SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

FCHPT-5415-68770

TIMETABLE AND INTERACTIVE FACULTY PLAN – GRAPHICAL USER INTERFACE

BACHELOR THESIS

2014 **Rudolf Halás**

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

TIMETABLE AND INTERACTIVE FACULTY PLAN – GRAPHICAL USER INTERFACE

BACHELOR THESIS

FCHPT-5415-68770

Study programme:	Automation, Information Engineering and Management in Chemistry and Food Industry
Study field number:	2621
Study branches combination:	5.2.14 Automation, 5.2.52 Industrial Engineering
Department:	Department of Information Engineering and Process Control
Supervisor:	Prof. Ing. Miroslav Fikar, DrSc.

Bratislava 2014 **Rudolf Halás**

Slovak University of Technology in Bratislava
Ústav informatizácie, automatizácie a matematiky

Faculty of Chemical and Food Technology
Academic year: 2013/2014
Reg. No.: FCHPT-5415-68770

BACHELOR THESIS TOPIC

Student:	**Rudolf Halás**
Student's ID:	68770
Study programme:	Automation, Information Engineering and Management in Chemistry and Food Industry
Study branches combination:	5.2.14 automation, 5.2.52 industrial engineering
Thesis supervisor:	prof. Ing. Miroslav Fikar, DrSc.

Topic: **Timetable and Interactive Faculty Plan – Graphical User Interface**

Specification of Assignment:

The aim of the work is to create a complex web application for interactive faculty plan. Moreover, it will provide navigation in student timetables. Also, it will simplify contact with teachers using their location data. The thesis is a part of a team project.

This project part is primarily focused on creation of graphical user interface of the application using various technologies (SVG, CSS, Javascript). It also contains plan of rooms in the buildings according to fire escape plans.

Tasks:

1 – Implementation of SVG technologies to view data
2 – Treatment of fire escape plans of FCFT STU
3 – Graphical design of web application
4 – Application of PHP framework with MVC architecture

Length of thesis: 40

Selected bibliography:

1. *Visualizing information using SVG and X3D : XML-based technologies for the XML-based web.* Londýn: Springer Verlag, 2005. 298 s. ISBN 1-85233-790-7.

Assignment procedure from:	17. 02. 2014
Date of thesis submission:	24. 05. 2014

L. S.

Rudolf Halás
Student

prof. Ing. Miroslav Fikar, DrSc.
Head of department

prof. Ing. Miroslav Fikar, DrSc.
Study programme supervisor

Acknowledgement

The researcher has taken efforts in this project. However, it would not have been possible without the kind support, supervision, encouragement, and help of prof. Miroslav Fikar.

My thanks and appreciations also go to my colleague Pavol Ďurina in developing the project and people who have willingly helped me out with their abilities.

Abstract

The goal of this project is a creation of web application that will be used as an interactive Floor plan of Faculty of Chemical and Food Technology. In addition it will offer timetable data. Application is meant to make contacting with teachers easier by providing their current location information based on their personal timetable. This bachelor thesis is a part of team project. This part is mainly focused on graphical user interface creation with the use of various web technologies (SVG – Scalable Vector Graphics, CSS – Cascading Style Sheets, JavaScript). Data used to display the rooms are extracted based on provided fire escape plans.

Keywords:

floor plan, timetable, graphical user interface

Súhrn

Cieľom práce je vytvorenie komplexnej webovej aplikácie, ktorá bude slúžiť ako interaktívny plán budovy fakulty a zároveň bude umožnovať prehľadnú navigáciu v rozvrhovej štruktúre. Jednou z ďalších úloh je zjednodušiť kontakt s vyučujúcimi prostredníctvom údajov o ich lokácii. Zadanie je časťou tímového projektu. Táto časť projektu je primárne zameraná na vytvorenie grafického rozhrania aplikácie použitím rôznych webových technológii (SVG, CSS, JavaScript). Súčasťou zadania je rozvrhnutie miestností v budove podľa požiarnych evakuačných plánov.

Kľúčové slová:

orientačný plán, rozvrh, grafické používateľské rozhranie

Contents

List of Abbreviations

CSS	–	Cascading Style Sheets
DTD	–	Document Type Definition
FCFT	–	Faculty of Chemical and Food Technology
GPS	–	Global Positioning System
GUI	–	Graphical User Interface
HTML	–	HyperText Markup Language
HTTP	–	HyperText Transfer Protocol
JSON	–	JavaScript Object Notation
MVC	–	Model–View–Controller architecture pattern
PHP	–	PHP: Hypertext Preprocessor
SQL	–	Structured Query Language
STU	–	Slovak University of Technology in Bratislava
SVG	–	Scalable Vector Graphics
URL	–	Uniform Resource Locators
W3C	–	World Wide Web Consortium
XML	–	Extensible Markup Language

Chapter 1

Introduction

In our lives we travel everyday. Since it is not possible to memorize the layout of every city or building we have ever entered we need some kind of plans to help us to get to our destination. Nowadays it is common to use Global Positioning System (GPS) in cars while driving or walking in the open-air environment. This has replaced maps in their printed form. We often expect more than plain sketch of our current location. Traffic information, weather data, advertisement offers, etc. are just the information we appreciate. This is what lead to creation of interactive maps or building plans.

Imagine you arrive at a shopping center that you have never been before. You need to buy a particular item and you expect to find it in a particular shop. Since you do not know the exact location of a shop you look for a plan that will show you. The traditional static plan will probably help you find the shop but its interactive version could offer you more. You could possibly notice that there is a discount of a product that you need to buy in another shop. Or that the toilettes you planned to use on your way to your car are out of order. In current competitive business environment the presence of additional information might be a small benefit of great importance when it comes to making a decision which shopping center will you visit.

Every year our faculty enrols many new students who are often lost in a maze of rooms created by unlabelled corridors, illogically numbered rooms and offices placed where you least expect them. The same applies to visitors. The things that might help to find the destination are concierge localized in entrance lobby or escape plans.

Me and my colleague Pavol Ďurina have decided to create an interactive faculty plan which will not only show location of a particular room but also provide additional information about ongoing events, vacancy of rooms, classes, and faculty employees. The target is to achieve this by creating web application based on widely used web technologies with modern approach.

My application will behave as a front-end communicating with back-end part using HyperText Transfer Protocol (HTTP) requests.

Chapter 2

Traditional Approach

Interactive floor plans have mostly been created using Adobe Flash technology directly or with the help of existing applications like Floorplanner [10]. Flash is popular with developers because of ActionScript scripting possibilities and simple graphics creation.

Main advantages of using Flash are:

Size of final application – Because of using mainly vector graphics in designing process.

Interactivity – Flash can easily react to mouse movement, clicking, etc.

Audio/Video files – Developers can use audio or video advertisement in their presentations.

Streaming of media files

Exact look of application on every platform

Simple design creation – Designers can draw shapes, buttons, etc. as if they used an image editor.

Support for file reading/editing

Because of these features Flash has been very popular lately. There is a countless amount of web pages that use Flash technology. It is used either for interactive banners and media playback or as the main technology to create the whole web page. However, is has several disadvantages that resulted into not using it for this project:

- **Back button in web browser cannot be used** – It can be very confusing for unexperienced internet users when they cannot use back button to return to previous Flash page.
- If previously not intended **the size of displayed text cannot be adjusted.**
- **Users cannot scroll using their mouse wheel** – It is not impossible yet designers prefer to use scrollbars.
- **History in users' browsers cannot log pages in Flash application.**
- **Browsers' favorites cannot get users to a particular place in Flash application** – Browser sees the whole Flash web page as only one application.
- If previously not intended **users cannot search for the expression they need** – Since Find is one of the most used features in all browsers this can create significant discomfort for the users.

