TAULA" onClick="reply_click(this.id)">AULA</a></li>
                  <li><a id="TBUN" onClick="reply_click(this.id)">BUN</a></li>
                  <li><a id="TFARMA" onClick="reply_click(this.id)">FARMA</a></li>
                  <li><a id="TJIH" onClick="reply_click(this.id)">JIH</a></li>
                  <li><a id="TKTV" onClick="reply_click(this.id)">KTV</a></li>
                  <li><a id="TMENSA" onClick="reply_click(this.id)">MENSA</a></li>
                  <li><a id="TREKTORAT"
onClick="reply_click(this.id)">REKTORAT</a></li>
                  <li><a id="TSTABLE"
onClick="reply_click(this.id)">STABLE</a></li>
                  <li><a id="TSIC" onClick="reply_click(this.id)">SIC</a></li>
                  <li><a id="TRK" onClick="reply_click(this.id)">RK</a></li>
                  </ul>
            </li>


      </ul>
      </li>
      <li><input id="route" class="submit" type="submit" value=" FIND! "></li>
      <li id="txt"><span class="menu">OR</span></li>

      <li><input id="loc" class="input" type="text" name="loc" placeholder="LOC">
      <ul>

            <li><a id="LFAPPZ" onClick="reply_click(this.id)">FAPPZ</a>
                  <ul>
                  <li><a id="LAFA" onClick="reply_click(this.id)">AFA</a></li>
                  <li><a id="LAFB" onClick="reply_click(this.id)">AFB</a></li>
                  <li><a id="LAFC" onClick="reply_click(this.id)">AFC</a></li>
                  </ul>
            </li>
```

```
<li><a id="LFZP" onClick="reply_click(this.id)">FZP</a></li>


<li><a id="LPEF" onClick="reply_click(this.id)">PEF</a>
      <ul>
      <li><a id="LEI" onClick="reply_click(this.id)">EI</a></li>
      <li><a id="LEII" onClick="reply_click(this.id)">EII</a></li>
      <li><a id="LEIII" onClick="reply_click(this.id)">EIII</a></li>
      <li><a id="LEIV" onClick="reply_click(this.id)">EIV</a></li>
      <li><a id="LEV" onClick="reply_click(this.id)">EV</a></li>
      <li><a id="LEI" onClick="reply_click(this.id)">E115</a></li>
      <li><a id="LE119" onClick="reply_click(this.id)">E119</a></li>
      <li><a id="LE223" onClick="reply_click(this.id)">E223</a></li>
      <li><a id="LE239" onClick="reply_click(this.id)">E239</a></li>
      <li><a id="LE255" onClick="reply_click(this.id)">E255</a></li>
      <li><a id="LE309" onClick="reply_click(this.id)">E309</a></li>
      <li><a id="LE315" onClick="reply_click(this.id)">E315</a></li>
      <li><a id="LE323" onClick="reply_click(this.id)">E323</a></li>
