SipNChip Tournament App

**Project Overview**

This project aims to build a system for managing putt putt tournaments, drink ordering, and sponsorship for the company SipNChip.

The system will allow customers to enroll in a putt putt tournament and keep track of their scores when in the tournament. This system will also allow sponsors to sponsor a tournament, or request a tournament on a certain day for them to sponsor. Each user will also be allowed to order drinks from a menu, which will be made and delivered to the hole the player is on.

**Team Organization**

Project Manager: Leonardo Craven (may change over the course of the project)

Designers and Developers: Gavin Murdock, Brock Sponenburgh, Andrew Ward

**Software Development Process**

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a “backlog”, and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

| **Phase** | **Iteration** |
| --- | --- |
| P1 | Requirements Capture |
| P2 | Analysis, Architectural, UI, and DB Design |
| P3 | Implementation and Unit Testing |
| P4 | More Implementation and Testing |

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

**Communication policies, procedures, and tools**

Discord – Main channel for communication. Used for group calls, file sharing, and other collaborative activities.

Google Drive – Storage for files needing collaborative effort and review for Milestone 1.

GitHub – Formal repository used for submissions, version control, data tracking, and communication with Professor Dan Watson and Rob Johnson.

Trello – Used as scrum board for organizing tasks throughout the project.

**Risk Analysis**

● Database Structure

o Likelihood – Low

o Severity – Very High

o Consequences – lack of data tracking leading to errors concerning transactions, account information and balances, tournament listings, drink listings, etc.

o Work-Around – None. System loses functionality without proper database implementation.

● Login/User Verification

o Likelihood – Low

o Severity – High

o Consequences – System will be unable to differentiate user types leading to errors concerning what users can perform which actions on the site.

o Work-Around – None. System loses functionality without being able to have verifiable user accounts to track permissible actions and account balances.

● UI

o Likelihood – Low

o Severity – Very High

o Consequences – Users will be unable to interact with the system to perform essential tasks.

o Work-Around – None. System loses functionality if users are not able to perform tasks.

● Hosting

o Likelihood – Low

o Severity – Med

o Consequences – System will be unable to host or serve information to users essential for system functionality.

o Work-Around – Host system through another hosting service.

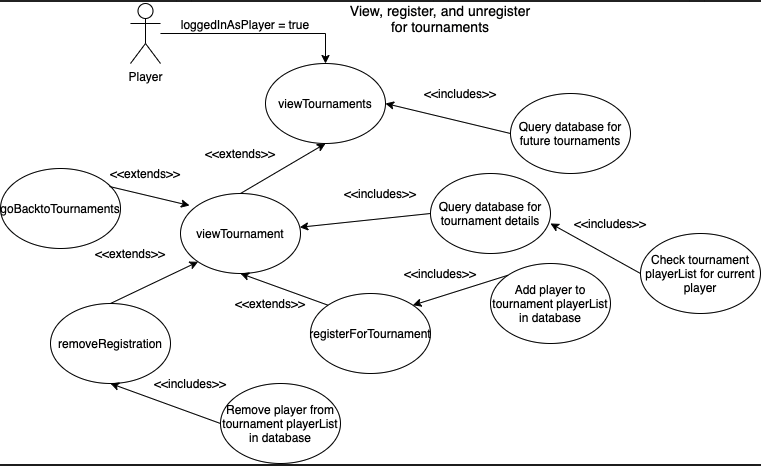
**Configuration Management**

See the README.md in the Git repository.

**Users and their Goals**

The following use case diagrams will describe the system’s actors and actors’ goals.

1 - Player views and registers/unregisters for tournaments.