Flash should not be used to create important web page elements. There are lots of users who browse the internet with Flash technology disabled. Thus they are not be able to interact with the web page as intended. Many developers use Flash technology at wrong parts of web presentation and consequently many users do not like using Flash at all. Nowadays there is an intention of creating web applications that can be accessed on multiple devices while offering the same features. However Flash does not fit this intention. Moreover Adobe stopped supporting Linux version of Flash and Windows version is most likely to follow. It's future might be in web applications that focus on media processing and playback but it's main opponent is HTML5 that has gained a lot of popularity lately. [19]

Chapter 3

Goals

Based on drawbacks described in the previous chapter it has been decided not to use Flash technology for this web application. Instead, the main goal is to create floor plan as interactive as it would be if Flash were used but based on commonly used Web technologies such as HTML, CSS, PHP, SVG, and JavaScript. It helps to reduce Flash limitations that can be seen in other floor plans. Moreover, using such web technologies allows developers to offer application for viewing on smart-phones or tablets with no additional need to edit the application code.

Additional goals are following:

- Implement SVG technology for display of data.
- Process faculty plans.
- Create graphical user interface of web application.
- Use PHP framework with Model–View–Controller (MVC) architecture to keep application structured.

Chapter 4

Used Web Technologies

4.1 PHP

PHP is a general-purpose server-side scripting language designed for web development to produce dynamic web pages that are viewed in web browsers [14]. The word “dynamic” means that they can be different after each refresh without direct change of code. It is one of the first developed server-side scripting languages to be included into an HTML source code. The code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP was created by Rasmus Lerdorf in 1995 [11] and nowadays it is an open source software released under the PHP License [13]. PHP is a recursive acronym standing for “PHP: Hypertext Preprocessor” [12].

I have used PHP in multiple projects and because of the need of running final web application on unix server, my experience with PHP scripting and user friendly documentation was the choice of main server-side scripting language straightforward.

Since this web application is meant to be used for longer period of time it is probable that there will be more different people administering it. Not only the GUI part but also the application itself. It is a good habit to maintain some structure of the code for easier understanding and better organization. Model–view–controller (MVC) architecture pattern has been selected.

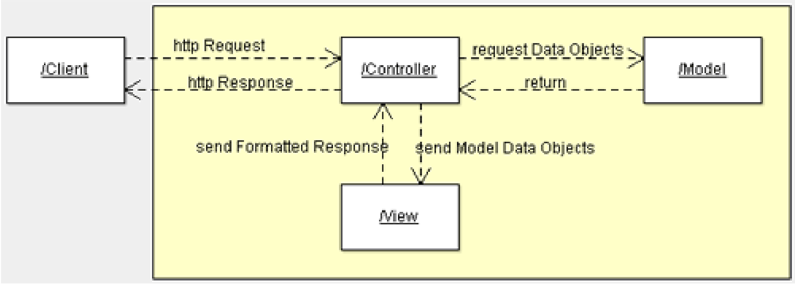


Figure 4.1: Model–view–controller concept.

4.1.1 Model–View–Controller

Model–view–controller (MVC, Figure 4.1) architecture [18] has been very popular recently. Zend invests money not only to PHP development but also improves Zend Framework with MVC architecture. Microsoft has created new web framework ASP.NET MVC and there are hundreds of frameworks with various license policies maintained by various creators for many technologies. There are predictions that MVC popularity will rise mostly because of its features.

It is known that one can grasp the basic idea of MVC in a very short time but mastering all the details can take months or years. There is a lack of MVC introductions which creates differences in developers' understanding of architecture and leads to "MVC web application" that does not follow MVC pattern at all. [9]

There are two main reasons why the MVC design pattern should be used:

- They are reusable: when the problems recurs, there is no need to invent a new solution, we just have to follow the pattern and adapt it as necessary.
- They are expressive: by using the MVC design pattern our application becomes more expressive.

Model The model object maintains all the data that need to be displayed. It is aware about all the operations that can be applied to transform that object. It only represents the data of an application and the business rules that govern access to and updates of this data. Model is not aware about how the data will be displayed to the browser.

View The view represents the presentation layer of the application. The view object refers to the model. It uses the query methods of the model to obtain the contents and renders it. The view must not be dependent on the application logic. It remains same if there is any modification in the logic. It maintains the consistency in its presentation when the model or data change.

Controller Whenever the user sends a request for something then it always goes through the controller. The controller intercepts the requests from view and delivers it to the model for the appropriate action. After the action has been taken on the data the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often have to work very closely together.

Connection amongst components is shown in Figure 4.1 There are only two direct references in MVC architecture:

- Controller has direct reference to Model so that it can edit its data.
- View has direct reference to Model so that it can view its data.

Practically with the use of specific MVC type it is common to have one or two way relation between Controller and View. Direct reference between Model and two other components must not exist. It is considered a fatal error in application design. Indirect reference can be found between View and Model which describes the situation when Model data are changed and View is notified by e.g. Observer pattern. [7]

4.2 JavaScript

JavaScript is a dynamic general-purpose scripting language that is weakly typed, case sensitive and has first-class functions. It is a multi-paradigm language that supports object-oriented, imperative, and functional programming styles [16].

JavaScript was formalized in the ECMAScript language standard and is primarily used in the form of client-side JavaScript implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites [15].

This enables programmatic access to computational objects within a host environment. JavaScript copies many names and naming conventions from Java but the two languages that are often mistaken are otherwise unrelated and have very different semantics [8].

4.3 Cascading Style Sheets

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content from document presentation. This separation improves content accessibility, provides more flexibility and control in the specification of presentation characteristics, enables multiple pages to share formatting, and reduces complexity and repetition in the structural content. CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed [26].

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable [25].

The CSS specifications are maintained by the World Wide Web Consortium (W3C). [6]

4.4 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a family of specifications of an eXtensible Markup Language (XML) based file format for two-dimensional vector graphics. SVG images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted, and compressed. XML files SVG images can be created and edited with any text editor. However, it is often more convenient to create them with drawing programs such as Inkscape or Adobe Illustrator. The SVG specification is an open standard that has been under development by the World Wide Web Consortium (W3C) since 1999. All major modern web browsers have support for SVG and can render mark-up directly. Earlier versions of Microsoft Internet Explorer (IE) do not support SVG natively [27].

