

Tournament App

1. Introduction and Context:

During large disk golf tournaments, effective organization can be difficult to achieve. To remedy the issue, this project's goal is to build an application that permits users to access only what they need for their roles in the tournament.

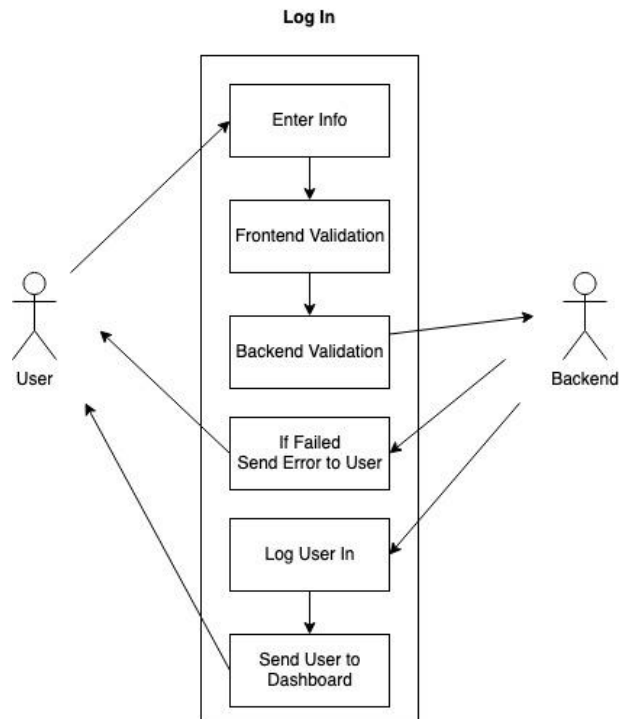
The application will allow users to define their role for all upcoming tournaments, as well as sign up digitally for any tournament designated. Users who plan out these tournaments have the option to use administrative tools to set up tournament dates, prize money, and other user roles.

The application will allow users to login to track their own score as they advance through the tournament. A leaderboard will be established using live data, which will be posted at the conclusion of each tournament.

2. Users and their Goals:

The following Use Case diagrams will describe the application's actors and their interactions.

Figure 1 - User Log in



Participating Actors: Default User, Managers, Drink Meister, Sponsors

Entry Condition:

A User wishes to sign in.

Exit Conditions:

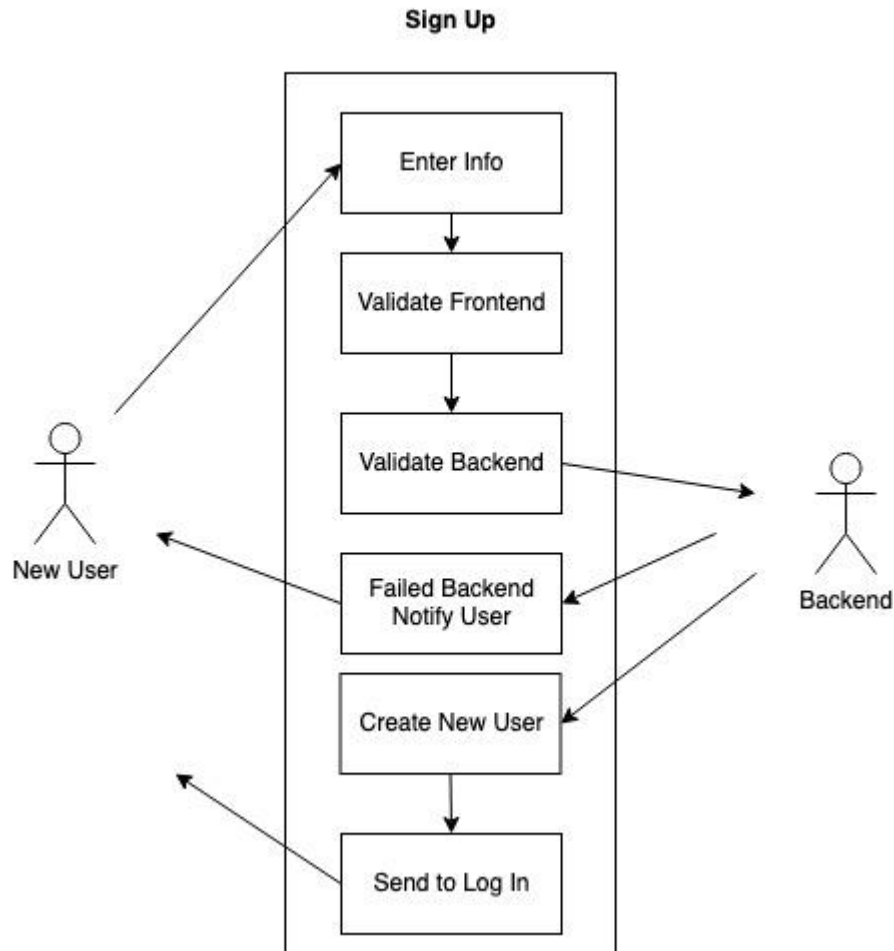
User logs in to dashboard

User is denied access to application

Event Flow:

1. User enters username and password
2. Application verifies user
3. Log User In to Dashboard
 - a. If incorrect login, send error and retry.

