Feasibility Study Document

This document addresses the current need and provides an overview of our application project. This document studies also its technical feasibility and offers a brief cost analysis for the different alternatives.

1. Initial Problem Description

There is a need among Politecnico di Milano students to identify the availability of spaces in the University study areas for academic activities specially in Leonardo and Bovisa campuses tha hosts most of the students. Currently, there is no tool available that allows them to get this information. Sometimes students end up searching a number of places before finding the right place and consequently, it results in waste of time, dissatisfaction and decrease of productivity.

2. Overview of the Application Project

PoliSAM (Polimi Study Area Manager) is a web application we want to develop in order to facilitate to Politecnico di Milano students the search for available seats, so that they can carry out their academic activities without any drawback. We want to create an application which is reliable and robust users, with low-cost maintenance and that is portable with laptops and mobile devices (smartphones, tablets).

3. Technical Feasibility

PoliSAM consists basically on a web interface that can monitor in real time the occupancy of study areas in both University campuses. There are many programming languages that provide enough capabilities for the development of these kind of interfaces namely HTML, CSS, Python, Java, JavaScript, MEAN-Stack and many other Visual Programming Platforms.

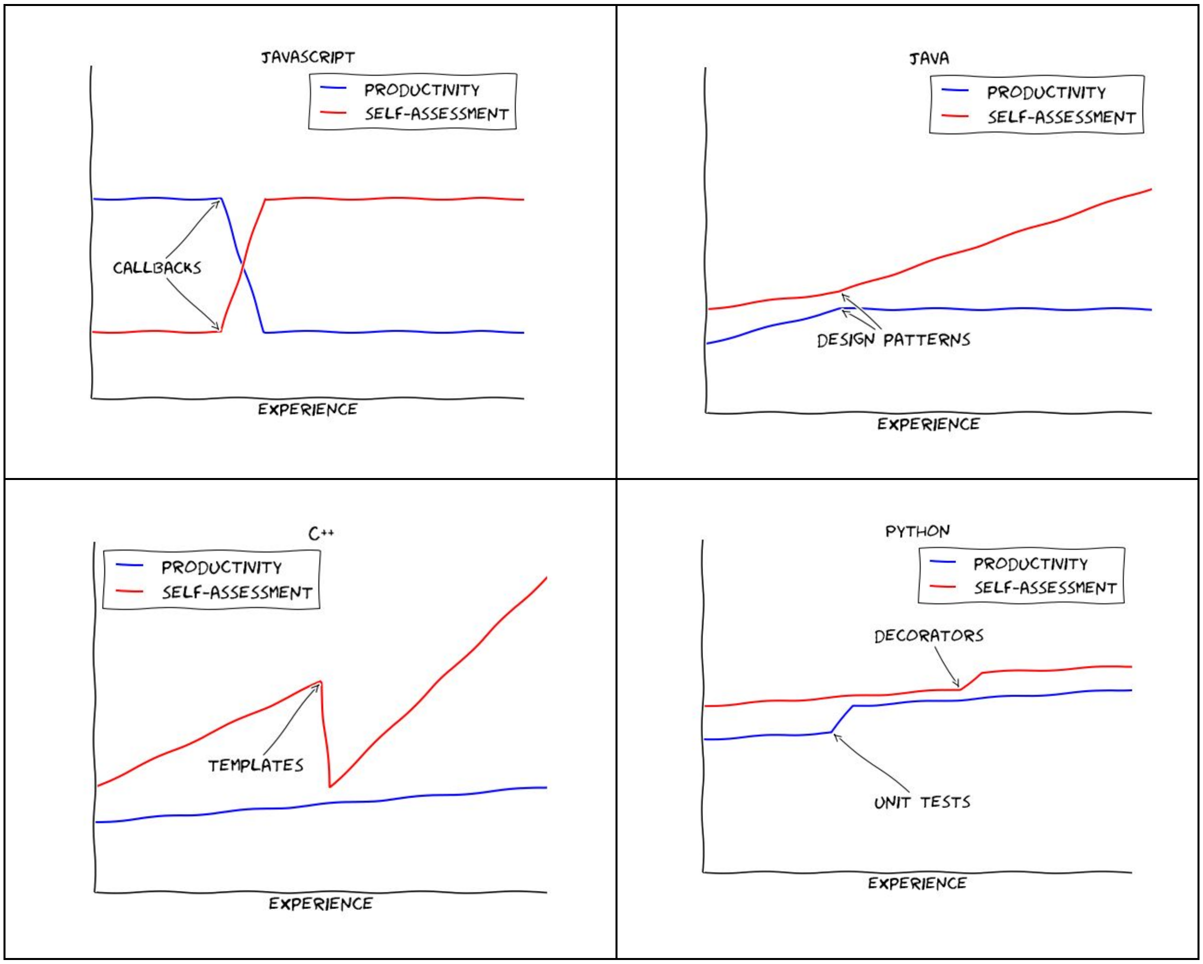
4. Social Feasibility

In the absence of any tool that provides this information at present, all the Politecnico students would welcome **PoliSAM's** deployment. After conducting a small online survey we got an acceptance rate of 92% from the students who frequently use University study areas to perform their activities. It was also highlighted from the survey the ease of usage of the proposed application as it can be easily accessed through Polimi Online Services.