An example of SVG file is shown below (Listing 4.1):

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="red" />
</svg>
```

Listing 4.1: Example of SVG code.

4.5 Software Framework

A software framework [5] is an abstraction in which software providing generic functionality can be selectively adjusted by user code so that developers can use it to support them in creating and managing other software projects. It often includes support programs, compilers, code libraries, an application programming interface (API), recommended developing patterns, and tool sets that bring together all the different components to enable development of a project or solution.

Frameworks contain key distinguishing features that separate them from normal libraries:

Inversion of control – In a framework, unlike in libraries or normal user applications, the overall program's flow of control is not dictated by the caller, but by the framework.

Default behavior – A framework has a default behavior. This default behavior must actually be some useful one.

Extensibility – A framework can be extended by the user usually by a selective override or specialized by user code providing specific functionality.

Non-modifiable framework code – The framework code, in general, is not allowed to be modified. Users can extend the framework, but not modify its code [22].

4.5.1 Laravel

I have decided to implement MVC architecture in this web application. To ease creation process I have chosen PHP framework Laravel [21] which I consider simple yet efficient enough. Laravel is a powerful PHP framework similar to Symfony with a small footprint allowing developers to create full-featured web applications.

Main features why developers use Laravel:

- Simple installation and framework updates leveraging Composer [20] – the most popular dependency manager for PHP and often described as the future of PHP.
- Clear, thorough documentation – user guide with tutorials and examples that makes it easy to understand features of Laravel.
- Facade system allowing for demonstrative code.
- Developed to have possibilities of creating easily testable applications and following SOLID principles [17].
- Exceptional performance.
- Full PHP 5.4 support.
- Bundle system for additional libraries and lots of libraries offered via Github.

4.5.2 jQuery

Working with JavaScript is known to be very frustrating mostly because of complex event handling and matching a set of elements in a document. jQuery is simple to understand and easy to use, which means the learning curve is small, while the possibilities are great. It makes developers’ life easier as shown on example below (Listing 4.2):

```
JavaScript:

function toggle(layer_ref){
    if (document.all) //IS IE 4 or 5 (or 6 beta)
        eval(’if (document.all.’ + layer_ref + ’.style.display==\’none\’)
              document.all.’ + layer_ref + ’.style.display= \’block\’;
              else document.all.’ + layer_ref + ’.style.display= \’none\’’);
    if (document.layers) { //IS NETSCAPE 4 or below
        if (document.layers[layer_ref].display == ’none’)
            document.layers[layer_ref].display = ’block’;
        else
            document.layers[layer_ref].display = ’none’;
    }
    if (document.getElementById && !document.all) // firefox
    {
        hza = document.getElementById(layer_ref);
        if (hza.style.display==’none’)
           hza.style.display= ’block’;
        else
           hza.style.display=’none’;
    }
}

jQuery:

$(’#button’).click(function() {
    $(’#layer_ref’).toggle();
});
```

Listing 4.2: Comparison of pure JavaScript and jQuery code

There are two functions that toggle visibility of layer_ref div after clicking a button. Effectiveness and simplicity of jQuery code is obvious.

Developers do not need to focus on JavaScript cross-browser optimization since jQuery is maintained to be fully functional in all modern browsers. jQuery is used as a JavaScript library that is included to our webpage. It can be accessed remotely or locally.

In Web2.0, development process is often expected to interact with many elements, create various animations, catch and process events or use AJAX. This is now very simple with jQuery compared to use of pure JavaScript. Nowadays 78.6% of top 10000 websites use jQuery which shows that it is very popular [4].

Chapter 5

Realization

Most of the coding has been done in Sublime Text 2 editor which is customizable Python-based code editor with multiple features that helped during the development [23]. Gimp [24] has been used as a raster graphics editor and Inkscape [3] as a vector graphics editor. Both are open source editors which have been sufficient to fulfil the needs of this task.

5.1 Design Layout

The whole design creation process has been painted in Gimp environment. This step was very important mainly because it offered an overall look on the application being created. Simple yet visually pleasing design has been the primary goal (Figure 5.1). I decided to use rectangular shapes combined in a way so that it looked modern, flat, yet well-arranged. Based on current graphic manual for our faculty the specific yellow has been used to blend in with other faculty applications combined with a few other fitting colors.

It has been important to take into consideration all the features that should be offered to users of the application. SVG plan is the largest part because of all the details to be visible. There is a search box which will be used as the main direct interaction spot seconded with sidebar offering tree list of rooms. Its position has been intentionally chosen to attract eyes of users. Additional feature is the possibility of accessing the search bar using F key on keyboard. Similar keyboard actions are offered when using plan. Both WSAD and HJKL can be used for panning

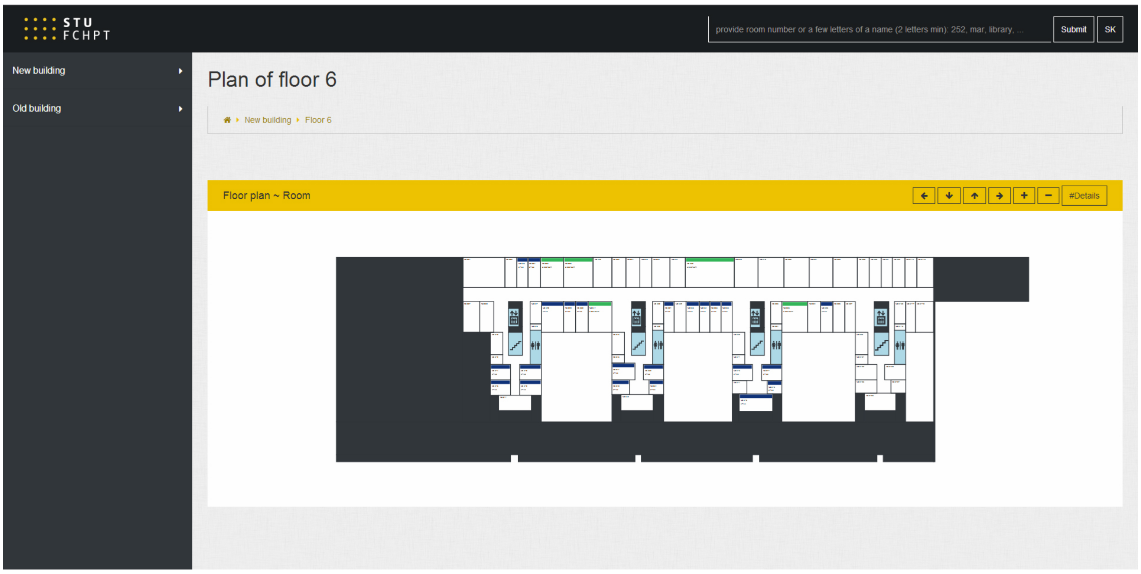


Figure 5.1: Default view of the 6th floor in new building when no particular room is selected.

an QZ can be used for zooming. This is meant to speed up the flow of using the application focusing on user experience for advanced users often searching for such possibilities. For users to come across provided keystrokes there are tips displayed when hovering section they relate to.

5.2 Processing SVG Data

I have been provided emergency escape plans which I used to create layout of floors in both old and new building. These plans were processed in Inkscape where rectangles have been placed to cover rooms in plans (Figure 5.2). Each rectangle was created in the specific rooms layer and rectangular elements were named according to rooms they have covered. This produced group element in final SVG file which was filled with rectangular elements when identifiers of these elements formed identifiers in database for a specific room (Listing 5.1).

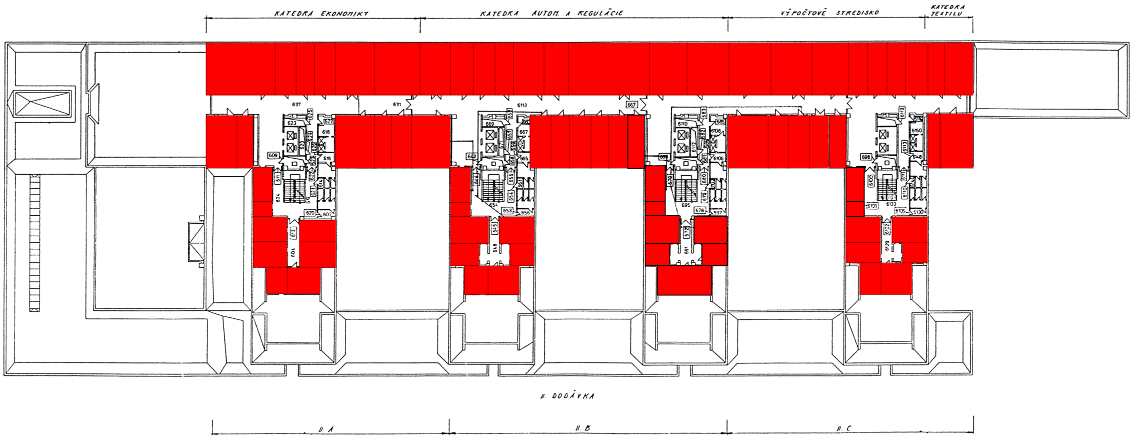


Figure 5.2: Processing a floor plan to get coordinates of rooms based on covering rooms with rectangles which coordinates are used.