Figure 2 - User Sign Up



Participating Actors: New User, Backend

Entry Condition:

User selects 'Sign Up' option

Exit Conditions:

User is denied new sign up

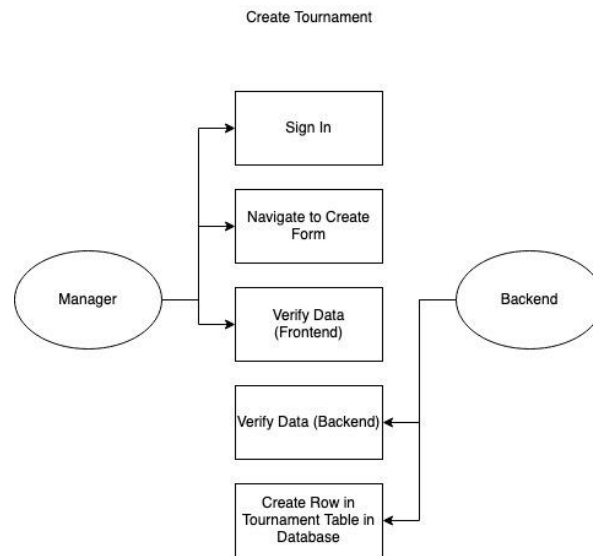
Pre-existing account

Application User is created

Event Flow:

1. User enters authorization info
2. Backend validates input
3. Creates new user
 - a. If failed, notify user to retry
4. Open User Log In page

Figure 3 - Manager creates tournament



Participating actor: Manager

Entry Conditions:

Manager wants to create tournament

Exit Conditions:

Manager verifies data for backend

Backend creates tournament in database

Event Flow:

Manager Signs In

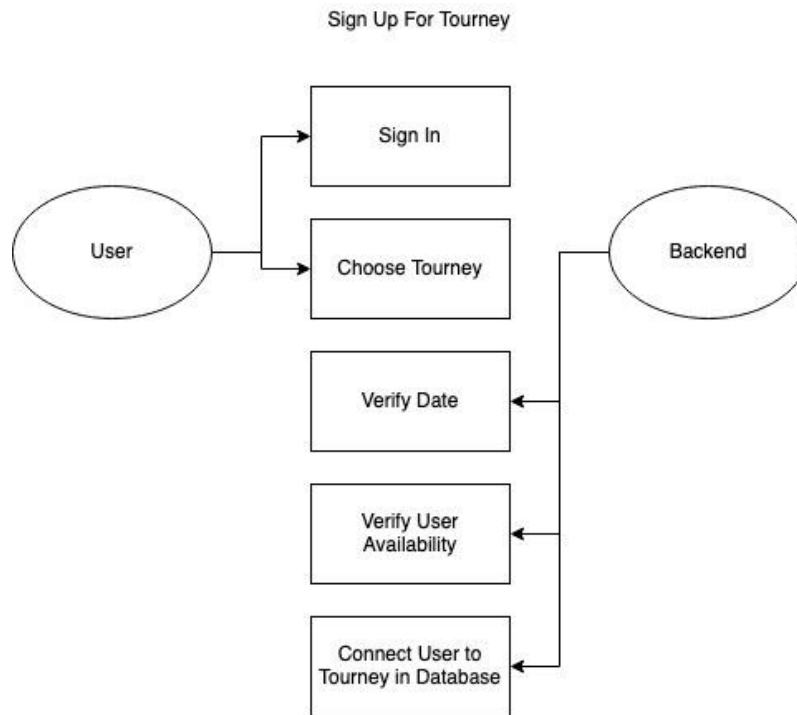
Manager Navigates to Tournament Creation Form

Manager completes all data for the tournament being submitted.

