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.

Introduction-Domain analysis

As highlighted from the study conducted by HP, RASD contributes to 20% of the entire efforts needed for the development of a software. And domain analysis studies the environment in which the application is used and hence holds a very strong role in the development process. So we need to explore already existing similar softwares, literature, advice from personnel working in the field and current and future requirement.

To correctly analyze the domain our first step was to **browse through similar applications** in the genre of booking systems which familiarized us with the application working environment. Normally the choice for developers could have been to exploit the reusability property of similar existing softwares. But not a viable option for our project as we cannot have our project entirely a derivative of another. Then we started to read **experiences shared by other developers** on web and **literature review** from online sources. This familiarized us more about the working methodology and level of complexity/ ease of each. After that we **discussed with professional software developers** and it was emphasized that for a successful application front-end is very important. As finally user satisfaction is the deliverable we desire, so correct functionality, appealing front-end and ease of use are the properties that we must aim to deliver. Finally and the most important step, we spoke to **Polimi students** to validate our obtained knowledge from above steps and obtain correct domain assumptions to start our development process. The result of this discussion is given below as **5W's**.

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

Identification of Requirements

- University students need an application able to show in **real time 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 and the application should serve all.
- It should be easy to use and operable only by Polimi Students.
- The **time horizon** for the project is 3 weeks.

Stakeholder Identification

Stakeholder Name	Number of Individuals	Critical Success Factor
Polimi Students	20k	Acceptability, Ergonomic/ Ease of use, Correct Functionality, Reliability, Appealing GUI, Robustness
Developers	3	Correct understanding of project scope, Innovative/ Quality, Completion within the deadline,
Politecnico di Milano	An entity	Scalability, Reliability, Robustness, Efficient, Maintainability

Derivation of Specifications for the application

Following the requirement and stakeholder identification, a first draft of software specification was created. The same document was later sent to the stakeholder (Politecnico di Milano- Professor G. Rossi). But later on further contemplation, some additional features were included to augment the value provided by the application. All these specifications are discussed in details in the following:

- In the application, each user will be characterized by a username and password. The application should allow **Sign up** and **Login** only through a valid Polimi Email ID that is uniquely assigned to every student. The application should not allow access through any other email addresses.
- The application will collect the credentials of each user in a **database** not to allow the duplication of usernames.
- It will allow an option to remember passwords for subsequent sessions.
- The application will display the username and a **Logout** option.
- The application will provide an information guide about the application interface.
- A **list of study areas** will be displayed from where information like address, occupancy, building, campus etc can be viewed.

- Application will allow options to **filter results** with respect to campus, the study area category i.e. Library, Common Area or Classroom and the code of the desired building and/or the name/code of the preferred study area category.
- An option that resets the filtered results and display all results will be provided.
- PoliSAM will obtains the user's current geographical location to integrate with google maps and enable the user to **filter results based on distance** (Km radius). If there are multiple study areas on a single address, the map will show all of them and clicking on it displays only those locations.
- In order to facilitate the elaboration of team projects, each study area will have 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 through an available option and it will appear only for this category.
- The application will count in real time the number of students using each study area which happens through a check-in and check-out option. The result will be displayed for each category as "Occupancy / Capacity".
- Once a user is checked-in in a certain study area, that information will be continuously displayed as long as he/she does not checks out. And finally the user will be automatically checked-out in 24 hours if he/she does not checks 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.

- **Assumption 2:**

  Polimi shares the data and allows us to host the application through Polimi Online Services or Polimi Website.

PoliSAM-UML Class Diagram

Domain analysis involves identification of domain objects and actions. Objects are entities that have a desired behaviour or provide some service. Similar objects need to be abstracted as classes. Then we need to define the interaction between objects through the actions.

Library, Classroom and Common Area are defined as **subclasses of study area** as they just have same general attributes as their superclass but some additional ones. GPS is defined as a **composition relationship** with the class Polimi student because its existence depends on the student itself. Google maps is also defined as a composition relationship with the class GUI2 as it cannot exist without it. Post is defined as a **multiplicity-association** with GUI2. Other relationships between the classes/objects are defined in the diagram below.

