

Programming Fundamentals I - Group Project

Report 1

December 3, 2014

1 Introduction

The following document aims to be a small introduction into the logic and planning of the group project for Programming Fundamentals 1. The topic chosen is “ *Implement full functioning travel planner for SBB/FFS based on Swiss Public Transport API* <http://transport.opendata.ch> ”.

2 Members

The group will be composed of the following people:

- Pennati Lucas
- Cammarata Davide
- North Alexander
- Mion Victor

3 Planned set of features

Although the task description does not include any details on what is expected, here follow some of the features that we wish to implement during the duration of the project.

- Point-to-point route planner
The user is able to insert a departure station as well as a destination, and the program shows all possible connections. This feature will also have the option to specify multiple additional parameters such as time, date, and special needs.
- Departure Board
There will be an option to specify a departure station, and all subsequent n connections will be shown, again with the possibility to enable multiple options.
- Graphical User Interface
The project will include a GUI. It will be designed as a webpage using HTML/CSS/JS, and the backend (logic) will be taken care by Python. In order to achieve this goal, Flask will be used to intercept GET requests and Jinja2 will be used to serve the modified HTML content. In order to have an idea of what the GUI will look like, Windows Phone can be taken as an example. Essentially the GUI will have the following elements:
 - Tiles interface
 - Simplistic, out of the way elements
 - Expandable entries, which reveal more information
- [Optional] Weather report
Implement a way to check the weather for the destination, either through a direct connection from the point to point planner, or from the main menu.

- [Optional] Ski report
Implement a way to check the conditions of different resorts for the destination, either through a direct connection from the point to point planner, or from the main menu.

4 Design Strategy

In order to implement the planned set of features, the system will have to be designed with robustness in mind, as well as lightness and simplicity. This means that when, for example, fetching data, the program should be able to handle any errors given back by either the APIs, or the HTTP request itself. This idea of robustness is going to be applied to the entire code, in order to be sure that the system will be running even after some major errors and/or problems. Also, the ability to adapt to different screen sizes will be a major point, so to be sure that the code is portable among different platforms and configurations. Also, the program should be executable with the minimal amount of effort, from a module standpoint, as well as from a configuration perspective.

5 Temporary conclusion

As stated in the introduction, this document aimed at giving a simple overview of the ideas behind such a project, and not a full blown explanation. Although many goals have been set, the challenge is going to be to ensure that everything works, and that it is up to a certain standard, even if this means reducing the feature set. If there is any time left, more features will be implemented, but the ones listed in this document are the major ones.