Application accepts all the data being submitted

Application adds tournament to application

Figure 4 - User Tourney Sign Up



Participating Actors: User, Backend

Entry Condition:

User desires to sign up for a tournament

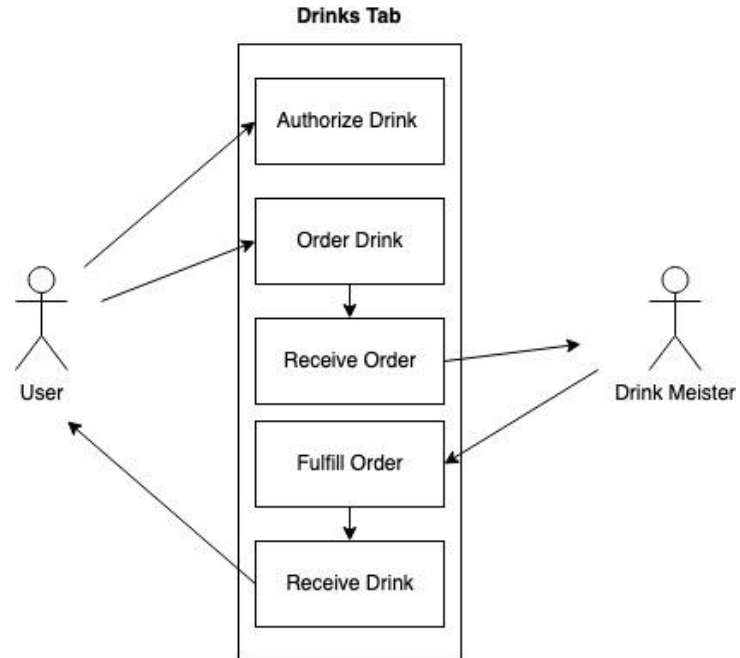
Exit Condition:

User is signed up for designated tournament.

Event Flow:

1. User signs in.
2. User selects tournament from list.
3. Backend verifies user is available for the date.
 - a. If not available, send error to user.
4. Backend adds user to the designated tournament.

Figure 5 - Drink Ordering



Participating Actors: User, Drink Meister

Entry Condition:

Customer requests a drink.

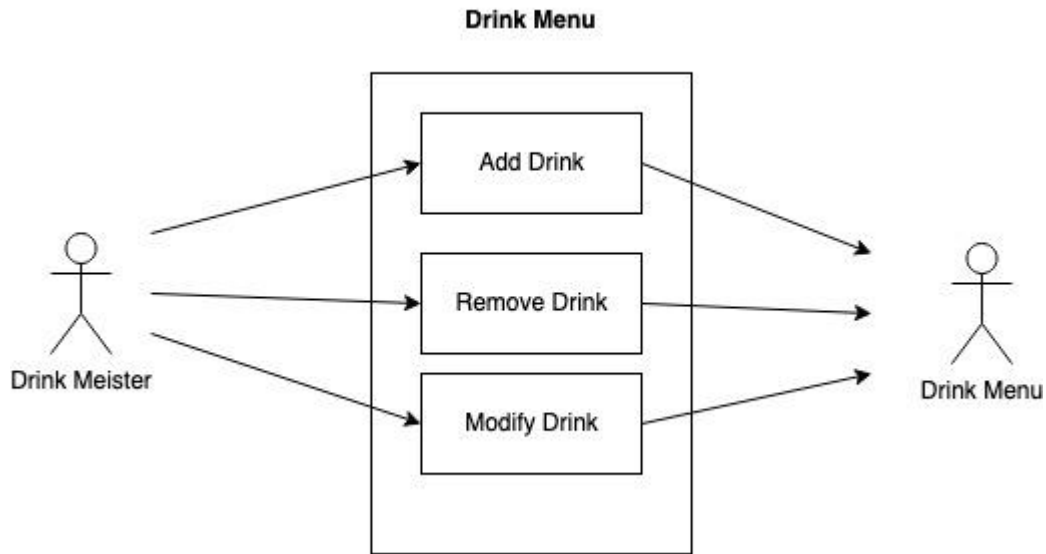
Exit Condition:

Customer receives the drink from the Meister.

Event Flow:

1. User goes to drinks tab
2. User selects drink
3. User inputs order of drink
4. Drink Meister Receives the order
5. Drink Meister dispenses the drink
6. User receives order.

Figure 6 - Drink Selection Customization



Participating Actors: Drink Meister, Drink Menu

Entry Condition:

Drink Meister goes to Custom Drink screen.

