Problem Statement:

A Better Betting Pool App

Prepared by TEAM C:

Will Blankenship, Ryan Manus, David Rosenberg

October 1, 2017

1. Project Description

The purpose of this project is to build a “better betting pool app,” specifically focused on the 2017-2018 Southeast college football schedule. Yet it should also be expandable to work with any sports schedule characterized by two opposing teams where a point spread is predicted for each scheduled game.

2. Objective and Success Criteria

This project will be considered a success if the team delivers a functional, reliable, and well-documented app, and does so within the scheduled time frames specified in class. The project software should be easy to use and intuitive.

3. Vision

For the sports enthusiast, or any betting pool participant, regardless of experience with betting pools, the Better Betting Pool app will be a simple yet exciting method of keeping track of betting pool winners and losers, based on a predetermined college football team schedule and predicted point spreads.

4. Project Scope

The Better Betting Pool app will use a database to take in data concerning scheduled games and point spreads. There will be a deadline for users to make their entries prior to the games. “What, when, why, where, who, how….”

5. Assumptions, Dependencies, and Constraints

Assumptions:

The user will have a stable internet connection to submit his or her pool with. Users will submit their pool before the assigned deadline. The software will have an interface, and will have to be easy to use.

Dependencies:

Since this project is complex, it is dependent on our efficient use of Django, an open-source web framework written in Python, to ease the creation of a complex, database-driven website.

Constraints:

All members of the group need to review the success criteria before final submission. The project must be completed by the end of the semester. The project will be designed in increments as the three members learn Python and Django from scratch. We are constrained by the guidelines provided by the end user.

6. Itinerary

(Over the entirety of the project learn more about python and Django)

(Our team plans to meet twice a week to keep ourselves in check)

10/1/17 – 10/15/17

- Determine what the consumer wants.
- Ask questions about the software project to determine what should be done
- Begin a formal design and start to mess with the software.

10/15/17 – 10/29/17

- By this time, we should really know what we are going for and we should be vigorously developing the software.
- During this time, we'll have our first sprint.
- We'll get to experience a bit of agile software development and the bulk of our work will be started.

10/29/17 – 11/12/17

- During this period, we'll be working on the bulk of our software.
- Most of the editing and troubleshooting will be done here.
- We should be at about 75% done at the end.
- We should have a good working interface.

11/12/17 – 11/26/17

- We will be finishing our project up in this period.
- The design and interfacing of the software will be complete and working well.
- We will only be making small enhancements to make the software work and look better.

11/26/17 – 12/05/17

- At this point we will be working on preparing for our project presentations
- The software will be complete for the presentation.

7. Risks

The primary risk of the project will be time management, i.e. not meeting when we should as a team. All the team members should invest time into learning Django and Python. If we don't invest enough time to studying the two, our final project won't be successful.

To make sure we have good time management we should stick to our own itinerary and hold each other accountable to our positions in the team. We should also keep in mind all the various resources that are available to us such as the one's listed below.

8. Resources

We will have plenty of online resources such as videos and tutorial pages that we can use to get the project completed. Dr. Bartz will also be a helpful resource for determining how the software should look and operate.