5. Scenarios, Costs and for Alternatives

1) **HTML/CSS/JavaScript for front-end and JAVA/Node.js/Python for back-end**
   Working in these environments is intensive and with the available human resource is not an optimal solution. It also incurs higher cost in terms of man-working-hours and slow learning curve for beginners.

2) **MEAN-Stack**
   An integrated platform of programming languages which allows the development of both front and back-end, although widely used but very programming intensive and we deviate from the course content.



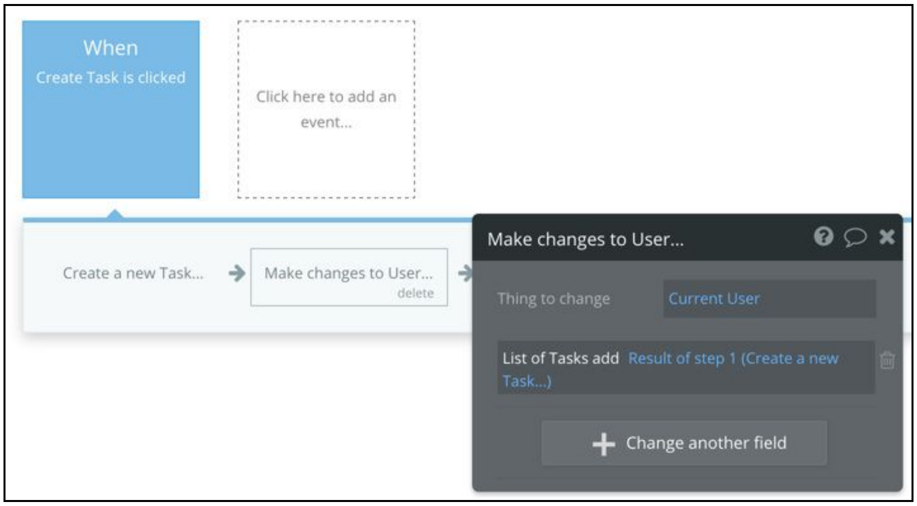
The above learning curves[1] might not capture the true picture as it highly depends on individual experiences. But, for beginners it is more or less as depicted. But, these programming languages are a very strong tool as they provide complete freedom and very less inherent limitations.

[1] The learning curve data is not supported by extensive empirical base, obtained through github

3) Visual Programming

Visual programming enables the development of web applications directly with the definition of logics. It makes the front-end development very easy and appealing at the same time. These consists numerous drag and drop platforms like

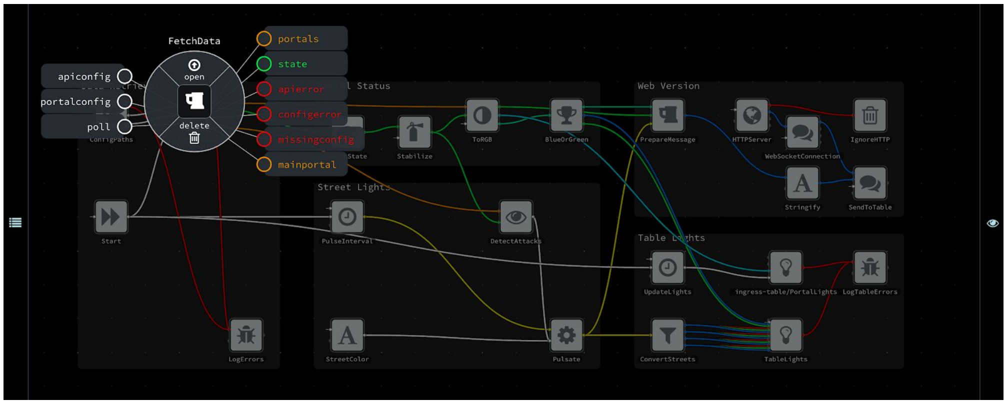
- **Bubble:** a visual-programming language that allows app development without any code. It is possible to develop fully functional applications like marketplace, social network, booking system etc. It also hosts the application on cloud and provides an integrated database. There is also a possibility of installing various plugins and if necessary users can also create their own plugins via coding. Bubble is mainly focussed on **defining Logics.**