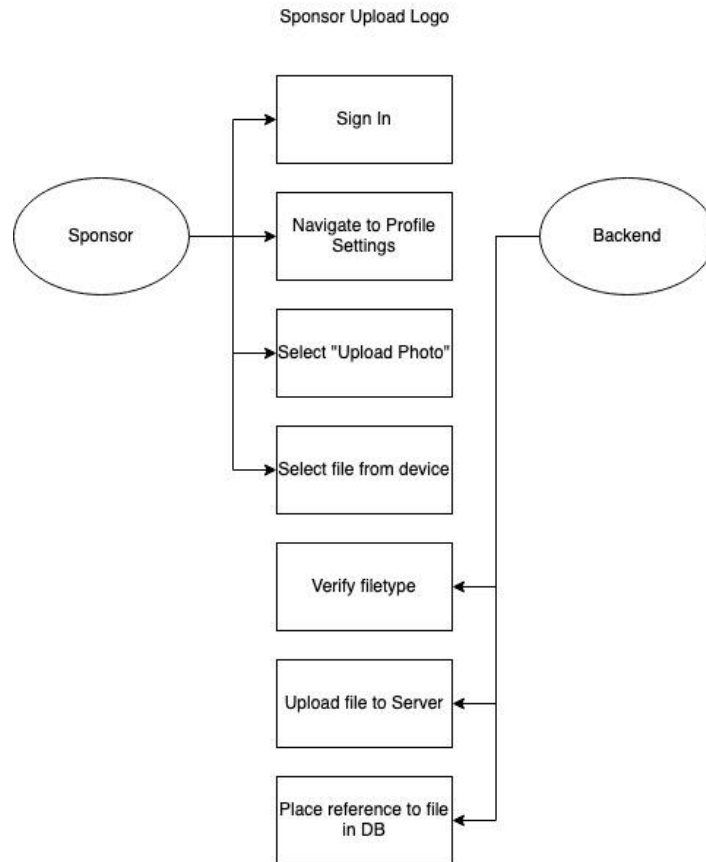
Exit Condition:

Drink Meister Adds, Removes, or Modifies drinks.

Event Flow:

1. Drink Meister Signs in
2. Drink Meister selects which command to run between Add, Remove and Modify
3. Drink Meister enters in new information.
4. The drink menu commits the alterations.

Event 7 - Sponsor Logo Upload



Participating Actors: Sponsor, Backend

Entry Condition:

Sponsor is signed up and ready to donate.

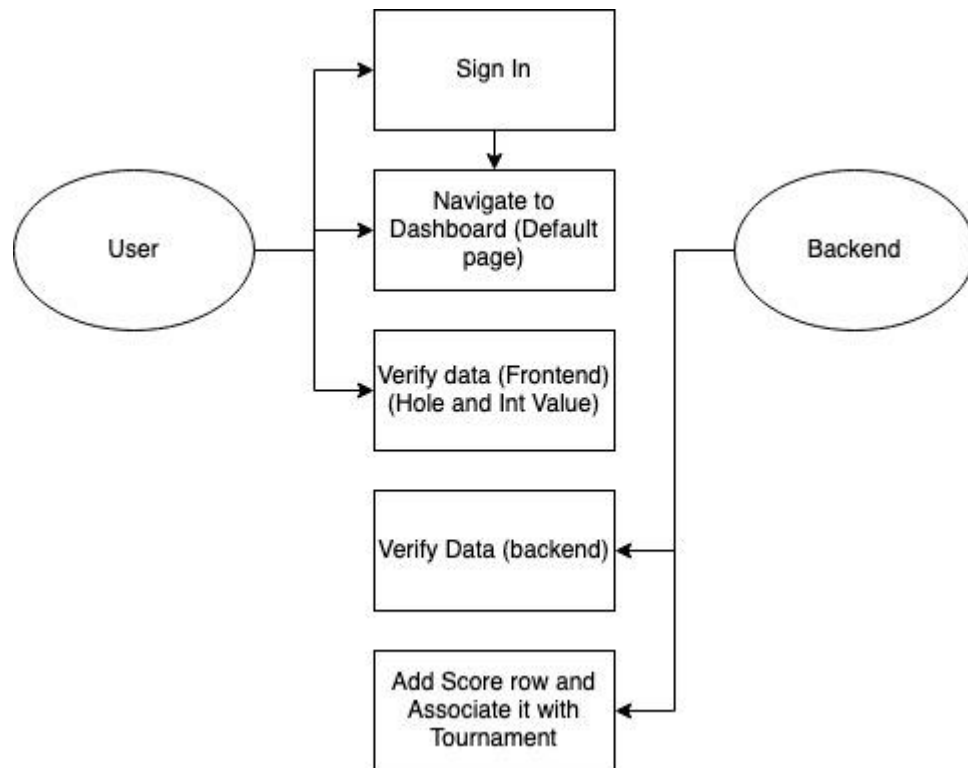
Exit Condition:

Logo is uploaded to system.

