Requirements Analysis and Specification Document (RASD)

This document gives a description of the features involved in the pre-development stage of our Web Application project. Therefore, it will mainly emphasize the specifications that were defined, the selection of our development platform, and the development plan that was followed by our team.

Domain analysis through 5W's

We commence this section by briefly answering the 5 W's and then we move to a more involved description of the requirements and specifications.

- **Who will use our app**: Polimi students from Leonardo and Bovisa campuses are our main targets. So the main stakeholders for the project will be Politecnico students and the University itself. But it can be easily extended to other University environments as well.

- **Why should it be developed**: Students really struggle to find places for studying on campus either when full capacity is reached or when potential study-areas such as classrooms are unavailable because of pre-scheduled lectures. According to the conducted online survey, students welcomed the initiative with a high majority and this gives strong reasons for the utility of the application. Our proposal evidently tackles the mentioned problem.

- **What will it provide**: It provides an interface for students to find suitable study areas on campus, while making use of all the available resources (libraries, common areas and classrooms). The app gives information about the status of all potential options at Polimi.

  It will be realized through a web application accessible to students at any moment from Polimi Online Services.

- **Where will it be used, on which architecture**: It would be used on devices such as smartphones, laptops, or tablets.

- **When and how long would it be used**: It can be used throughout the year. The application is compatible with all the current electronic devices and can continue to run with updates whenever necessary, and thus assisting Polimi students in studies.

Development Platform

A number of programming languages are being used in application development these days and the ones that are more commonly used, were mentioned in the feasibility study document. Being a dynamic website, it demands the knowledge of two separate programming languages for front-end (web designing) and back-end. This makes difficult for us to respect our personal deadlines for the app delivery. On the other hand **Visual Programming** with a right approach could speed up lot of serious and complex development efforts as it lets humans describe processes using illustration.

Our working platform- Bubble: is a visual programming tool and a cloud platform. The programming tool lets you build applications, while the cloud platform hosts and runs it. The editor is accessed through a browser and hence accessible from any device. It also allows creation of database, connection to authentication providers, and building of responsive applications. Moreover, to counter the traditional shortcomings of such platforms, *Bubble* is fully extensible to Javascript plugins. There may be significant limitations let say to build a platform game, but for the task at hand it furnishes enough functionality.

Identification of Requirements

- University students need an application able to show in real time the occupancy of the different study-areas at University in order to determine where to develop their academic activities.
- This problem occurs mostly in Leonardo and Bovisa campuses.
- There are three types of study-areas at each campus: library, classroom and common area.

Derivation of Specifications for the application

Graphical User Interface 1 (before Login)

- In the application, each user is characterized by a username and password. The application allows **Sign up** and **Login** only through a valid Polimi Email ID that is uniquely assigned to every student. The application does not allow the access as it mandates an authentication via Polimi Email otherwise it generates an error message “Use a valid Polimi Email”.

- The application collects the credentials of each user in a database that does not allow the duplication of usernames by users. In case if a user tries to sign up with an existing username, the application generates an error message “The username already exists”.

- The Login section of the app also allows an option to remember the passwords, for the subsequent sessions.

- The password should not be more than 8 characters.

Graphical User Interface 2 (after Login)

- The application displays the username and a **Logout button**.

- The application shows an **Info button** to guide the user about the interface and the various picture / color codes used to define study area categories.

- A **list of study areas** is displayed on the left edge, clicking on a given study area displays the specific information viz. Name, address, occupancy, building etc.

- A separate section of the application allows the user to get filtered results. With respect to **Campus** option, it is possible to choose either Leonardo or Bovisa. Regarding the **Category** choice, either Library, Common Area or Classroom can be selected. Additionally it is possible to introduce the **code of the desired building and/or the name/code of the preferred study area** category.

- Also a **Display all button** is available, that allows to nullify the filtered results and shows a complete list of study areas.

- PoliSAM also obtains the user’s current geographical location and is integrated with google maps that enables the user to **filter results based on distance** (Km radius). If there are multiple study areas on a single address, the map shows all of them and clicking on it displays only those locations.

- In order to facilitate the elaboration of team projects, each study area has an associated **chat room**, that displays specific messages from all the users who are interested in studying there or are studying there. With this feature, teams can decide if the distribution of available seats on a specific study-area is suitable for them or not. A user can only delete his own sent messages. In addition to save database space, messages of each chat room are deleted in every 24 hours.

- Libraries and Common Areas can be occupied at any time by students and there does not exist any schedule for them. For Classrooms, on the other hand, the

availability should be verified by viewing the schedule by clicking on a **Schedule button** that appears only for this category.

- The application counts in real time the number of students using each study area which happens through **Check In and Check Out Buttons**. The result is displayed for each category as “Occupancy / Capacity”.

- Once a user is checked-in in a certain study area, that information is continuously displayed on the right edge as long as he/she does not checks out. If the user logs out without checking out, the check-in status is automatically reverted to check-out.

Underlying assumptions

- **Assumption 1:**

  The intended benefit of the application relies on a disciplined behaviour of the users i.e. they act correctly. Every student who enters or leaves a study area registers his/her presence through the “Check in” and “Check out” options of the application. Also the credibility of the data displayed depends on the accuracy of the information provided by the users.