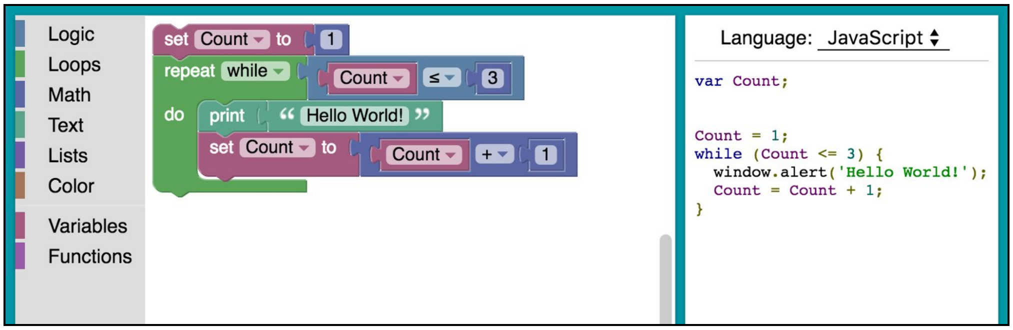
Thing to change Current User

List of Tasks add Result of step 1 (Create a new Task...)

- **Flowhub:** another data drive, visual tool for building software systems. It allows users to manage interactions and logics between entities. A programming paradigm where software is defined through "**black boxes**" that communicate via information packets sent over connections that user has to define visually.



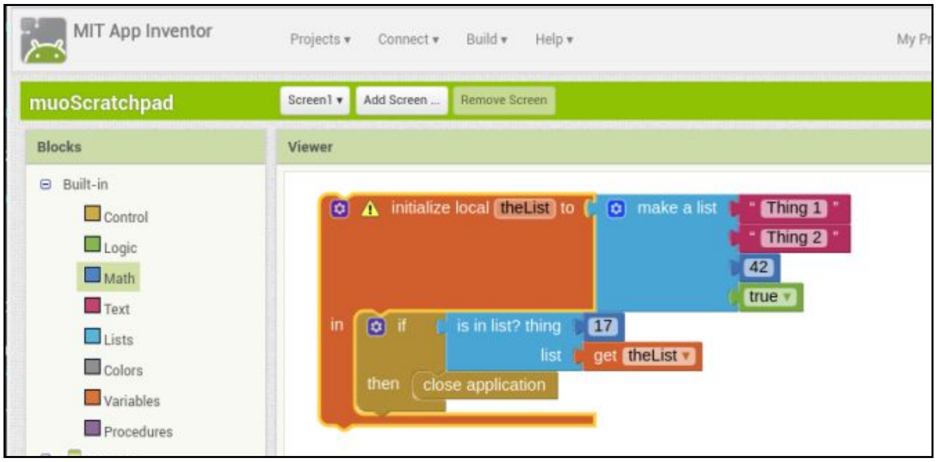
- **Blockly:** an intuitive visual way to build codes and is essentially a text box that contains syntactically correct user generated code. Blockly can export blocks to many languages, including these popular options: JavaScript, Python, PHP, Lua and Dart.



```
var Count;


Count = 1;
while (Count <= 3) {
  window.alert('Hello World!');
  Count = Count + 1;
}
```

- **App Inventor for Android:** It allows to create software applications for the Android operating system (OS). It uses a graphical interface, very similar to **Scratch** and the **StarLogo TNG** user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices.



6. Outcome

There is a non-exhausting list of programming languages (interpreted and compiled) and visual programming languages capable of handling the task at hand. As there are certain constraints with the developers and usually the task also carries some itself, the final decision will be made once the requirements and specifications are clearly laid down. So the selection of the development platform will be discussed in detail in “*Requirements Analysis and Specification Document*”.