```
<g id="rooms">
    <rect id="r1-2-206" x="224" y="109" fill="#FF0000" width="18"
        height="16"/>
    <rect id="r1-2-222" x="242" y="79" fill="#FF0000" width="52"
        height="46"/>
    ...
</g>
```

Listing 5.1: Example of named SVG group element

Contents of specific group elements has been cut out and inserted into equally named text files as a source data for processing final SVG plans. Text file created this way has been parsed and data has been stored in database to form list of possible rooms to work with. Another part of application has processed all these files and provided final SVG plans which can easily be changed by editing text files and re-running processing method. The reason for such storage is the fact that creating SVG plans on the fly would not be effective and it would inevitably prolong responsiveness of the application. So far plans are produced in a simple and minimalist way, but since it is just plain SVG there are no obstacles for future come detailed plans or even 3D.

Currently floor plans for all floors in old and new building have been created. However, processing of older emergency escape plans resulted in some rooms incorrectly labelled or even not present anymore. On the other side this can be easily solved when improved plans provided that can be easily reprocessed.

5.3 Application Creation

GUI was intended to be created mostly using CSS3 and vectors. Staff photos and background pattern are the only bitmap images present. This speeds up loading process for the first time on all devices and it is particularly useful on machines that clear cached items on each browser closing.

Laravel PHP framework has been used to create application logic. This lightweight framework has features which have helped to keep the development process organized. Access to application resources has been shared via SFTP and uniformity of MVC architecture helped when it came to developers interaction. View is used to display data in form that was previously described. Controller is used to manage user requests based on search bar input, in-application hyper text references or URL data.

Since this application is a front-end it basically behaves as a view in MVC architecture so it can be very easily altered or modified to fit current needs. It is also possible to create entirely different GUI for smart-phones which are very popular nowadays. Currently displaying this application on an older smart-phone with lower resolution can be difficult because of the SVG technology presence. However, new versions of mobile operating systems are rolling out each year providing enhanced support for SVG technology.

5.3.1 User Interaction

Users can interact with application using clicks on viewed items and links, entering search query to search input field or directly via GET requests. Faculty logo can be used to redirect to faculty homepage. Course names and employees' names redirect directly to respective AIS profiles.

Floor Plan

Plan is created using SVG technology. Code is generated by PHP class located application source code providing simple way to edit plans' source an recreate SVG files after. Interaction with plan is managed by JavaScript/jQuery via events. This allows implementation of keystrokes for controlling the plan. Both WSAD and HJKL keys can be used for panning the plan an QZ can be used for zooming it. These can be seen as an addition to more traditional mouse wheel zooming and mouse click panning. All data viewed in plan section are asynchronously loaded for better user experience.

Viewing Room Data

Clicking on specific room redirects to the page of that room where an information view is changed based on a type of accessed room:

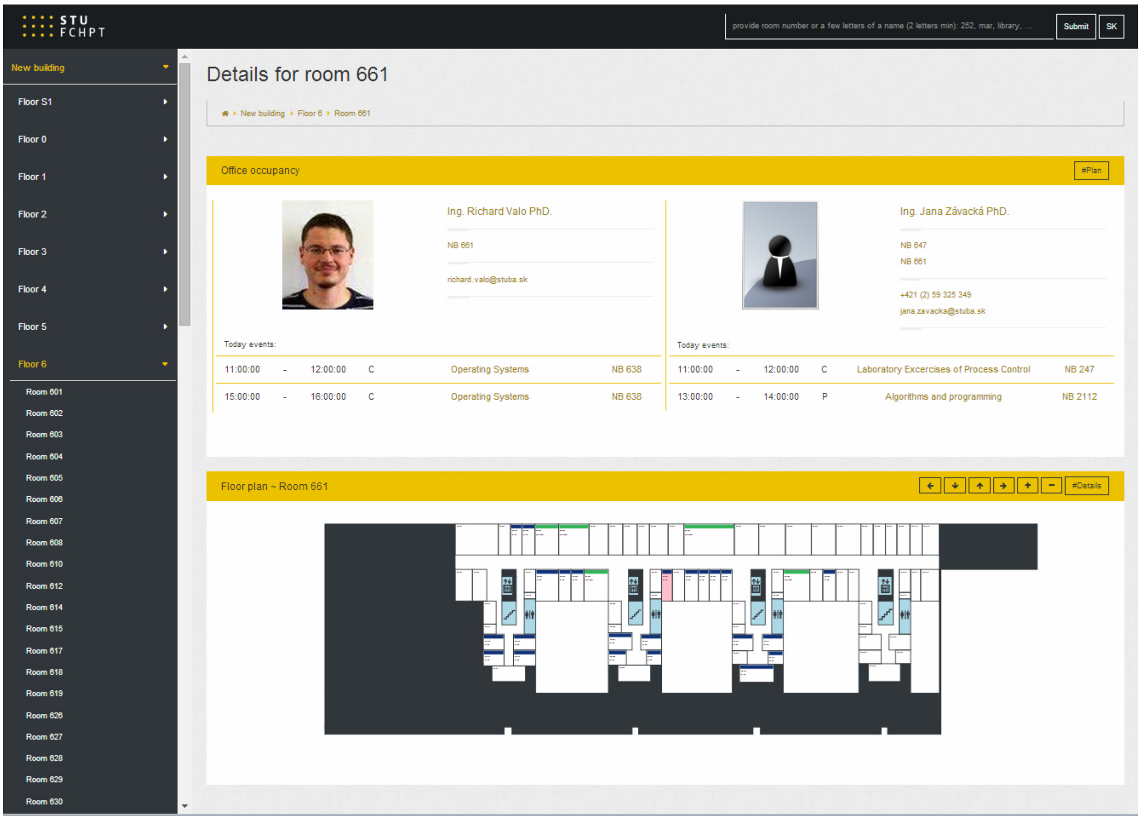
Office (Figure 5.3): There are multiple sections showing multiple teachers in current office. These provide contact information and personal timetable for current day (Figure 5.4).

Lecture Room (Figure 5.5): This view provides simple timetable of events occurring during current week. Lecturers are displayed where possible. This gives possibility to follow link directly to their office for contact information (Figure 5.6).