Participating actor(s): Player

Entry Conditions:

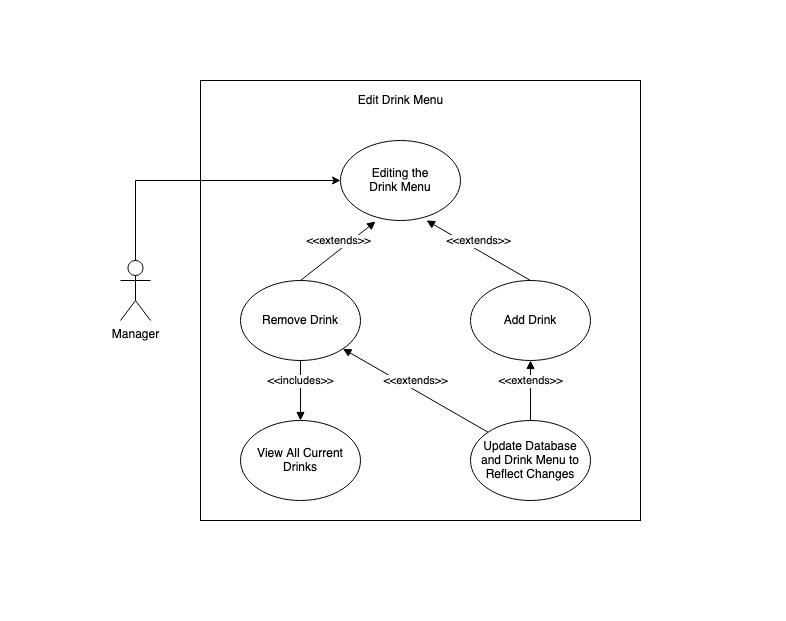
* User is logged in as Player
* Player wants to view tournaments and/or register/unregister for tournaments.

Exit Conditions:

* Player stops looking at tournaments or tournament
* Customer registers for tournament
* Customer unregisters for tournament

Event Flow:

* Player is logged in.
* Player views current tournaments.
* Player selects a specific tournament.
  + Player can return to tournaments view.
* Player register for tournament (if not already registered) **OR**
* Player unregisters for tournament.
* System returns to list of tournaments view.

2** - Manager edits the drink menu

Participating actor: Manager

Entry Conditions:

* Manager wants to change the drink menu

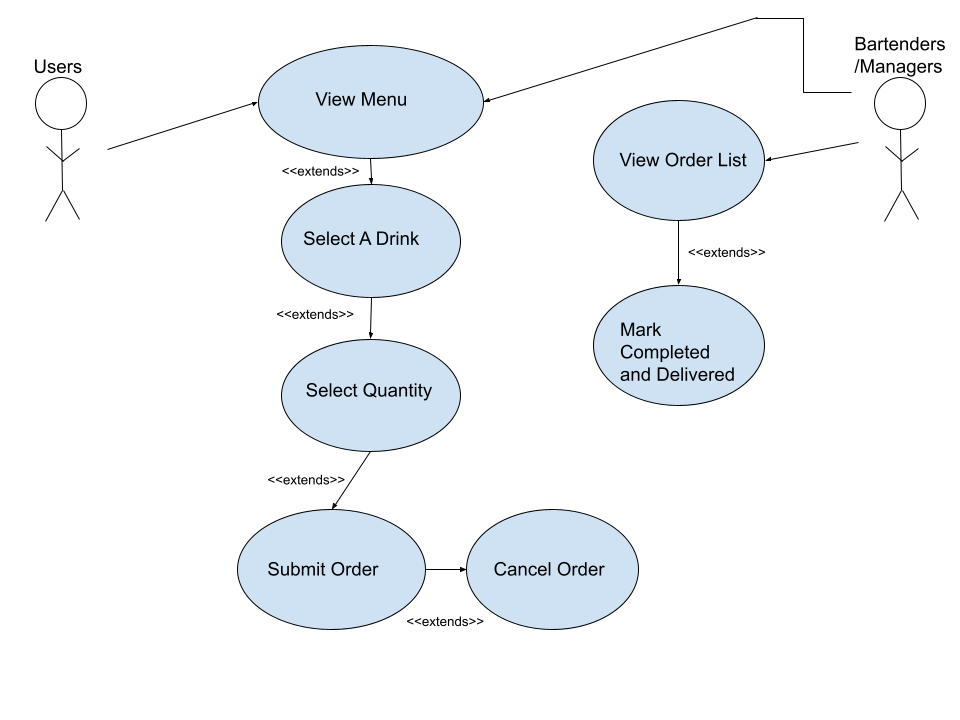
Exit Conditions:

* Manager adds or removes a drink

Event Flow:

1. Manager views page to edit drink menu
2. Manager may decide to add a drink to the menu
   1. The site requests information about the new drink
   2. Manager inputs drink information and submits request
3. Manager may decide to remove a drink from the menu
   1. Manager selects a drink to remove from the drink list and submits request
4. System updates database and the drink menu to reflect new additions/removals

3 - Any user ordering a drink



Participating Actor: Any User for ordering, Bartenders and Managers have access to the order list

Entry Conditions:

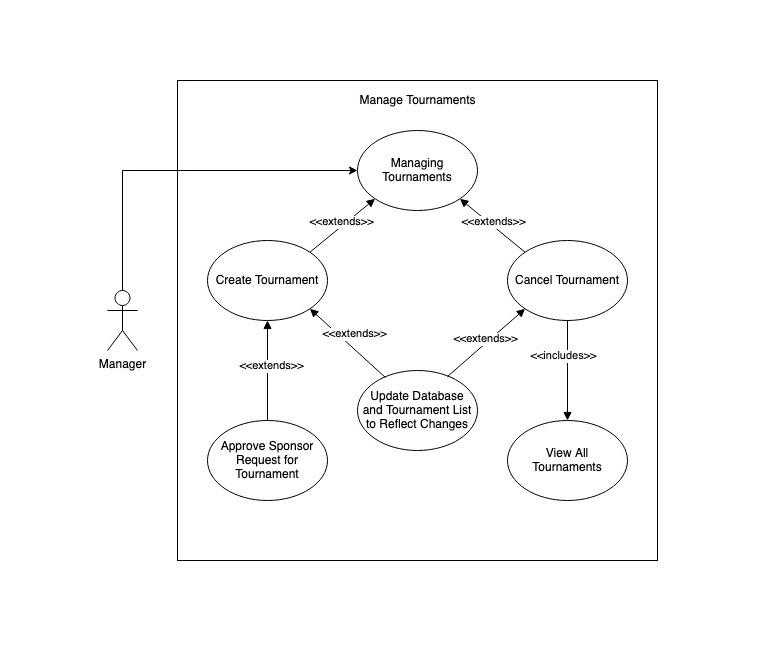
* User wants to order a drink
* Bartender or Manager wants to view the list of orders to fill them

Exit Conditions:

* Drink is ordered
* Drink has been made

Event Flow:

* User views the menu to decide on a drink
* The user selects a certain drink they wish to order.
* Set quantity of that drink
* The user submits the order.
* Before the bartender has marked the order as completed, the user can cancel the order.
* Bartenders/Managers want to view the list of orders.
* Each order has a button to mark the order as completed in the database.
* This updates the customer’s screen that their drink has been made and has been delivered.

4 - Manager edits tournaments

Participating Actor: Manager

Entry Conditions:

* Manager wants to edit the tournaments

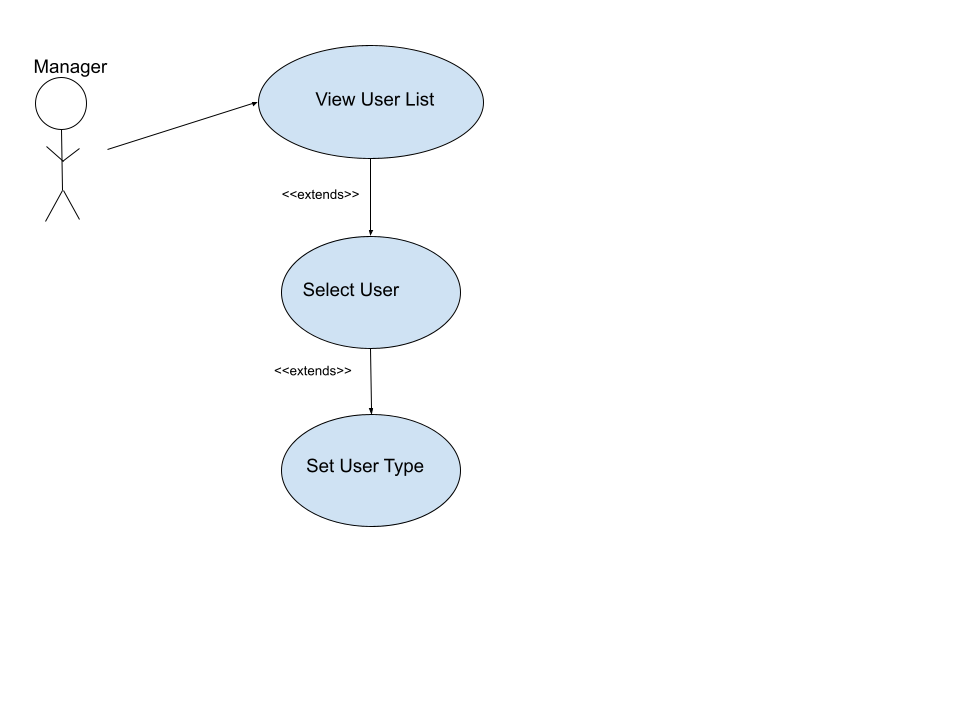
Exit Conditions:

* Manager creates or cancels or a tournament

Event Flow:

1. Manager views page to edit tournaments
2. Manager may decide to create a tournament from scratch
   1. The site requests information about the new tournament
   2. The manager inputs information about the new tournament and submits request
3. Manager may decide to create a tournament from a sponsor request
   1. Manager selects a sponsor request they want to approve and submits request
4. Manager may decide to cancel a tournament
   1. Manager selects a tournament to cancel from list of tournaments and submits request
5. System updates database and the tournament list to reflect new additions/cancellations

5 - Managers assigning user types

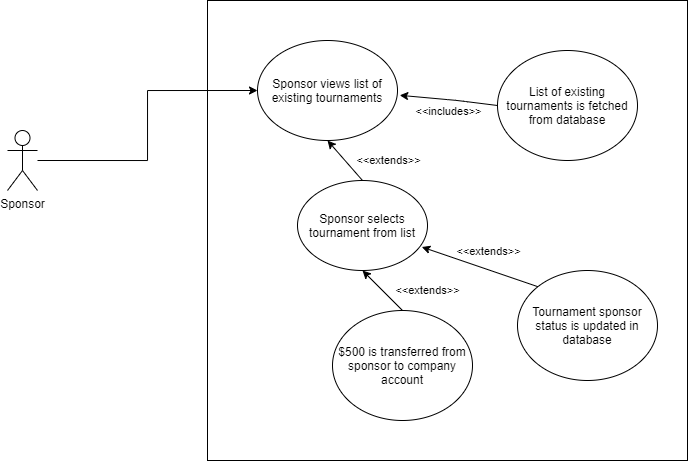


Participating Actor: Manager

Entry Conditions:

* Manager wants to set a specific user to a certain user type (Manager, Bartender, Sponsor, Player)
* All users are defaulted as Player when account is created.
* The managers can view a list of all users and their assigned type.
* The manager selects a user
* The manager sets the user type. It updates the database and gives that account the necessary permissions.

6 - Sponsor sponsors existing tournament



Participating actor: Sponsor

Entry Condition:

* The sponsor accesses the page for sponsoring existing tournaments.

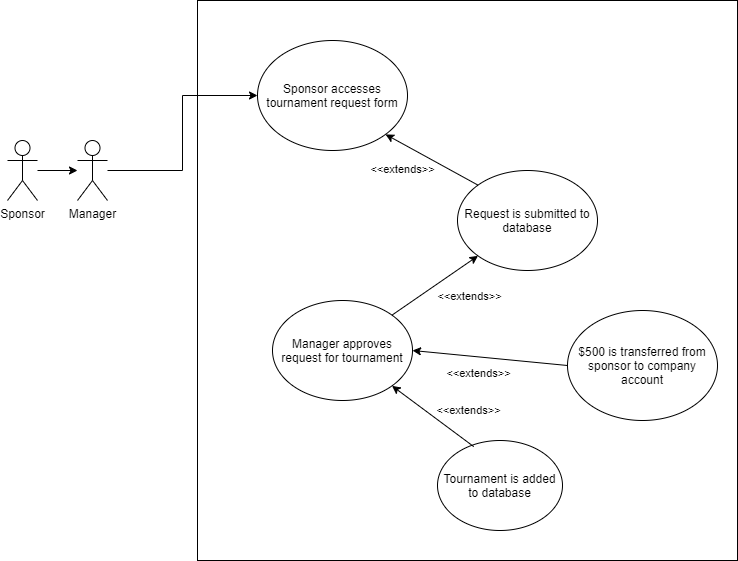
Exit Conditions:

* The sponsor successfully sponsors a tournament
* The sponsor navigates away from the page

Event Flow:

1. The sponsor navigates to the page for sponsoring existing tournaments
2. The app displays the list of existing tournaments from the database
3. The sponsor selects a tournament from this list.
4. The app prompts the sponsor to confirm sponsoring the selected tournament
5. The sponsor agrees to this prompt.
6. The app displays confirmation of the sponsorship and payment completion

7 - Sponsor requests new tournament



Participating Actors: Sponsor, Manager

Entry Condition:

* The sponsor navigates to the page for requesting a new tournament

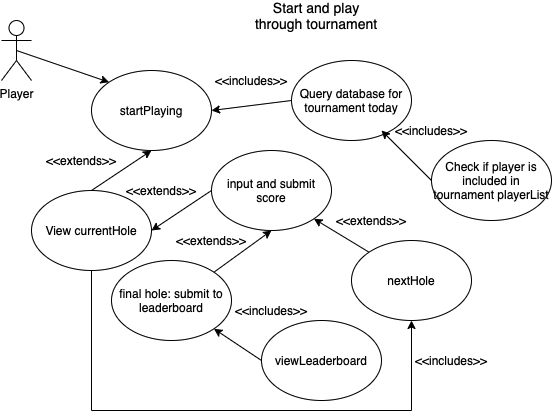
Exit Conditions:

* The manager approves a request for a new tournament
* The sponsor navigates away from the request page before submitting a request

Event Flow:

1. The sponsor accesses the page for requesting a new tournament
2. The sponsor fills out the form for a new tournament
3. The sponsor submits the request to the app database
4. The manager approves request for new tournament
5. The new tournament is added to database
6. The sponsor is notified that their request was accepted and payment was completed

8 - Player starts and plays through a game



Participating actor(s): Player

Entry Conditions:

* User is logged in as Player
* Player wants to start playing right away

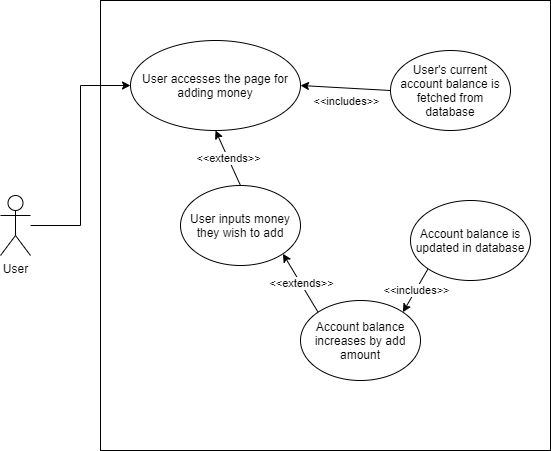
Exit Conditions:

* No tournament is today
* Player is not registered for tournament today
* Player finishes tournament/playing

Event Flow:

* Player navigates to startPlaying
  + If no tournament is today or player is not registered for tournament, then corresponding message is displayed
* Player views current hole
* Player submits score and system goes to next hole
* System reaches final hole and player’s score and other data is added to leaderboard
* Player is redirected to leaderboard

9 - User adds money to their account



Participating Actor: Any user

Entry Condition:

* The user accesses the page for adding money to their account

Exit Condition:

* The user successfully adds money to their account
* The user navigates away from the page

Event Flow:

1. The user accesses the page for adding money
2. The app displays their current account balance and prompts the user to input the amount of money they wish to add
3. The user inputs the amount of money they want to add
4. The account balance is increased by the input amount and updated in the database
5. The user is notified by the app that the money was added to their account successfully

**Glossary**

List of important terms and their definitions.

*Player -* a user that uses the system to register/play in tournaments and order drinks.

*Manager -* a user that organizes and oversees tournaments, drinks, and users.

*Sponsor -* a user that can request tournaments to be made and pay to sponsor a tournament.

*Bartender -* a user that can view, make, and fulfill drink orders/requests.

*System -* the application that the project aims to build.

*User -* any type of user role of the system (Player, Manager, Sponsor, Bartender)