

UC1: LOG IN

Preconditions: The user exists in the database. e.g. is an administrator or is a student enrolled the system tournament

Requirements: RQ1.1, RQ2.1 - RQ2.4

Basic Course

The User navigates to the base url of the website. The system displays the Public Portal. The User clicks the login link and the system displays the LogIn Page. The User enters his or her username and password and clicks the login button. The BotBattleApp verifies the submitted credentials against the Database. Once verified, the system redirects the user to the Administrator Portal if they are an Administrator or the Enter Tournament Page if they are a student user.

Alternate Course

- 1) The user entered invalid credentials: The system displays an error message and the user may then try to log in again.

UC2: INITIALIZE THE SYSTEM

Precondition: The admin has the BotBattleApp.

Requirements: RQ2.4, RQ3.4, RQ3.6, RQ4.1, RQ4.2

Basic Course:

The admin starts the system by running the BotBattleApp and passing it as an argument the port number for the server. The admin uses a browser to navigate to the Initial Configuration Page. The admin fills in the required information. The admin selects the done button. The BotBattleApp system verifies all the information is present and then sets up the file system and database. If successful the system displays the public portal.

Alternate Course:

- 1) The admin inputs incorrect information or missing information: The system highlights the problematic fields and informs the admin of the problem. The system then waits for the admin to try again.

UC3: UPDATE RULES.PDF OF A GAME MODULE

Precondition: The system has read and write access to the Game Modules directory

Requirements: RQ4.1, RQ4.3

Basic Course:

The admin navigates to the Admin Portal. Using the update rules file field the admin selects the new rules.pdf file, then clicks submit. If it was successful the system overwrites the previous rules.pdf file with the new one.

Alternate Course:

- 1) Admin uploads a file with the incorrect name: The system informs the admin of the problem.

UC4: ADD USER TO TOURNAMENT ROSTER

Precondition: Admin is logged in and at the Admin Portal. And the tournament has not yet been run.

Requirements: RQ2.2, RQ2.4, RQ3.7

Basic Course:

Admin navigates to the User Management Page of a tournament and enters the username in the inline add user fields, then clicks submit. The BotBattleApp verifies that the new username is unique, then adds the user to the database.

Alternate Course:

- 1) The username entered already exists in the tournament roster : The system displays an error inline on the User Management Page.

UC5: REMOVE USER FROM TOURNAMENT ROSTER

Precondition: Admin is logged in and is at the Admin Portal. And the tournament has not yet been run.

Requirements: RQ2.2, RQ3.7

Basic Course:

Admin navigates to the User Management Page of a tournament and selects Delete for the user they wish to delete. The BotBattleApp deletes the user from the database and deletes their bot from the tournament if they have uploaded one and reloads the page (or just refreshes the user list.)

UC6: ENTER BOT INTO TOURNAMENT

Precondition: The student has logged in and is at the Enter Tournament Page.

Requirements: RQ1.1, RQ3.1,

Basic Course:

The student selects the the path to their bot source file using the inline bot upload file field. The user then clicks upload. The BotBattleApp uploads the file, then verifies that the file has one of the supported source file extentions and compiles. If the file compiles successfully the system stores the bot in the correct private tournament directory, then add the necessary references to the database. The system updates the Enter Tournament Page to reflect that the student has successfully submitted a bot.

Alternate Course:

- 1) The student already has a bot uploaded: The uploaded bot and overwrites the old bot.
- 2) The bot does not have a supported file extension: The system notifies the student of the problem and allows them to try again.
- 3) The bot does not compile: The system displays the compilation errors to the user.

UC7: UPDATE UPLOAD DEADLINE FOR THE TOURNAMENT

Precondition: Admin is logged in and in the Admin Portal. And the tournament has not yet been run.

Requirements: RQ2.2, RQ3.4,

Basic Course:

Admin changes the upload deadline for the system's tournament and then selects submit. The system validates the date and changes the deadline.

Alternate Course:

- 1) The date entered is in the past: The system notifies the admin of the problem and allows them to try again.

UC8: TOURNAMENT PLAYBACK

Precondition: Tournament has been run and admin is logged in and has access to the tournament.

Requirements: RQ1.1, RQ3.5, RQ3.5.3, RQ6.1

Basic Course:

User navigates to the View Tournament Website. System displays the tournament bracket in its initial state (no games have been marked as won or lost). This page has three main playback advancement features, play next game, skip to next round, and skip to final results. Clicking play next game will display the game in detail (show each bot's moves in turn) then return to the bracket and mark the winner of that game. Clicking skip to next round will update the bracket with the results of each game in the corner of the bracket, if that round has been made available by the admin. Clicking skip to final results will update the bracket with the results of each game. At any point the user can click one a game in the bracket and will be shown a detailed turn by turn playback of that game. They can also press pause during the detailed playback of a game which will stop the playback. They can then press resume to continue the playback where it left off.

UC9: ADMIN RUNS THE TOURNAMENT

Precondition: The admin is logged in and is at the Admin Portal.