Other One section that displays information from database if present.

Search Bar

Search input field (Figure 5.7) can be used as an option to interact with application. Executing search query with expressions (Figure 5.8) will result in appropriate action (Figure 5.9). Users are able to search for rooms entering directly the number (e.g. 641, 601) or providing at least part of name of an employee (e.g. valo, bak), or given they search for a room of a specific known name (e.g. library, gym), it is possible to use that as a query as well.



STU FCHPT

provide room number or a few letters of a name (2 letters min): 252, mar, library, ...

Submit SK

New building

Floor S1
Floor 0
Floor 1
Floor 2
Floor 3
Floor 4
Floor 5
Floor 6

Room 601
Room 602
Room 603
Room 604
Room 605
Room 606
Room 607
Room 608
Room 610
Room 612
Room 614
Room 615
Room 617
Room 618
Room 619
Room 626
Room 627
Room 628
Room 629
Room 630

Details for room 661

New building › Floor 6 › Room 661

Office occupancy #Plan

Ing. Richard Valo PhD.

NB 661

richard.valo@stuba.sk

Today events:

11:00:00	-	12:00:00	C	Operating Systems	NB 638
15:00:00	-	16:00:00	C	Operating Systems	NB 638

Ing. Jana Závacká PhD.

NB 647
NB 661

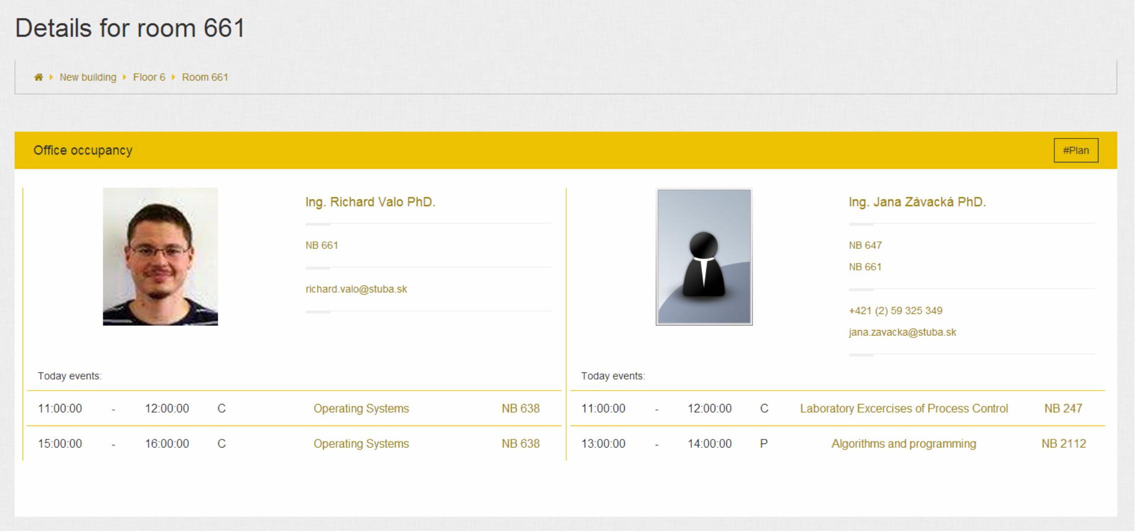
+421 (2) 59 325 349
jana.zavacka@stuba.sk

Today events:

11:00:00	-	12:00:00	C	Laboratory Excercises of Process Control	NB 247
13:00:00	-	14:00:00	P	Algorithms and programming	NB 2112

Floor plan ~ Room 661 #Details

Figure 5.3: Application with an office selected.



Details for room 661

New building › Floor 6 › Room 661

Office occupancy #Plan

Ing. Richard Valo PhD.

NB 661

richard.valo@stuba.sk

Today events:

11:00:00	-	12:00:00	C	Operating Systems	NB 638
15:00:00	-	16:00:00	C	Operating Systems	NB 638

Ing. Jana Závacká PhD.

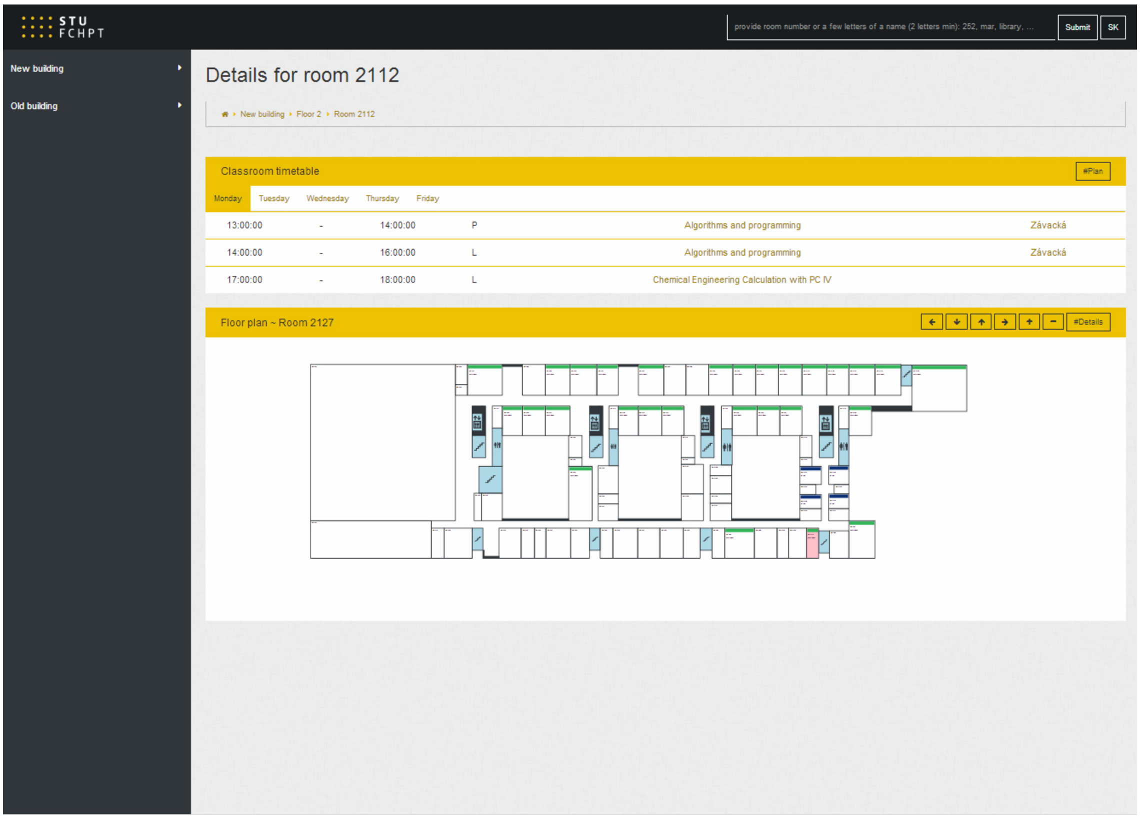
NB 647
NB 661

+421 (2) 59 325 349
jana.zavacka@stuba.sk

Today events:

11:00:00	-	12:00:00	C	Laboratory Excercises of Process Control	NB 247
13:00:00	-	14:00:00	P	Algorithms and programming	NB 2112

Figure 5.4: Detail of office view.



STU
FCHPT
provide room number or a few letters of a name (2 letters min): 252, mar, library, ...
Submit
SK
New building
Old building

Details for room 2112

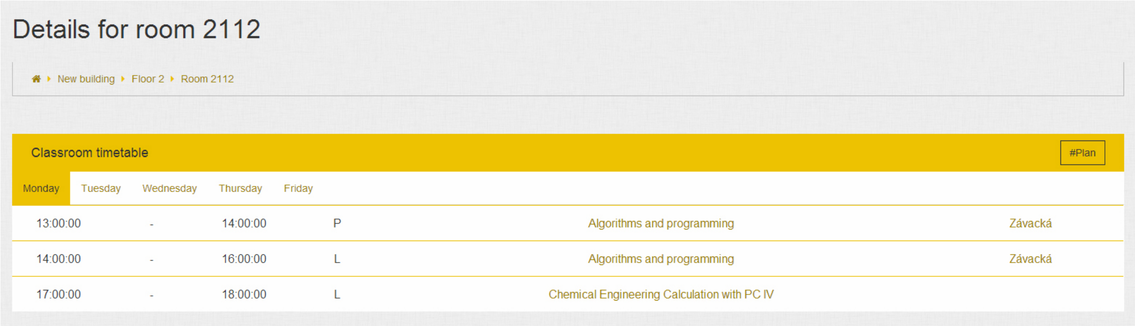
New building ▸ Floor 2 ▸ Room 2112

Classroom timetable

13:00:00	-	14:00:00	P	Algorithms and programming	Závacká
14:00:00	-	16:00:00	L	Algorithms and programming	Závacká
17:00:00	-	18:00:00	L	Chemical Engineering Calculation with PC IV	

Floor plan ~ Room 2127

Figure 5.5: Application with a lecture room selected.



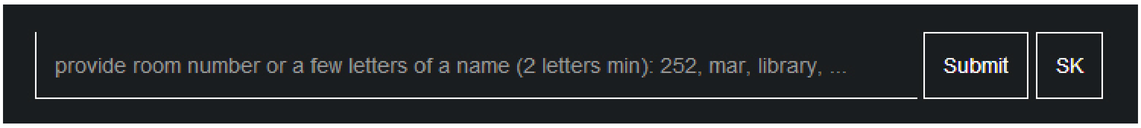
Details for room 2112

New building ▸ Floor 2 ▸ Room 2112

Classroom timetable

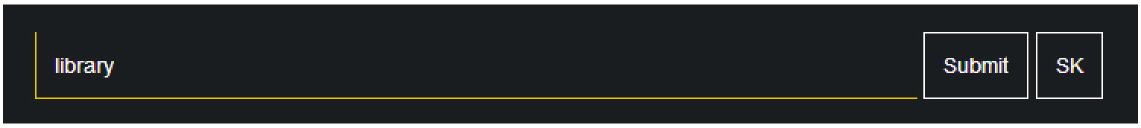
13:00:00	-	14:00:00	P	Algorithms and programming	Závacká
14:00:00	-	16:00:00	L	Algorithms and programming	Závacká
17:00:00	-	18:00:00	L	Chemical Engineering Calculation with PC IV	

Figure 5.6: Close up of lecture room view.



provide room number or a few letters of a name (2 letters min): 252, mar, library, ...

Figure 5.7: Input field for searching.



library

Figure 5.8: Filled input field for searching.

Result is displayed as two column view dividing rooms and employees from which user can select to get further details.

Application Programming Interface

Since this application is meant to behave as a standalone front-end it communicates using created API with back-end. No direct connections to database are present despite the fact that both front-end and back-end are currently located in the same project root folder. This level of independence is planned to be kept.

Data for application views are requested via HTTP requests and they are returned in JSON format which enables easy parsing in PHP. After data are being processed they are offered to user in above-mentioned forms.

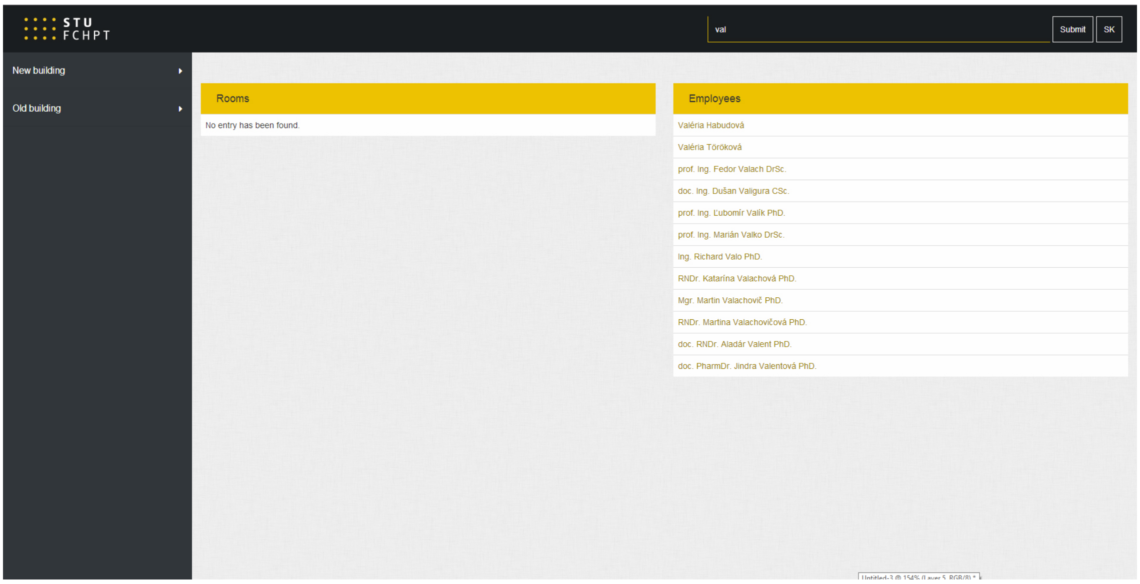
This GUI has its own API functionality as well. It reacts to HTTP GET requests (Listing 5.2) and offers users an option to display the data of the room they specifically need.

