

System Design

The system will have three components. There will be a client that a user uses to signup and enter tournaments, a server that handles requests from the client code, and a database that persists data about the users and tournaments.

View:

The first component is the client that users will have to interface with the backend. This would be the view in the model, view, controller design pattern. In the client, users will be able to view and edit account details including results of past tournaments and their bot submissions. They will also be able to see open tournaments and sign up for them using their submitted design for the game that the tournament is on. They will also be able to submit their code to be run in the tournaments, this is saved with their account details under the game that it is for. When a user first opens the client, they will be prompted to login to their account or create a new one. Their account information will persist throughout the session. From there they can add in their new submissions for a given game, or search and enter tournaments. Nothing is handled on the side of the client, it simply converts the user's clicks into messages sent back to the controller on the server which will modify their account information as needed.

Controller:

The second component is the server which handles the requests from client code. This is an HTTP server which will receive the client requests and modify the local files which make up the model, which we get into later. The server also handles the scheduling and running of the tournaments. Every so often it will check to see if a tournament has closed its sign up and then it will run the tournament. This involves running the users' submitted code in new processes so that the main server process is insulated from any issues in user submitted code. Assuming the user code is functional it will make the decisions for an instance of the game, and once a winner is determined prizes are distributed.

Model:

The model is the third component of the system. It will be where user, tournament, and submission data is stored. We will be using a relational database style, allowing us to reference users in the tournaments, and visa versa. For the user data we need to store their account information, including entered tournament ids, and submission ids. The tournament data will include references to the users who entered and which of their submissions will be fun for the tournament. Finally, the submissions will be referenced by filename. We will have a total of 3 tables, one with user ids and their information. One with tournament ids with tournament centered information, and one final table containing a map between user id, tournament id, and submission file.