      <li><a id="LE333" onClick="reply_click(this.id)">E333</a></li>
      <li><a id="LE341" onClick="reply_click(this.id)">E341</a></li>
      <li><a id="LE355" onClick="reply_click(this.id)">E355</a></li>
      <li><a id="LE409" onClick="reply_click(this.id)">E409</a></li>
      <li><a id="LE423" onClick="reply_click(this.id)">E423</a></li>
      <li><a id="LE431" onClick="reply_click(this.id)">E431</a></li>
      <li><a id="LE439" onClick="reply_click(this.id)">E439</a></li>
      <li><a id="LE455" onClick="reply_click(this.id)">E455</a></li>
      <li><a id="LC11" onClick="reply_click(this.id)">C11</a></li>
      <li><a id="LC12" onClick="reply_click(this.id)">C12</a></li>
      <li><a id="LC13" onClick="reply_click(this.id)">C13</a></li>
      <li><a id="LC14" onClick="reply_click(this.id)">C14</a></li>
      <li><a id="LC21" onClick="reply_click(this.id)">C21</a></li>
      <li><a id="LC22" onClick="reply_click(this.id)">C22</a></li>
      <li><a id="LC23" onClick="reply_click(this.id)">C23</a></li>
      <li><a id="LC24" onClick="reply_click(this.id)">C24</a></li>
```

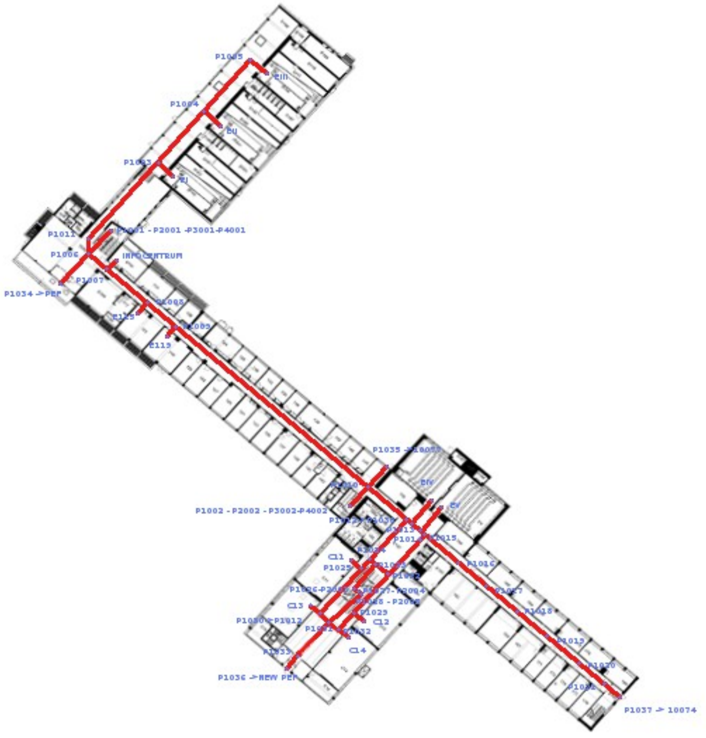
```
                  </ul>
            </li>
            <li><a id="LMI" onClick="reply_click(this.id)">TF</a>
                  <ul>
                  <li><a id="LMI" onClick="reply_click(this.id)">MI</a></li>
                  <li><a id="LMII" onClick="reply_click(this.id)">MII</a></li>
                  <li><a id="LMIII" onClick="reply_click(this.id)">MIII</a></li>
                  </ul>
            </li>
            <li><a>OTHER</a>
                  <ul>
                  <li><a id="LAULA" onClick="reply_click(this.id)">AULA</a></li>
                  <li><a id="LBUN" onClick="reply_click(this.id)">BUN</a></li>
                  <li><a id="LFARMA" onClick="reply_click(this.id)">FARMA</a></li>
                  <li><a id="LJIH" onClick="reply_click(this.id)">JIH</a></li>
                  <li><a id="LKTV" onClick="reply_click(this.id)">KTV</a></li>
                  <li><a id="LMENSA" onClick="reply_click(this.id)">MENSA</a></li>
                  <li><a id="LREKTORAT"
onClick="reply_click(this.id)">REKTORAT</a></li>
                  <li><a id="LSTABLE"
onClick="reply_click(this.id)">STABLE</a></li>
                  <li><a id="LSIC" onClick="reply_click(this.id)">SIC</a></li>
                  <li><a id="LRK" onClick="reply_click(this.id)">RK</a></li>
                  </ul>
            </li>