```
http://plan.intentio.sk/p/nb-6-675
```

Listing 5.2: Example to show API room focus.

Route calls Controller providing room ID which is then passed to Repository to acquire and format data returning View provided for user.

This way API can be used to link from teachers' user profiles on department website. Using links in information mails instead of plain text is an option as well. Both can prevent students from wandering when searching for a lecture room they have not been to.



Rooms

No entry has been found.

Employees

Valéria Habudová
Valéria Töröková
prof. Ing. Fedor Valach DrSc.
doc. Ing. Dušan Valigura CSc.
prof. Ing. Ľubomír Valík PhD.
prof. Ing. Marián Valko DrSc.
Ing. Richard Valo PhD.
RNDr. Katarína Valachová PhD.
Mgr. Martin Valachovič PhD.
RNDr. Martina Valachovičová PhD.
doc. RNDr. Aladár Valent PhD.
doc. PharmDr. Jindra Valentová PhD.

Figure 5.9: Result of searching for 'val' query.

5.3.2 Language Options

Currently two language options have been offered – English and Slovak. For this to be possible the libraries provided directly by Laravel have shown very suitable. Native support resulted in very simple usage and extensibility.

Particular strings that are used in application and need to be multilingual are stored in associative arrays. Thus we can use keys in these associative arrays in code. The application identifies language based on current URL and it returns corresponding strings which are basically labels in a particular language.

Main advantage of this approach is simple editing of language files. A change is needed only in one text file and it instantly changes values through all parts of the application. Moreover, it is extremely easy to add new language mutation. New language file can be created and all complete currently used array is needed to be translated. This offers developers to handle error messages in multiple languages which is very useful in multilingual environment.

Language switching can be achieved by URL editing (Listing 5.3) in address bar. More convenient method of language switching is offered as a button next to search bar (Figure 5.7). It allows to switch to other currently inactive language. If another language is to be added this button will be replaced by a list of buttons for specific languages for easy selection without the need of scrolling through language options.