Requirements: RQ1.1, RQ3.1, RQ3.2, RQ3.5, RQ3.5.2

Basic Course:

The admin clicks the run button for the tournament. The system generates the tournament roster to include all students who have uploaded a bot for the tournament. The system then spawns an instance of the GameManager in tournament mode and passes it all the necessary information to run the tournament. The GameManager program executes each Game Instance of the tournament in turn and processes all of the results into a complete website stored in the private tournaments folder. Once the tournament has completed the GameManager program sends a success or failure message via stdout to the BotBattleApp, and the GameManager terminates. After receiving verification that the GameManager has terminated, the system displays the contents of the success or failure message to the admin, and if it was a success makes the view tournament, and publish next round of tournament links clickable on the Admin Portal.

Alternate Course:

- 1) Less than two users have uploaded bots for the tournament: The tournament will not be started. The system will notify the admin of the problem.

UC10: ADMIN MAKES PRIVATE TOURNAMENT ROUND PUBLIC

Precondition: The admin is logged in and on the Admin Portal and the tournament has been run.

Requirements: RQ3.5.1, RQ3.5.2

Basic Course: The admin clicks the “publish next round” button from the tournament section of the Admin Portal. The BotBattleApp copies the corresponding files for the next round from the private tournaments folder into the public tournaments folder. If the last round has been published, then the “publish next round” button is disabled. After the data has been successfully copied the BotBattleApp notifies the admin that the copy was successful and the round is now viewable from the public portal.

Alternate Course:

- 1) Copy failed for some reason: Perhaps webapp doesn't have permissions to the public and or private tournaments directory. The admin is notified of the problem.

~~UC11: ADMIN MAKES PUBLIC TOURNAMENT PRIVATE~~

~~**Precondition:** The admin is logged in and on the Admin Portal and the tournament has been run and has been previously published.~~

~~**Requirements:** RQ3.5.1, RQ3.5.2~~

~~**Basic Course:** The admin clicks the unpublish button. The system deletes the tournament from the public tournaments folder. The BotBattleApp replaces the unpublish button with the publish button.~~

~~**Alternate Course:**~~

- ~~1) Delete failed for some reason: Perhaps BotBattleApp doesn't have permissions to the public and or private tournaments directory. The admin is notified of the problem.~~

UC12: USER PLAYS AGAINST THEIR OWN BOT IN THE TEST ARENA

Precondition: The admin has at least one game module uploaded

Requirements: RQ1.1, RQ5.1, RQ5.2, RQ5.3, RQ5.4, RQ6.1, RQ6.2, RQ7.1, RQ7.2, RQ7.3

Basic Course:

The user navigates to the Test Arena. The user selects the human checkbox for player 1. The user uploads a bot for player 2. The user adjusts the move delay if desired. The user presses the start button. The BotBattleApp compiles the bot. The Game Manager creates a game instance with the bot and user as the players. The BotBattleApp displays an alternate version of the Test Arena with the ability to get user input. The Game Manager plays the game instance, getting moves for one of the players from the user.

Alternate Course:

- 1) Player sends data in the wrong format or inputs an invalid move: The BotBattleApp informs the player of the problem and waits for the player to input a new move.
- 2) The bot does not compile correctly: The BotBattleApp displays an appropriate error message. The user corrects the errors and tries again.

UC13: USER PLAYS BOT AGAINST BOT IN THE TEST ARENA

Precondition: The admin has at least one game module uploaded

Requirements: RQ1.1, RQ5.1, RQ5.2, RQ5.3, RQ5.4, RQ6.1, RQ6.2, RQ7.1, RQ7.2, RQ7.3

Basic Course:

The user navigates to the Test Arena. The user uploads a bot for player 1. The user uploads a bot for player 2. The user adjusts the move delay if desired. The user presses the start button. The system compiles the bots. The Game Manager creates a game instance with the two bots as the players. The Game Manager runs the game instance streaming the turn by turn results to the Test Arena.

Alternate Course:

- 1) A bot does not compile correctly: The system displays an appropriate error message. The user corrects the errors and tries again.

UC14: BOT PLAYS A TOURNAMENT GAME INSTANCE

Precondition: The bot exist in the system and the system is executing a game instance with the bot as one of the players.

Requirements: RQ1.1, RQ5.1, RQ5.2, RQ5.3, RQ5.4, RQ6.1, RQ6.2, RQ7.1, RQ7.2, RQ7.3

Basic Course:

The Game Manager gives the bot the current board via standard in. The bot sends a move to the Game Manager via standard out. The Game Instance checks that the move is valid. The Game Manager updates the board and saves the html for this game state. The Game Manager checks that the game is not over. The Game Manager sends the updated board to the other player and gets move. The Game Instance validates the move and the Game Manager checks that the game is still not over. The Game Manager repeats this process until the game is over.

Alternate Course:

- 1) Bot exceeds memory, CPU usage, or time limit: The Game Manager disqualifies the bot and marks the other player as the winner.
- 2) Bot does not communicate via standard I/O or sends data in the wrong format: The Game Manager disqualifies the bot and marks the other bot as the winner.
- 3) The bot returns an invalid move: The Game Manager disqualifies the bot and marks the other player as the winner.

UC15: VIEW PUBLIC TOURNAMENT

Precondition: The admin has published at least one round of the tournament.

Requirements: RQ1.1

Basic Course: A public user navigates to the Public Portal. The application displays a link to the View Tournament Website of the system's tournament in the public tournaments folder. The user clicks the link. The application loads the View Tournament Website. The user can then playback the tournament as described in UC8.