      </ul>
      </li>
<li><input id="location" class="submit" type="submit" name="location"
value="SHOW!"></li>
      <li id="mistake"><input id="mistake" class="submit" type="submit"
name="location" value="Find mistake!">
</ul>
```

```
</div>

<div id="loading">
<h1>Finding Route... Please wait</h1>
</div>

<canvas id='canvas0' style='background:url("+canvas1+");'width='1000pt'
height='600pt'></canvas>
<canvas id='canvasG1' style='background:url("+canvas1+");'width='1000pt'
height='600pt'></canvas>
<canvas id='canvas1' style='background:url("+canvas1+");'width='1000pt'
height='600pt'>We are sorry, but your browser does not support JavaScript
Canvas</canvas>
<canvas id='canvasG2' style='background:url("+canvas1+");'width='1000pt'
height='600pt'></canvas>
<canvas id='canvas2' style='background:url("+canvas1+");'width='1000pt'
height='600pt'></canvas>

<div id='position'>
<h1 id='pos'>You are currently at: </h1>
</div>
<div id="footer">
<span class="text">Martin Riha &#0169; CULS 2013</span>
</div>
</body>
</html>
```

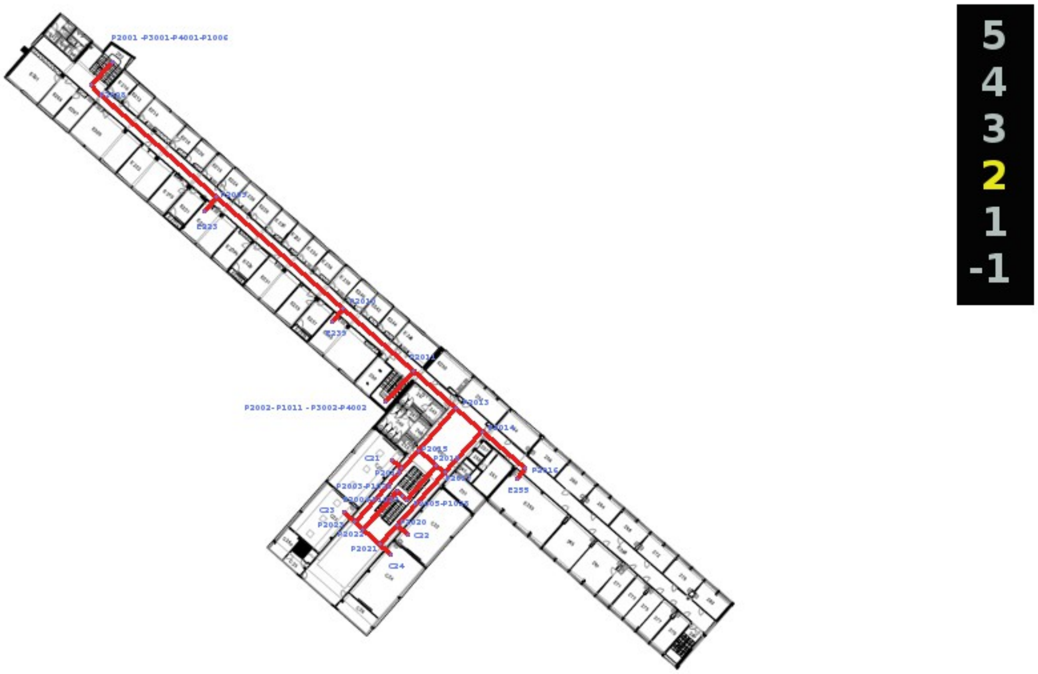
7.6. CSS

```
.submit {
  color: white;
  background-color: #1E82CC;
  font-weight: bold;
  height:100px;
  font-size:32px;
  padding: 0 10 0 10px;
  border: none;
  width: 125px;
}

.menu {
  color: white;
  background-color: #1E82CC;
  font-weight: bold;
  height:100px;
  font-size:22px;
  width: 125px;
  border: none;
  line-height: 4.5em;
}

.submit:hover {
      border-right:1px solid #fff;
      border-left:1px solid #fff;
      background:#5798B4;
      color:white;
      text-decoration:none;
}

.input {
  height:96px;
  font-size:30px;
  width: 110px;
  margin: 0px;
  padding: 0px;

}

.input:hover {
      border-right:1px solid #fff;
      border-left:1px solid #fff;
      text-decoration:none;
}

a:link {  color: white;
  font-size: 20px;
  vertical-align:middle;
  font-weight: bold;
  line-height: 90px;
  text-decoration: none
}
a:visited {  color: white;
  font-size: 20px;
  vertical-align:middle;
  font-weight: bold;
  line-height: 90px;
  text-decoration: none
}
body {font-size: 14px;font-family: Verdana; color: white; background-color: 
#8CA9FF;text-align:center;}

#header {
      position: relative;
      background-color: #1E82CC;
      height: 100px;
      vertical-align: middle;
      text-align: center;
      max-width: 100%;
      }

#main {
      position: relative;
      background-color: #1E82CC;
      height: 100px;
      vertical-align: middle;
      text-align: center;
      width: 1000px;
      margin: 10px 0 0px 0;
      line-height: 30px;
      margin-left: auto;
      margin-right: auto;
      }

#position{
      position: relative;
      background-color: #1E82CC;
      height: 50px;
      margin: -15px 0px 10px 0px;
      text-align: center;
      margin-left: auto ;
      margin-right: auto ;

      width:1000px;
}

#footer {
      position: relative;
      background-color: #1E82CC;
      height: 100px;
      vertical-align: middle;
      text-align: center;
      max-width: 100%;
      }
#pos {
   color: white;
  font-size: 20px;
  vertical-align:middle;
  font-weight: bold;
  line-height: 50px;
  text-decoration: none
}

.text {
  color: white;
  font-size: 15px;
  vertical-align:middle;
  font-weight: bold;
  line-height: 80px;
  }

.inline {
    height: 100%;

    display: inline-block;

    border-radius: 5px;

}
.bb {
  width:1200px;
  background-color: white;
  text-align: top;
  margin-left: auto ;
  margin-right: auto ;
  max-height: 100%;
}
#menu {
  width: 1000px;
}

#menu, #menu ul {
      list-style:none;
      padding:0;
      margin:0;

