**Project Documentation**

**1. Introduction**

Our application is an incrementation game. The concept is of a person who gains money with a tap. There are two main ways of increasing money: changing the incrementation value and changing the income rate. The incrementation value can be adjusted by upgrading the avatar while the income rate changes by hiring friends.

Game Name: Tapitalist

**2. Objective & Target market**

The objective of this game is purely for recreational and entertainment purposes. A secondary objective of this game would be to show how difficult it is to make money.

This type of game covers a very wide range of consumers. Kids starting from ages 3-4 can easily play the game. Students and adults can also play this game when they need a short break or have spare time.

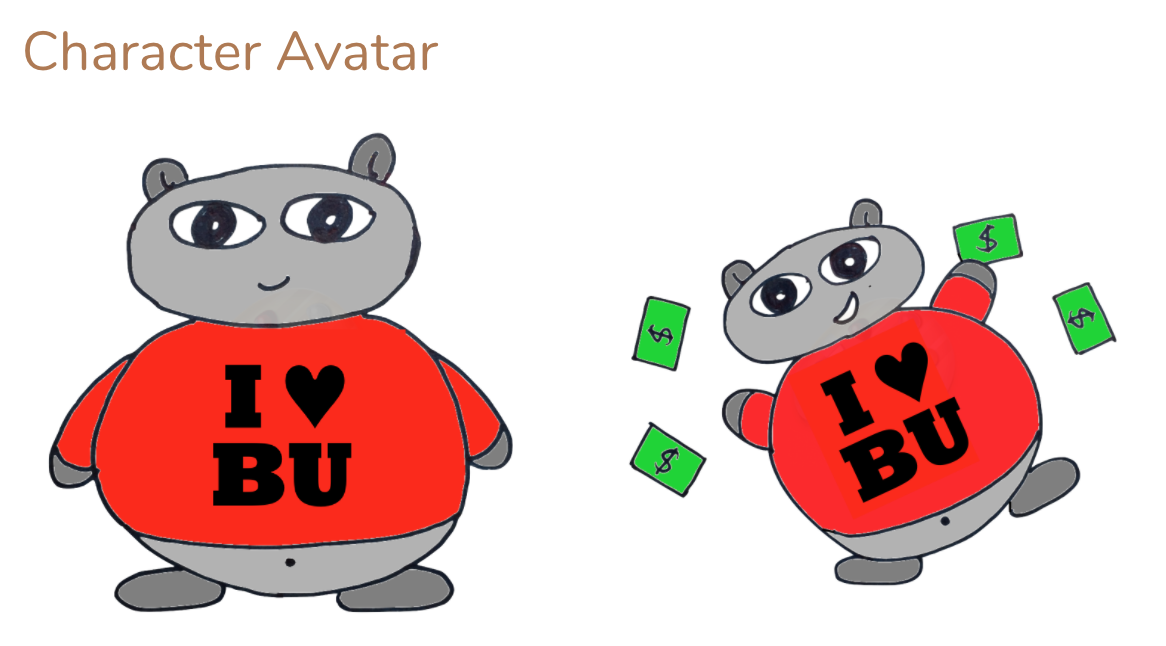
**3.Front End**

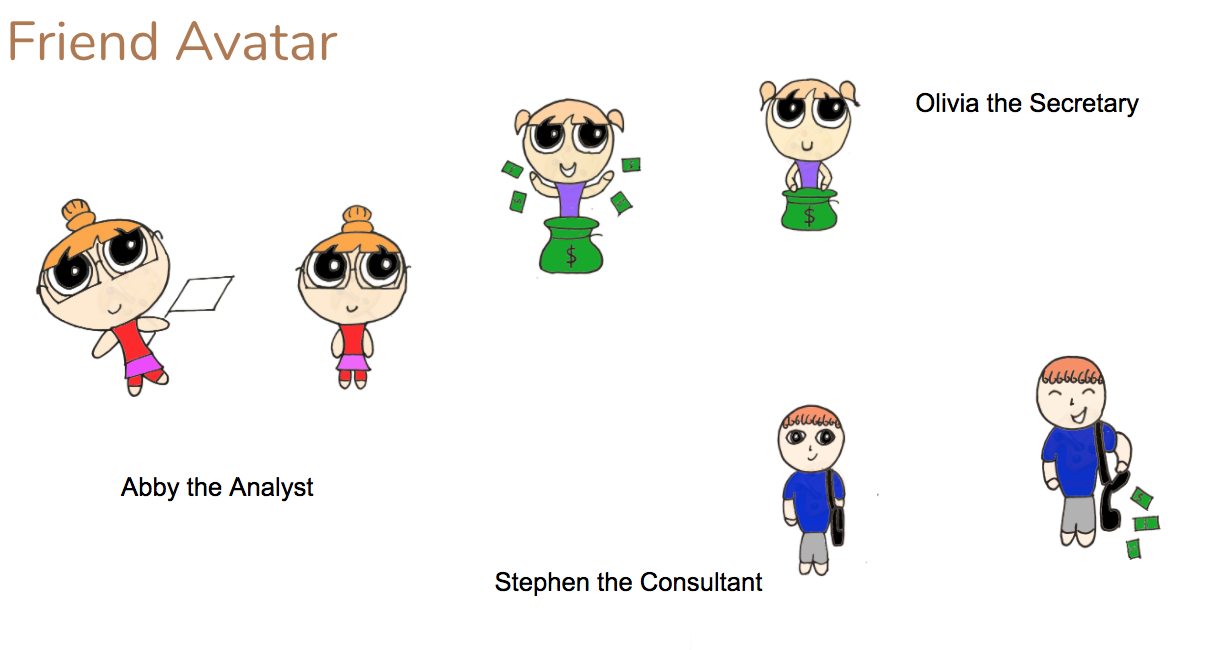
**3.1 GUI**

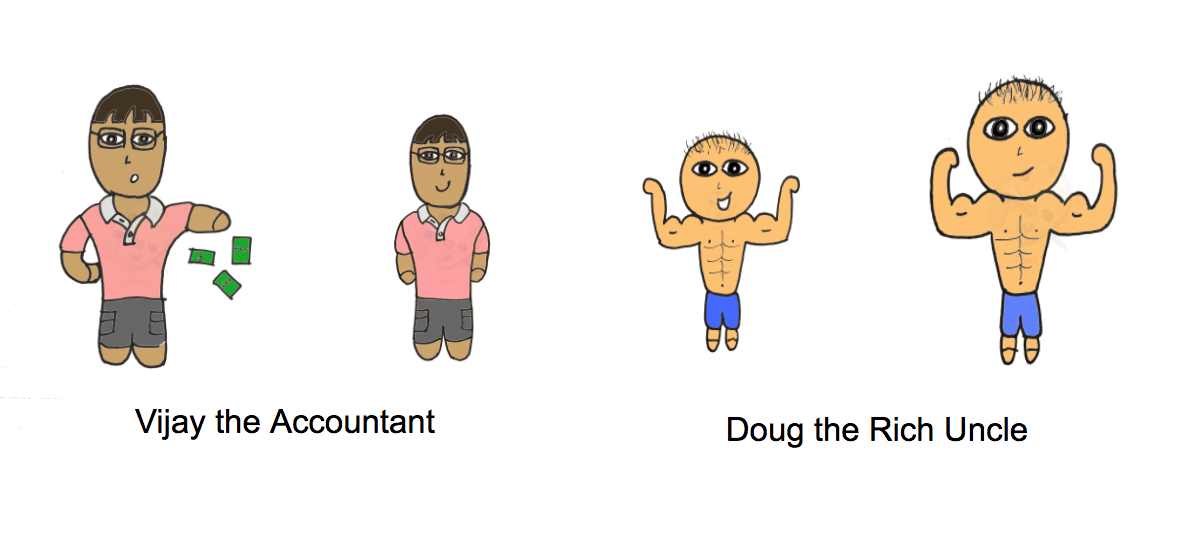
When the user opens the app, the tap to start screen is displayed. Here, the user can press a blinking button to proceed to the next activity.