```
English:
http://plan.intentio.sk/en/p/nb-6-675

Slovak:
http://plan.intentio.sk/p/nb-6-675
```

Listing 5.3: Example of URL with language selection.

5.3.3 Future Plans

The application is currently in staging state and partially backed by unit tests. One of the future goals is to provide fully tested application allowing for easy additions and refactoring without the need of worrying breaking part of the application.

Spreading SVG and enhanced usage nowadays offers opportunities in creating alternative libraries to interact with plans which could provide better user experience mostly on touch devices since library currently used is still working on the feature of using touch gestures for interaction.

Another big improvement can be achieved by using client side MVC framework like Ember [2] or Angular [1] in the place of views provided by Laravel itself. This would result in much better user experience which can not easily be achieved using jQuery only. Given the site is already fully responsive (Figure 5.10) and styling is basically finished, the transition to such framework would not be complicated.

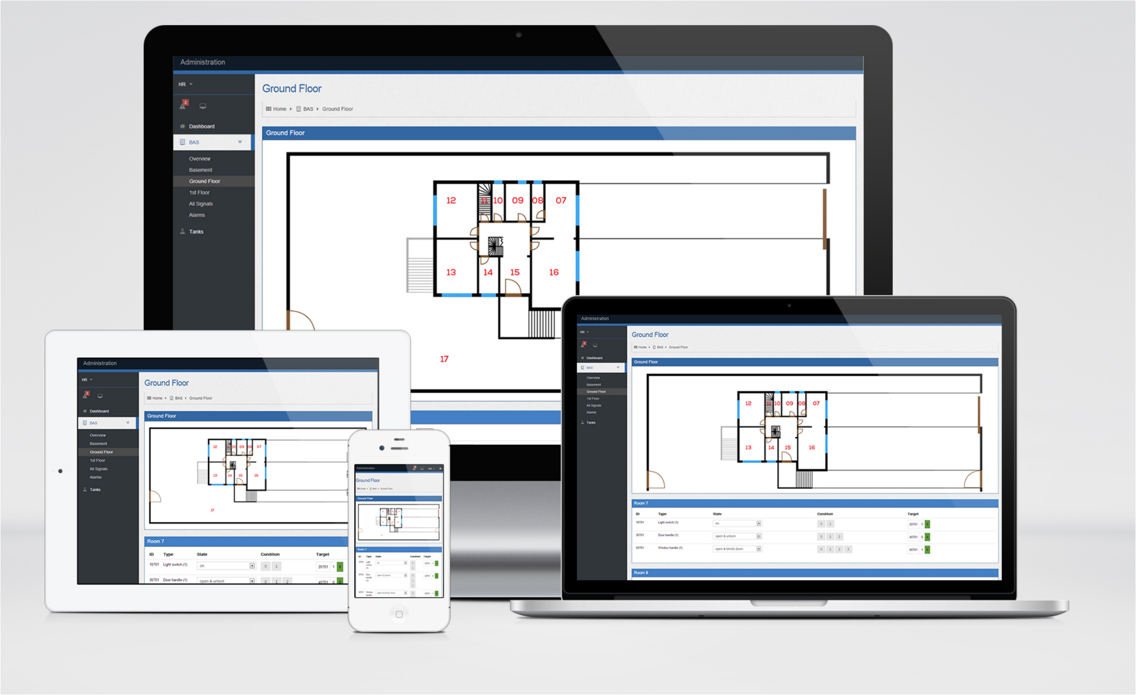


Figure 5.10: Example of fully responsive design. Layout is adjusted with changing dimensions of browser window.

Chapter 6

Conclusions

The web application that has been created is intended to serve as an interactive faculty plan. Traditional Flash technology approach was avoided and widely used web technologies have been used instead. It has been attempted to used all the modern features of such technologies which would result in a web application that could satisfy users and offer them what they expect. Disadvantages of using Flash are not present yet application offers its interactivity.

SVG has been proven to be suitable for building plans and since SVG data are stored in XML files working with this technology is comfortable. I have came up with method how to process faculty fire escape plans and use it to get dimensions and coordinates of rooms. GUI has been created using CSS3 technology which does not require bitmap images to create likeable design. This helps to speed up web page loading process. Moreover, in order to edit design developers do not have to use graphics editor. The whole application is based on MVC architecture. The implementation has been easier because of using PHP framework Laravel that directly supports it. Application is currently partially backed by unit tests and there is a plan for it to be fully tested.

Application back-end communicates with front-end GUI using API HTTP requests that are used for data exchange. Front-end offers it's own API as well. It supports direct referring to room information and location.

This project is currently in staging state which means it is tested in closed circle of users and it will soon be ready for full production deployment. The work on this project continues and it is planned to be released in the beginning of 2014 winter term to offer new-coming students this web application. Staging version of Floor Plan is accessible on http://plan.intentio.sk. Git repository with provided source code and installation guide is located at https://bitbucket.org/hrpd/fchpt_floor_plan_gui.

During development process I have learned many benefits of using new HTML5 and CSS3 technologies, software frameworks and more importantly I have understood the advantages of well-written documentation which both Laravel and jQuery offer. Team work has been an useful experience as well.

Bibliography

[1] AngularJS —- Superheroic JavaScript MVW Framework. Retrieved May 11, 2014 from https://angularjs.org/, 2014.

[2] Ember.js – A framework for creating ambitious web applications. Retrieved May 11, 2014 from http://emberjs.com/, 2014.

[3] Inkscape – Open Source Scalable Vector Graphics Editor. Retrieved May 11, 2014 from http://inkscape.org/, 2014.

[4] jQuery Usage Statistics. Retrieved May 11, 2014 from http://trends.builtwith.com/javascript/jQuery, 2014.

[5] Software framework. Retrieved May 11, 2014 from http://en.wikipedia.org/wiki/Software_framework, 2014.

[6] The world wide web consortium (w3c). Retrieved May 11, 2014 from http://www.w3.org/, 2014.

[7] Borek Bernard. Úvod do architektury MVC. Retrieved May 11, 2014 from http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/, 2009.

[8] Douglas Crockford. Javascript: The world's most misunderstood programming language. Retrieved May 11, 2014 from http://www.crockford.com/javascript/javascript.html, 2001.

[9] John Deacon. Model–view–controller (MVC) architecture. Retrieved May 11, 2014 from http://www.jdl.co.uk/briefings/MVC.pdf, 2009.

[10] Floorplanner. Floorplanner. Retrieved May 11, 2014 from http://www.floorplanner.com/, 2014.

[11] The PHP Group. History of PHP and related projects. Retrieved May 11, 2014 from http://www.php.net/history.

[12] The PHP Group. PHP FAQ. Retrieved May 11, 2014 from http://us.php.net/manual/en/faq.general.php.

[13] The PHP Group. PHP licence. Retrieved May 11, 2014 from http://php.net/license/index.php.

[14] The PHP Group. What is PHP? Retrieved May 11, 2014 from http://www.php.net.

[15] Ecma International. ECMAScript language specification. Retrieved May 11, 2014 from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf, 2011.

[16] Dušan Janovský. Úvod do javascriptu. Retrieved May 11, 2014 from http://www.jakpsatweb.cz/javascript/javascript-uvod.html, 2014.

[17] Robert C. Martin. Principles of ood. Retrieved May 11, 2014 from http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod, 2014.

[18] H. McHeick and Y. Qi. Dependency of components in MVC distributed architecture. In *Canadian Conference on Electrical and Computer Engineering*, pages 691–694, 2011.

