Requirements Documentation

8th-Wonder

The project we intend to build is a web application that will run a Putt Putt Golf tournament. The users for our application are managers, players, sponsors, and drinkmeisters. This project will be made using the Django framework in Python. For more information about the project please see the Project Plan.

**Meeting Time**: Our team will be meeting every Monday, Wednesday, Friday from 12:20 - 1:00. The lengths of these meetings will vary depending on how much we have to discuss about our project.

**Functional Requirements**: These requirements may be subject to change as we develop our application

F1. User Authentication and Access

* 1. The system must require all users to authenticate themselves before they can access the web application.
     1. The user must login using their email address and password. If the user has not created an account, they will need to do so. Each account must contain the user’s name, email address, phone number, and user type (Sponsor, Player, Drinkmeister, or Manager). The user must also create a password for their account
     2. The user must be allowed access if they enter their email and password in correctly otherwise, they must be prompted to try again.
  2. Each user should be able to do different things within the system based on their user type as described below.

F2. User Features

1. The system must allow any user to modify their own password and email address.
2. The system must not allow anyone other than the manager to view other accounts
3. The system must allow any user to view their own bank account and add money into it or withdraw money out of it. There will be no use of real money in our system.
4. Every user must be able to view the drink menu and order drinks if they have enough money inside their bank account.

F3. Player Features

1. The player must be able to purchase entry into the tournament.
2. The player must be able to track his or her own score as well as what hole they are currently on.
3. The player must receive a prize if they finish 1st, 2nd, or 3rd place once the tournament is over.

F4. Manager Features

1. There must be only one manager in the system at any given time.
2. The manager must be able to verify sponsors and drinkmeisters after their accounts have been created.
3. The manager must receive all the money from the Tournament. They must also receive the money after a user orders a drink.
4. The manager must be able to update the drink menu as they see fit.

F5. Drinkmeister Features

1. The Drinkmeister must have access to the drink menu as well as a recipe for each drink so they can make it.
2. The Drinkmeister must be able to see all current orders for drinks and must be able to alert the system when they are ready to be delivered.
3. The Drinkmeister must only begin making drinks after they have verified that a user can pay for the drink they ordered otherwise, they must tell the user that they need more money to purchase a drink.

F6. Sponsor Features

1. The Sponsor must provide prize money for 1st, 2nd, and 3rd place.
2. The Sponsor must be able to select what Tournament they would like to sponsor such as Monday’s Tournament or Thursday’s Tournament. After a Tournament has been sponsored it must be renamed after the Sponsor.

**Non-Functional Requirements:**

NF1. The system must use a database

* 1. One database must store user account information including the following: Username, Password, Email Address, Phone Number, and Account Balance
  2. The System must have another database for the drink menu which must include the following: Drink Name, Price, Recipe, which user ordered the drink, and where the user is currently at.

NF2. The team must use GitHub as a remote repository

NF3. The system must be deployable by a local host or cloud service.

NF4. The system’s interface must be mobile device friendly