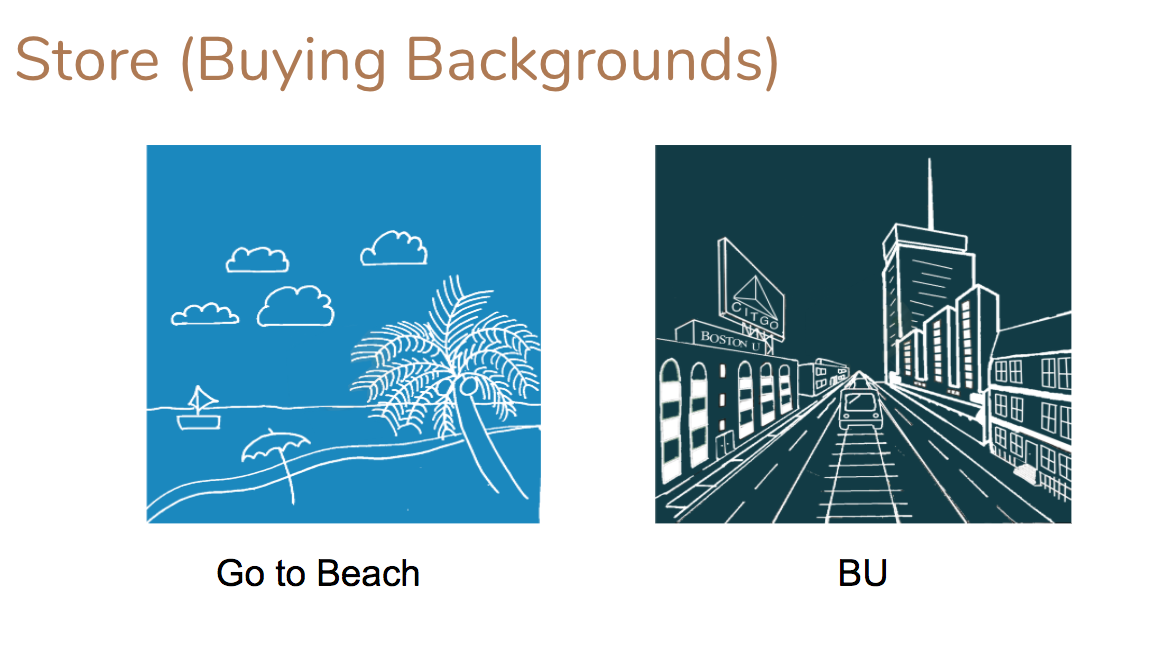
The main activity has 5 buttons and 2 textboxes. The first button is the character avatar. This button is used to increase the in game currency. The second button is a button to display a menu where the user can buy friends to help them increase their in game currency without tapping. This menu consists of the name, level and cost to purchase them. The third button in the main screen displays the tutorial of the game. The fourth button is a menu that displays a few options that the player can explore when they are sufficiently rich. It allows the user to change the background of the main activity to give an impression that the avatar is travelling. The fifth button is used to upgrade the avatar so that each tap is worth more than its previous one. A text box displays the amount of in game currency the user currency has. Another textbox displays the level of the avatar.

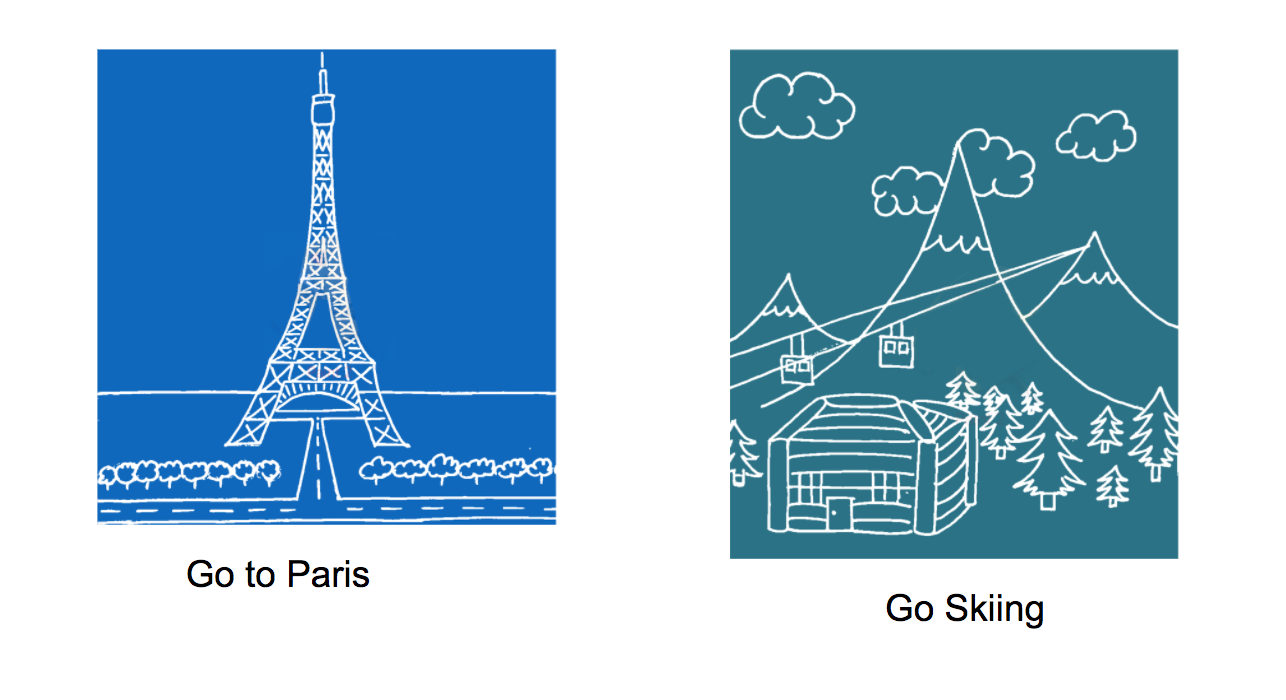
**3.2 Designs (Character, Friends, Background)**