[19] Tomáš Mitana. Flash – výhody vs. nevýhody. Retrieved May 11, 2014 from http://grafika.sk/clanok/flash-vyhody-vs-nevyhody/, 2006.

[20] Jordi Boggiano Nils Adermann. Composer – dependency manager for php. Retrieved May 11, 2014 from https://getcomposer.org/, 2014.

[21] Taylor Otwell. Laravel. Retrieved May 11, 2014 from http://laravel.com/, 2014.

[22] Dirk Riehle. *Framework Design: A Role Modeling Approach*. ETH Zürich, Zürich, 2000.

[23] Jon Skinner. Sublime Text 2. Retrieved May 11, 2014 from `http://www.sublimetext.com/features`, 2014.

[24] The GIMP Team. GIMP – GNU Image Manipulation Program. Retrieved May 11, 2014 from `http://www.gimp.org/`, 2014.

[25] W3C. Cascading style sheets level 2 revision 1 (css 2.1) specification. Retrieved May 11, 2014 from `http://www.w3.org/TR/2011/REC-CSS2-20110607/`, 2011.

[26] W3C. What is css? Retrieved May 11, 2014 from `http://www.w3.org/standards/webdesign/htmlcss#whatcss`, 2013.

[27] Zicara. File formats of Vector Images. Retrieved May 11, 2014 from `http://www.zicara.com/techbytes/web-design/file-formats-vector-imagescont`, 2014.

Resumé

Každoročne naša fakulta prijme mnoho študentov, ktorí majú problém s orientáciou v budovách fakulty. Rovnako si toto môžeme všimnúť v prípade návštev, ktorých často jedinou možnosťou je opýtať sa na cestu okoloidúcich študentov. Spolu s mojim kolegom Pavlom Ďurinom sme sa rozhodli vytvoriť interaktívny plán budov fakulty obohatený o možnosť zobrazovania rozvrhov, obsadenia učební a informáciach o zamestnancoch fakulty. Namiesto použitia Flash technológie, ktorá je v prípade vytvárania interaktívnych plánov najpoužívanejšia, sme zvolili cestu použitia rozšírených webových technológií (HTML, CSS, PHP, SVG a JavaScript), ktoré môžu zaručiť interaktivitu Flashu bez zanesenia jeho nevýhod (Kapitola 2).

Väčšina aplikačného kódu vznikla v Sublime Text 2 editore [23]. Ako editor rastrovej grafiky bol použitý Gimp [24] a ako editor vektorovej grafiky Inkscape [3]. Pri vytváraní designu aplikácie bol kladený dôraz na jednoduchosť a prehľadnosť (Obr. 5.1). Hlavnými časťami sú formulár pre vyhľadávanie, informačný pruh a samotný plán vykresľovaný pomocou SVG technológie.

Dôležitou časťou práce bolo získanie dát pre vykreslenie výsledného plánu. Ako zdroj dát poslúžili dodané požiarne evakuačné plány, ktoré boli následne upravené v Inkscape editore. Miestnosti boli prekryté obdĺžnikmi (Obr. 5.2) a takto vytvorený SVG súbor (Ukážka kódu 5.1) bol následne spracovaný a poskytol dáta k miestnostiam, ktoré mohli byť uložené do databázy.

Používateľské rozhranie bolo takmer celé vytvorené pomocou CSS3. Na rozdiel od použitia rastrových obrázkov, tento prístup umožňuje rýchlejšie načítavanie stránky. Či už sa jedná o prvé načítanie, keď prehliadač nemá dáta v udržiavacej pamäti, alebo používateľ vyčistí túto pamäť pri ukončovaní prehliadača. Aplikačnú logiku zabezpečuje PHP framework Laravel [21], ktorý umožňuje jednoduché dodržanie MVC architektúry [7]. Celé používateľské rozhranie slúži ako front-end pre databázovú časť aplikácie, ktorá je vyvíjaná samostatne. Komunikujú spolu pomocou HTTP požiadaviek. Používateľské rozhranie ponúka vlastné API, ktoré umožňuje priame zobrazenie informácií o požadovanej miestnosti (Ukážka požiadavky 5.2).

Plán fakulty je vykreslený SVG technológiou. SVG časť kódu je generovaná pomocou PHP skriptov a celkovú interaktivitu aplikácie zabezpečujú JavaScript a jQuery. Kliknutie myšou na miestnosť presmeruje na stránku prislúchajúcu danej miestnosti a zobrazí príslúchajúce údaje podľa typu miestnosti. V prípade kancelárie (Obr. 5.3) sa zobrazí jeden alebo viacero bblokov, kde každý obsahuje dostupné informácie o vyučujúcom vrátane predmetov, ktoré v aktuálny deň vyučuje. Ak je zvolená miestnosť miestnosť pre pedagogiku (Obr. 5.5), tak sa zobrazí časť o aktuálnych udalostiach v miestnosti. Pri zvyšných miestnostiach sa zobrazujú zodpovedajúce informácie uložené v databáze.

Okrem možnosti myšou je možné aplikáciu – prevažne orientáciu v pláne – ovládať aj klávesnicou (WSAD/HJKL pre posúvanie, QZ pre približovanie a F pre vyhľadávanie), čo určite používatelia ocenia. Premenlivé rozloženie (Obr. 5.10) ponúka zjednodušenie zobrazovania na smartfónoch a tabletoch, kedy používatelia budú môcť využiť maximum z ich rozlíšenia zariadenia.

Aplikácia ponúka voľbu jazyka používateľského rozhrania pomocou parametra v URL (Ukážka 5.3), ako aj tlačidla priamo na stránke. Pri implementácii tejto funkcionality boli využité prostriedky, ktorá ponúka priamo Laravel – jednotlivé prekladané časti textu sa ukladajú ako pole podľa kľúča do prislúchajúcej súborovej štruktúry. Využitie tohto spôsobu umožňuje jednoduchú manipuláciu s jazykovými súbormi, či pridávanie ďalších jazykov.

Aplikácia je momentálne vo verzii pre uzavreté testovanie a je z veľkej časti pokrytá unit testmi. Súčasná verzia Fakultného plánu je dostupná na `http://plan.intentio.sk`. Git repozitár so zdrojovým kódom a návodom na inštaláciu aplikácie pre použitie na vlastnom serveri je umiestnený na `https://bitbucket.org/hrpd/fchpt_floor_plan_gui`.

SVG sa ukázalo ako vhodná technológia na zobrazovanie plánov. V kombinácii s použitými webovými technológiami sme vytvorili interaktívnu aplikáciu bez použitia Flashu. Aplikácia dodržiava MVC architektúru a reaguje na povolené API požiadavky. Práca na dokončení aplikácie pokračuje a plánované zverejnenie je v čase pred začiatkom zimného semestra 2014/2015, aby mohla byť ponúknutá novým študentom. Počas doterajšej práce na projekte som objavil mnoho výhod HTML5 a CSS3 technológií v porovnaní s ich predchodcami, pochopil som výhody a potrebu prehľadne napísanej dokumentácie, ktorú Laravel a jQuery ponúkajú a veľkým prínosom bola aj skúsenosť s prácou v uzavretom tíme.