Noah Howell Milestone 1 Tuesday, February 2, 2021

Team Name: Foodie Fighters

For this milestone, I will provide what key tools, technologies, and the process model used for this course project. Below, are all 3 of those, divided into different sections to provide a structured layout of the plan as of now:

Key Tools:

- **Visual Studio Code:** IDE for coding the app.
  - **Reasoning** - The Visual Studio Code IDE is efficient for coding. It does not take up extra clutter in the project folder like Microsoft Visual Studio. It can also run on Linux easily, in case we want to port our software to Linux. It supports C# and is maintained by Microsoft, which means it should smoothly integrate with C# and be easy to use.

- **UWP Apps**: Output for the project.
  - **Reasoning** - is that it allows us to build apps for any Windows device. This platform provides a common type system, APIs, and application model for all devices that run Windows 10.

- **Github**: used to set up and embed various versions of our app online.
  - **Reasoning** - is because it is free for public usage and it is easiest to use for my group collaboration. Not only is it easy to maneuver, but it is also convenient for the TA's and Professor to watch our group's progression throughout the design/project. Lastly, it's hosting-service provision as well as virtual peer property could both prove to be highly beneficial factors for us to use later in the project.

- **Google docs**: Documentation for the app.
  - **Reasoning** - it will facilitate user experience with our app, which contains references on navigating the app. We will provide screenshots of every problem the user may encounter while using the application.

Technologies:

- **C#** - An Object Oriented Language that is simple and intuitive for designing apps. We all already have C and C++ experience, and some of us have C# experience. C# is much easier to develop a GUI for than C and C++. Our restaurant app will be very GUI-oriented so this is a strength for us. Its strong Object Oriented nature also helps us in the modelling stage --we can lay out exactly what classes we want and plan out how they should interact.

- **Discord** - A platform for meetings and communication. We can hold group calls, post code we're working on, post screenshots of our progress and trouble we run into, and we can share our screens. It does everything Zoom would do but it's more permanent than a Zoom room. The Discord server is up 24/7 and so each individual can post updates to the rest of the group at any time.

- **Why Github? Why not something else? (BORIS)**
  "Main reason would be version control. Everything is stored on a remote server, keeping the developers up-to-date with the current version they are working on, and also allowing for different branches so that the changes would not interfere with each other. Git allows you to undo mistakes so that you can backtrack if you feel you ever made a mistake. Then, the second most important reason, it doesn't force you to change the way you work, and lets you do the work normally, but just don't forget to commit so that the newest version of the files are uploaded."

Process Model:

- **The process model our team will be using is the V-Model (V&V).**
  - A sequential process model, so that I will be able to do various instances of testing during the modeling and construction framework activity. The reasons for this decision are as follows:
    - High quality is a crucial point to this design. During the code implementation, we want to be able to directly test each section to be sure every function works as it should, which leads to higher chances of success and better code quality.
    - During the application usage, because our model is so high-quality-based, we won't or will rarely come across bugs or unfamiliar programming instances in the system.
    - LALALA😎😋😊😁😌😈😈😈😈😈

- **I am aware however of the drawbacks that this process model has, such as:**
  - It is quite rigid - inflexible.
  - The software is developed during the implementation phrase, so no early prototypes of the software will be produced.
  - If changes occur midway, then the procedure and test documents will have to be updated.

This is what we think are the appropriate tools, technologies and the model that is necessary for us to get the project done (for now).