****







****

**4. Back End**

**4.1 Mathematics of Income Rate and Incrementation Value**

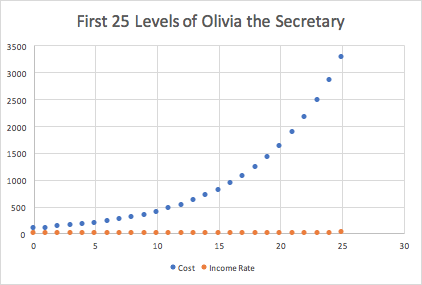
One of the major components of the back end aspect of our project is the mathematics behind the income rate and the incrementation value. Income rate is obtained by hiring friends. Every friend has its own income rate and base cost. The cost and income rates increase as you upgrade the level of the friend. The same concept is applied for the incrementation value. The incrementation value increases as you upgrade the level of the character avatar. As the level increases, the price to upgrade the character also increases. Therefore, the income rate and its price, as well as the incrementation value and its price, are functions of time. Below is the logic for income rate and incrementation value:

Mathematics of the Price and Income Rate (Friends)

|  |  |  |
| --- | --- | --- |
| Friends | Base Cost (Cost at Level 0) | Base Income Rate (Income Rate at Level 0) |
| Olivia the Secretary | 100 | 0.7 |
| Steven the Consultant | 500 | 4 |
| Abby the Analyst | 3000 | 10 |
| Vijay the Accountant | 10000 | 40 |
| Doug the Rich Uncle | 250000 | 400 |

Multiplier Value = 1.15

Below is an example of the price and income rate for Olivia the Secretary for 25 levels:



The graph shows that the price has a exponential function while the income rate will be linear. This is intended to keep the game challenging for the players.

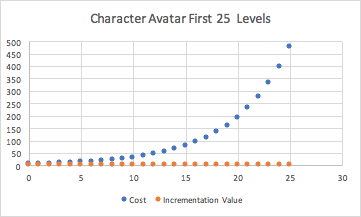
Mathematics of the Incrementation Value (Upgrading Avatar)

Base Increment Value = 0.1

Base Cost = $4.99

Multiplier Value = 1.2

The graph below also shows that the price has an exponential function while the incrementation value for every upgrade will have a linear pattern. This will maintain the difficulty of the game and reduce boredom for the players.



**4.2 Logic Of Timer**

The timer function is based on the Java timer API. Using this API, we were able to create timer tasks and use the scheduleAtFixedRate function to repeat a task at every “x” milliseconds. This function is called once the player hires a “friend” that generates an income rate every “x” seconds. When a friend is upgraded, the timer is deleted using a cancel function and replaced by a new task that generates an increased income rate within the same “x” milliseconds. For our game, we used 5000 milliseconds, meaning that the income rate increases the amount of money by every 5 seconds.