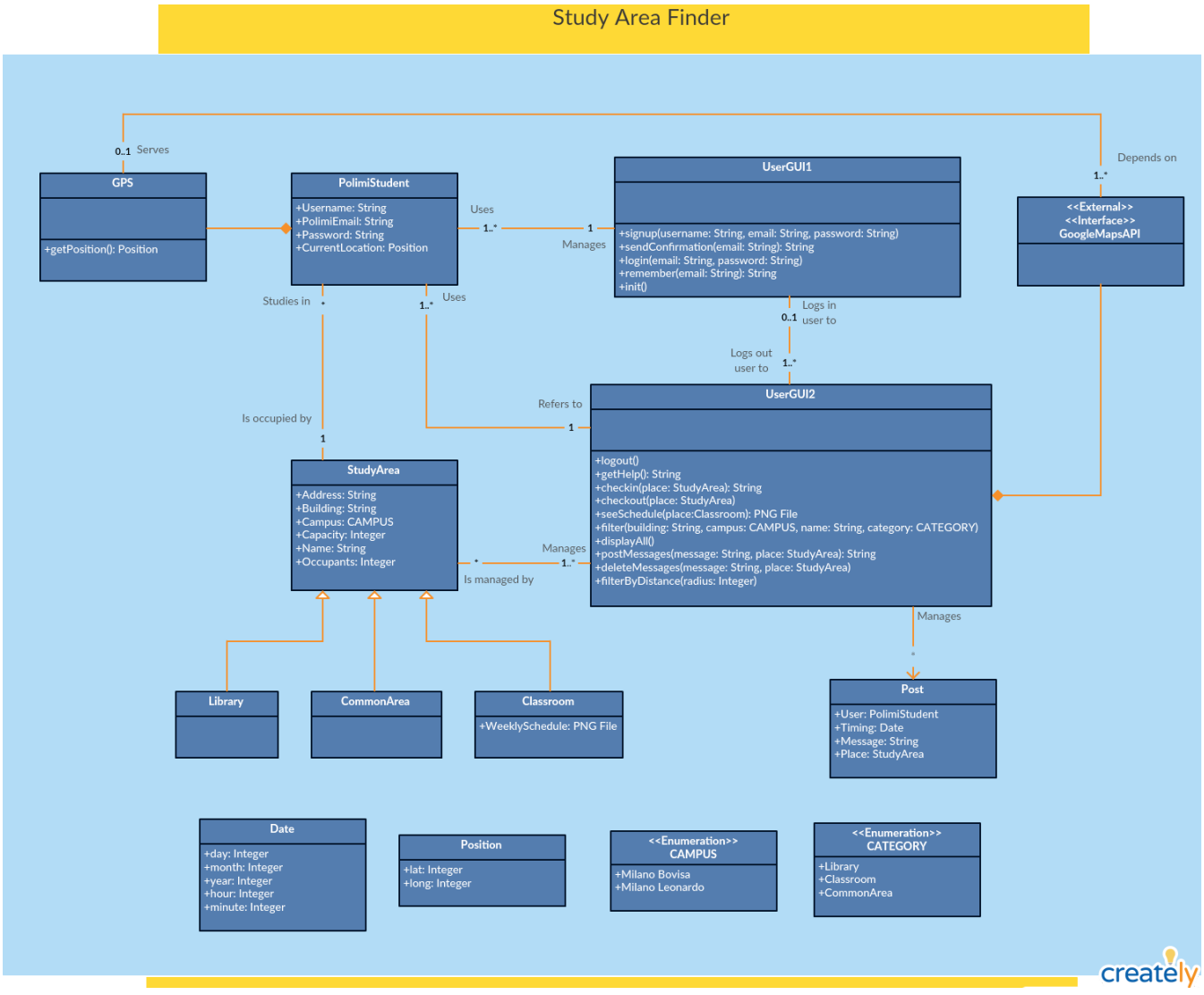

Fig. UML Class Diagram of PoliSAM

PoliSAM-Sequence Diagram

The idea is to describe the interactions between the various objects described above, in the UML Class diagram. It describes actions that lead to change in state by sending or receiving messages during the interaction.

The sequence diagram below explains the interaction of user during sign up, login, check-in, check-out and logout activities. During sign up user interacts with GUI1 and during login interaction is first with GUI1 and then an initialization takes the user to GUI2. While for check-in and check-out user interacts with GUI2 directly while in turn GUI2 interacts with the object study area to update the database. During logout, user interacts with GUI2 and then in turn there is an interaction between GUI2 and GUI1.