}
#menu li {
      float:left;
      position:static;
      width: 125px;
      line-height: 2.5em;
}

#menu li ul {
      position:absolute;
      margin-top:0px;
      margin-left:.0px;
      display:none;
}
#menu a {
      display:list-item;
      background:#1E82CC;
      color:white;
      text-decoration:none;
      padding:0 0px;
}
#menu a:hover {
      background-color:#5798B4;
      color:#fff;
}
#menu ul {
      border-top:1px solid #fff;
}
#menu ul a {
      border-right:none;
      border-right:1px solid #fff;
      border-bottom:1px solid #fff;
      border-left:1px solid #fff;
      background:#1E82CC;
}

/* SHOW SUBMENU 1 */
#menu li:hover ul, #menu li.over ul {
      display:block;

}
#menu li:hover ul ul, #menu li.over ul ul {
      display:none;
      position: absolute;
}
/* SHOW SUBMENU 2 */
#menu ul li:hover ul, #menu ul li.over ul {
      display:block;

}
```

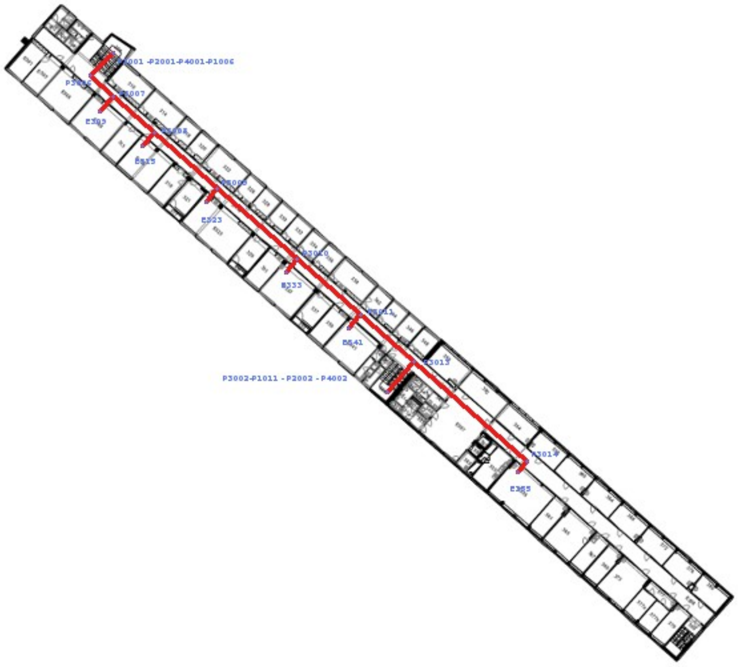
7.7. Nodes - First Floor PEF



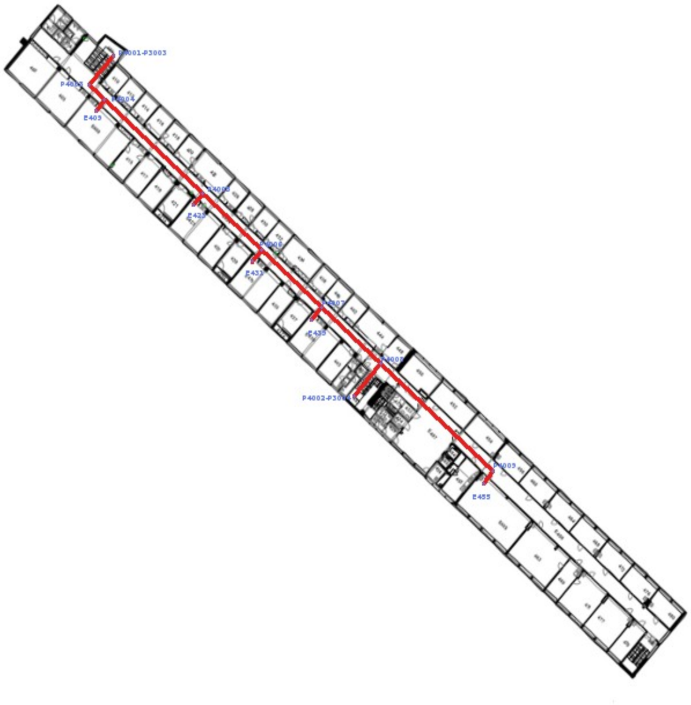
7.8. Nodes – Second Floor PEF



7.9. Nodes – Third Floor PEF



7.10. Nodes – Fourth Floor PEF


P4001-P3003
P4003
P4004
E403
P4005
E425
P4006
E433
P4007
E439
P4008
P4002-P3004
P4009
E455