# Programming Fundamentals I - Group Project

Report 2 - Transitor

December 5, 2014

## 1 Tools used

In order to achieve our goal, we are going to use the following modules as well as tools:

- Flask
  Used in order to watch a certain URL, everytime it gets a request, the appropriate function gets called which starts the processing phase. Once the processing is done, it returns the correct content
- Jinja2
  Jinja2 is a tool that is able to substitute tags inside a template document. The module is called whenever python is done processing the requested information.
- HTML/CSS/JS/AJAX
  These web technologies are used to create the interface of the app, being as they provide a very simple way to add an event based system. HTML will provide the structure of the page, and CSS will allow us to design and animate the pages. JS will be used for AJAX loading, as well as other interactive functionality.

## 2 Main Logic of the Program

As explained in the previous document, our program will have different categories, but the logic behind them will be fairly similar. The general procedure will go along the following steps:

1. The user will go to the URL, and the Webapp will open loading the main screen as well as all the possible options
2. Once one of the options has been chosen, the appropriate window is shown
3. The user can insert its search options, and then a request is made using an encoded URL
4. Flask (Python) intercepts the request, splits up the parameters, and passes them to a function
5. A function makes a request to the APIs, which then returns the data
6. The incoming data is stripped of the irrelevant information, and then passed to another function to generate the HTML code
7. Using Jinja2 the HTML is created, filling in all appropriate fields
8. The page is returned to the browser, which displays the results

## 3 Intended Data Structures

There are a few data structures that are going to be used, but mostly the following ones:

- Lists: A variety of lists are going to be used, mostly to hold the retrieved data. As an example the trains leaving from a certain station will be saved inside a list, where each element is a dictionary. This provides a very easy way to access a certain value thanks to their ordered nature.
- Dictionaries: Mostly used in conjunction with lists, they allow a very simple addressing by using a descriptive key, making it easy to know what has to be extracted. They also provide a very verbal way to access data.

## 4 Functions to implement

- Point-to-Point fetch & Stationboard fetch
  These functions will take care of fetching the appropriate data by applying all appropriate parameters when performing the API call. Also, the data will be cleaned up and returned in a container to the function that creates the HTML page.
- Create the HTML content
  This function will take care or receiving the data and create the HTML content. In order to do so, HTML templates encoded in Jinja2 format will be used, which allow for a very easy way to create the appropriate content.
- Animations
  Written in CSS these functions will take care of animating the GUI so to provide a smooth interface without making it feel too rough and clunky.
- Other functions
  The ones described here are the main ones. There will be a handful of helper ones, to make it easier to implement certain features, as well as keep the repeated code to a minimum.