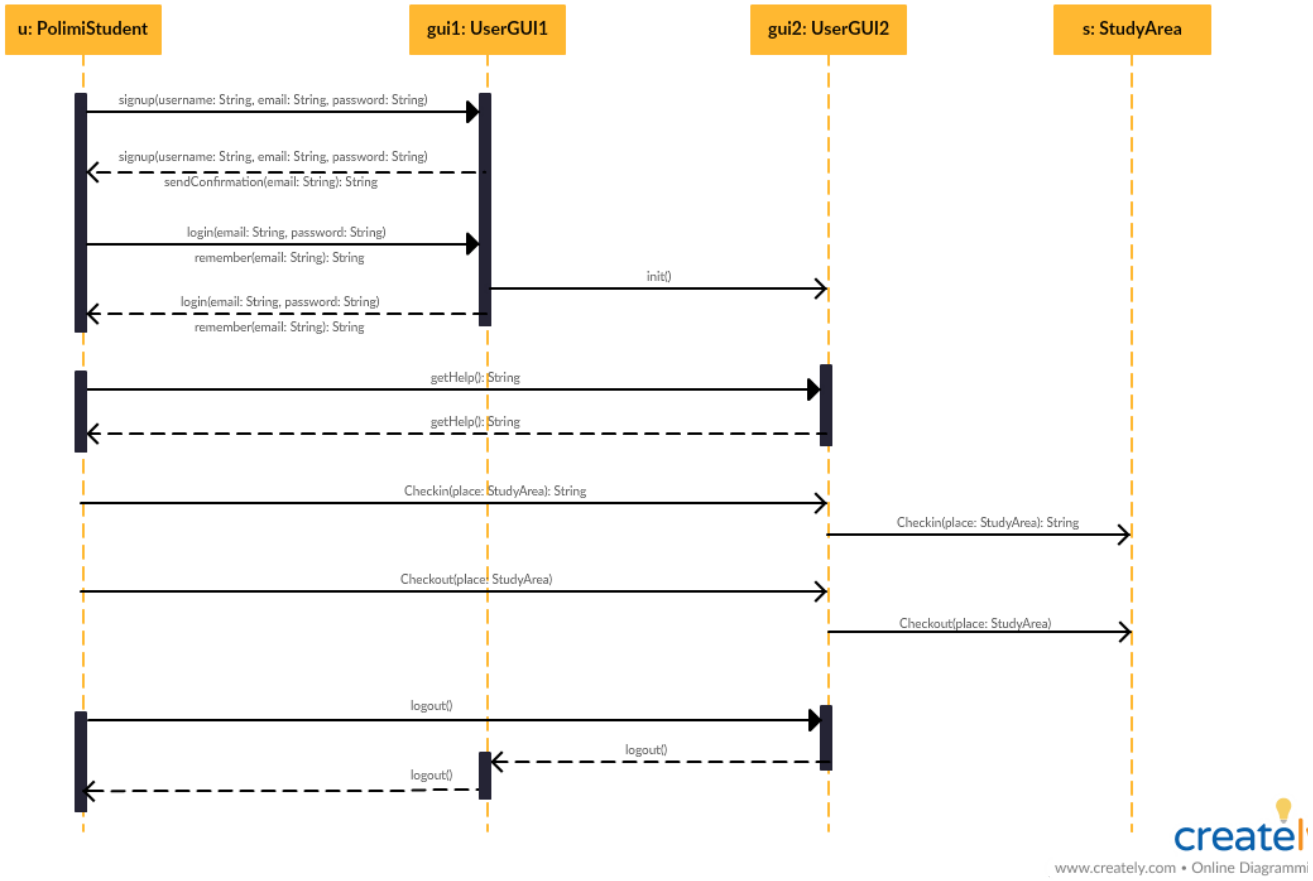

Fig. Sequence diagram of PoliSAM

6. 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 feasibility study, in the list of alternatives. Being a dynamic website, it demands the knowledge of two separate programming languages, one for front-end (web designing) and the other for back-end.

The developers lack a strong base in programming languages used in this domain. Despite the fact that regular programming languages offer huge advantages over visual programming languages and a very strict deadline is a major driving force for making decisions. Regular programming languages do have slow learning curves but they offer a high level of freedom and very less limitations. This compelled us to explore more about the visual programming tools. Already familiar with similar environments like Simulink and others used commonly with micro-controllers like Arduino, we analyzed the possibility of requirement fulfillment through them. **Visual Programming** with a right approach can speed up lot of serious and complex development efforts as it lets humans describe processes using illustrations. But they have some very serious concerns:

- Generate slow code: very difficult to optimize
- They are not extensible: allows to do limited set of things easily but edge cases are far too difficult

But looking at the critical success factors for our stakeholders viz. appealing GUI, completion within deadline and correct functionality among others, we aimed to explore the best possible alternative. One of the visual programming that particularly capture our attention was Bubble, as we found out replicas of web applications like twitter and Airbnb have been created using it. Although we did find some serious limitations like: inability to perform recursion or looping operation on data, writing a custom machine learning algorithm, to design new user experience and can also have some browser compatibility issues. Being a high level language, it has a **shallow learning curve**. Going through the interface we found really appealing features for front-end design and is entirely **logic driven**. We decided to finally use this platform, as we foresee to fulfill our success criteria mentioned above. Moreover, it lets us **focus on the essential concepts learned during the course**.

Our working platform- Bubble: 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, creation of user accounts, real-time-updates, connection to authentication providers, and building of responsive applications. These features can serve most of the functionalities that we aim to build. Moreover, to counter the traditional shortcomings of such platforms, *Bubble* is extensible via Javascript plugins. There may be significant limitations but for the task at hand it provides sufficient features.

7. Distribution of responsibilities

At the project initiation, the literature review and the search for the alternative solutions was handled by everyone. But broadly the responsibilities assigned are given below (the functions are arranged sequentially on the bases of the contribution for each task).

Jean Pierre Sleiman: Design (UI Builder), Testing, Requirement Analysis

Marcelo Noriega del Castillo: Requirement Analysis and Specification Definition, Feasibility Study, Testing

Shobhit Yadav: Front-end Design (Responsive mode of Bubble), Design (UI Builder), Testing

The working methodology was to meet every second day for brainstorming at the commencement of the project. During the design phase, discussions were held every third day to exchange ideas and to solve faced problems. Various short term goals were set for each 3 days so as to keep the project on track.