Event Flow:

1. Sponsor signs in.
2. Sponsor selects Uploads Photo from Profile Settings
3. Sponsor selects an image file on local device to upload.
4. Backend verifies valid file type.
5. Backend loads the Logo into the application.

Figure 8 - Score Update



Participating Actors: User, Backend

Entry Condition:
User is wanting to update score.

Exit Condition:
Score is incremented.

Event Flow:

1. User signs in
2. User lands on Dashboard with option to input score.
3. User inputs score.
4. Backend verifies score values
5. Backend updates the scores for User.

3.Functional Requirements

1. User Authentication and Access

- a. The system must require all users to authenticate themselves before giving them access to the system.
 - i. On first login, the system must allow the user to sign up a new account with username and password.
- b. On subsequent login, the system must allow users to enter their username and password. If entered correctly, the user must be given access to the system. If entered incorrectly, the system must allow the user to try again.
- c. Users have a combination of the following roles/permissions: Manager, Drink Meister, Sponsor.
- d. Users with Manager permissions must have access to all Manager features.
- e. Users with Drink Meister permissions must have access to all Drink Meister features.
- f. Users with Sponsor permissions must have access to all Sponsor features.

2. User Profile Features

- a. The system will allow any authenticated user to verify their own username and password.
- b. The system should not allow any user without Manager permissions from viewing or modifying any other user profile.
- c. The system should allow user to view the balance in their account
- d. The system should allow users to preload money into their account.
- e. The system should allow users to use the money from their account.

3. User Features

- a. All users will be given Default Permissions and will have access to all Default features.
- b. The system will allow users with Default rights to sign up for, track, and edit score for tournaments.
 - i. The system will allow the User to select a tournament from a list. When the user selects an event, the system will display a new section on the dashboard to enter score.
- c. Users will be able to select a drink at any time, provided by the Drink Meister.
 - i. If the user decides to order a drink, the system requires the user pays before placing the order.

- d. The User should be able to view a leaderboard of past tournaments.
 - i. Tournaments leaderboards will be populated once the given tournament reaches completion by the Manager's order.
- e. The user should be able to leave the tournament at any point.
 - i. No refunds for the drinks ordered.

4. Manager Features

- a. Managers will be given Manager rights and have access to all Manager features.
- b. Managers will have the option to start and stop tournaments at any given moment.
 - i. Managers can allocate prize money, which is contingent on the donations of sponsors.
- c. Managers can change the roles of all other users.

5. Drink Meister Features

- a. Meisters will be given Drink Meister rights and have access to all Drink Meister features.
- b. Meisters must view customer orders.
- c. Meisters can Add different drink selections for users.
- d. Meisters can Remove different drink selections for users.
- e. Meisters can Modify different drink selections for users.

6. Sponsors

- a. Sponsors will be given Sponsor rights and have access to all Sponsor features.
- b. Sponsors must upload a picture file for their logo.
- c. Sponsors allocate donations to be handles and distributed by the Manager role.
- d. Sponsors that donate will have their logos appear on user screens.

4. Non-Functional Requirements

1. The system must use a database
 - a. The system's database must store user account information, including Username, Password, and Account Balance.
 - b. The system must store information about the tournaments represented in the system.
 - i. The system must keep track of the individual users within the tournament by representing them with User ID's.
2. The team will use the Git version control system, with GitHub as a remote repository.
3. The system must be deployable
 - a. Can either be local or cloud hosted.
4. The system's interface must be mobile device friendly.

5. Future Features

This section contains a list of features that are beyond the scope of the project but could be implemented in future versions.

1. The system's Leaderboard could live update with every score updated by a user.
2. The system's authentication system can send a verification email to users that forgot their sign-in information.
3. The system could provide Users with a quick tutorial of all available functions
4. The system could email confirmations on joining a tournament.

6 Glossary

This section contains a list of important terms and their definitions.

Default User - Can also be considered a "Player" role. Initial set of permissions for every User that first logs onto our application. They will have immediate access to sign up for a tournament, order drinks, input funding (ATM), or request an Admin/Manager for additional roles.

Managers - Application admins. They approve or decline all role requests by the default user. They set up and close out the tournaments and manage/reward the money donated by Sponsors.

Sponsors - Provide the money to the Manager for the prize money. Upload an image of logo is required to run this role, to be displayed when funding is provided.

Drink Meister - Receives orders for drinks by Default User, approves drink and supplies the drink to them.

System/Application - refers to the application the project aims to build