**PROJECT INFORMATION**

**NAME OF THE ORGANIZATION:** Soviet Russia

**PROJECT TITLE:** Wheel of Food

**PROJECT SUMMARY:**

**Objectives:**

This app targets people who are trying to find a place to eat quickly and in the least use time as possible. It also caters towards the indecisive people that cannot decide between a category of food or certain restaurants. This app is not intended to be a stand alone, but as an extension to the very popular food app Yelp.

**What:**

This app will aim to create a user interface that displays a list of preferences that the user wants to input. These are inputted manually or by the meta-tags that we will implement into the software. The app can take in a single meta-tag or many tags that use the union the two or more options. Once the meta-tags have been decided, the app will move the user to another interface that holds the most popular restaurants that fit the meta-tag via the Yelp open API. The user can then add the restaurants to the roulette board up to a later determined maximum. After the user has determined their top restaurants, the app will take the user to another interface that will have the player play a Wheel of Fortune style game that determines the restaurant. The user can also filter the restaurants with the “Open Now?” drop down menu. The game will feature a wheel that has the tagged categories in each slot of the wheel. The first game will be to decide the meta tags of the types of restaurants. After the meta tag has been decided, the second UI will lead the user to another round of the same game except with the top restaurants within the area depending on the meta tag. Once decided, another interface will appear with the restaurant’s information via Yelp with the directions accessible through a “Go There Now” button.

**Goals:**

The goal of this app is to efficiently and effectively determine a food location with an interactive game that allows customization in options and restaurant choices. The app is meant to enhance the quality of life by taking away hindrances such as indecisiveness within a large group of people, since this app allows for the addition of any number of meta-tags.

**PROJECT METHODOLOGY:**

**Approach:**

The project will follow the Agile methodology. We want to avoid Waterfall because it may be time consuming if we’re stuck on one part of the project. Agile allows flexibility within our schedules to meet when we can and work when we can as long as we meet the scheduled deadlines. We also want to avoid the Spiral methodology because we want to get the minimum viable product done as soon as possible. We wish to build on our previous tasks, but we do not want to spend too much time on a single portion of the project.

**Work Breakdown (schedule/tasks):**

The project will take in many factors. There is a need to create several user interfaces such as the mini-game itself, accessing the Yelp API in order to find the restaurants information, communicating with the phone’s GPS system as well as finding efficient algorithms to create the searches within Yelp that do not take up time when finding a restaurant.

**Work Schedule:**

* October
  + Early: Begin planning the structure of the app. Look into class diagrams, flowcharts and UML documentation
  + Late: Begin creating the structure of the UML, removing and adding features that the app may or may not need
* November
  + Early: Begin creating the structure of the class diagrams, removing and adding classes that the app may or may not need
  + Late: Finish UML documentation and focus on class diagrams
* December
  + Early: Finish class diagrams and begin looking into flowcharts for logical purposes
  + Late: Finish flowcharts and begin designing the UI via drawings or taking inspiration from other sources
  + Winter Break: Begin learning Android studio and finalize flowcharts.
* January
  + Early: Finalize Android studio and begin the layout coding of the app.
  + Late: Begin designing and coding the stencil code of app
* February
  + Early: Finalize stencil code. Begin coding logic of app
  + Late: Solidify logic operations and core of app
* March
  + Early: Continue working on logic operations, core of app and begin research and implementing Yelp features into app
  + Late: Finalize core logic of app, logic operations and continue Yelp feature implementation
* April
  + Early: Finish Yelp feature implementation. Begin creation of UI
  + Late: Finalize UI creation
* May
  + Early: Finalize app via unit testing, and overall testing of app
  + Before graduation: Deployment

**PROJECT TIME-FRAME:**

**Planning:**

When planning for this project, we want to create UML diagrams and begin to think of pseudocode for certain parts of the project. The main focus of this project is to create a very friendly UI for the user as well as making it simple and elegant without adding too much “fluff” to try and entice the user to use the app.

**Coding:**

Because we are creating an app, our team has decided to use Android Studio for creating this app. We originally wished to use Swift for the iPhone, but quickly realized that Android Studio would be a far better choice for this particular project. We want to be able to quickly start coding the project and not have to wait for Apple to give us permission for their app store.

**Testing:**

When testing this app, we will need only one thing: an Android device. However, we wish to test the app on different brands of Android devices to ensure that our app works for a large variety of Android devices.

**Deployment:**

Naturally, we will be deploying our app to the Google Play Store because we are using Android studio.

**PROJECT RISK MANAGEMENT:**

**Risk Register:**

During the course of this project, we will come across many risks that will affect the progress of our project.

1. Series of bugs that cannot be solved
   1. In order to overcome these bugs, we will have to talk to the team in order to find ways around the bug, whether that be to redesign the section of code that is causing the bug, work through the bug and try to find a solution or get rid of the feature overall if the bug persists in other sections of the project.
   2. We have to keep in mind our timeline when trying to redesign the solution if we run into problems that cannot be solved in an efficient way. If redesigned, we will have to design a feasible solution without it consuming the allocated time we have to the particular section of the project
2. Inability to finish project
   1. In the event that the project cannot be completed with the current features, we will remove features we find that are the most time consuming. We wish to implement the minimal viable product, or MVP. After completing the MVP, we want to optimize the code we have but if the optimization causes more bugs as well as leads us to not complete the project. In the instance this occurs, we will not optimize the code and deploy it as the MVP.
3. Inability to collaborate
   1. In the event that the group members are unable to collaborate, the team leader will bring these concerns to the project manager. However, the team leader will try to suffice and communicate with the team try to fix the miscommunication

**PROJECT CONTACTS:**

Kyle Ho

James Dinh

Ashwin Bholse

Alex Han

Kevin Nguyen