
Design Document for Insatiable Insatiable Inchworms

Group 2_Muzakr_8

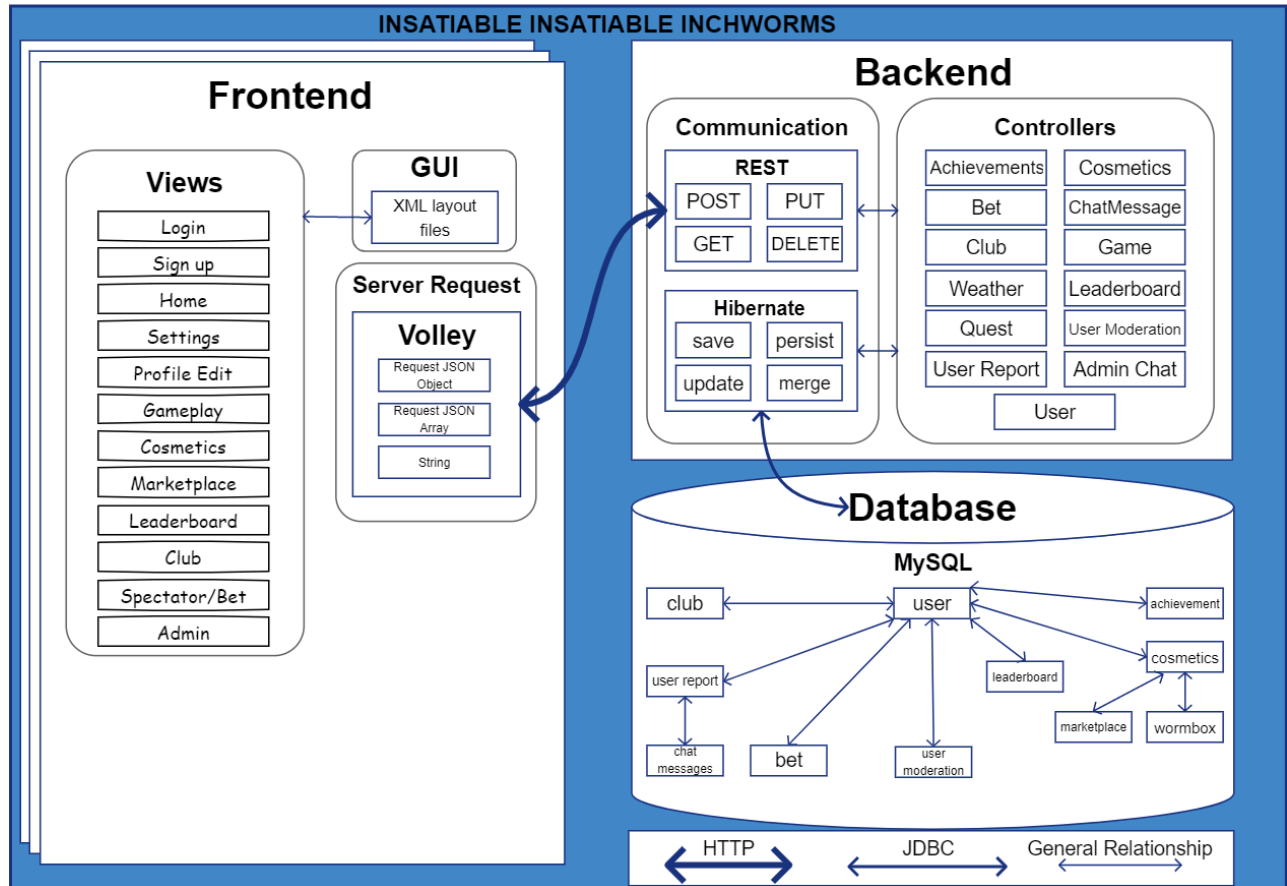
Kaden Neddermeyer: 25% contribution

Emmet Stuart: 25% contribution

Carter Hauschild: 25% contribution

Jared “Realist” “Geology” Krug: 25% contribution

PUT THE BLOCK DIAGRAM PICTURE ON THIS PAGE! (Create the picture using pencil or drawIO)



Use this third page to describe complex parts of your design.

Frontend:

- Login: Given user input of username and password, perform GET to check this information
- Sign Up: Given user input of username, email, SSN, and password, perform POST
- Pages to view user associated information: user's skins, achievements, quests, leaderboard: GET
- Marketplace provides a view to buy and sell skin based off user's worm bucks and skins: POST
- Gameplay and Spectator show real time interaction between players as received via websocket from the backend. Spectators can create bets (POST), while players click to send a request to the backend to "eat" the balls in front of the player and increment score

Backend:

Each table represented in Controllers has Rest mapping for that table and its related tables within the database.

The backend utilizes URL mapping with controllers to send, update, or delete database information:

- Post: Creates a new object in the database based on given parameters
- Delete: Deletes the select object(s) from the database
- Get: Returns the given information about the object requested
- Put: Adjusts select values of select entries in the database

Specific REST Functions

- Most Posts are 1-1 and associated with the creation of a user account - Cosmetics, Achievements, Quests, Leaderboard.
- Delete in Bets and User Report are performed after each game associated with that bet ends, or when the report is resolved.
- Puts are frequently used in Cosmetics (earn/spend wormbucks, buy/sell skins), Achievements and Quests (increase counts towards goals), Leaderboard (earned a new high score), and Clubs (a new user joined a club).

Specific Web Socket Functions

Chat messages are utilized by Global Chat and Club chat, displaying user messages to one another, and filtering by club for club chat. Admin Chat also utilizes this information to quickly search messages for offending content.

Game provides all the logic for creating a game object and performing gameplay, such as putting players in the game, starting gameplay, spawning points, and scoring based on player input, and then determining the winner before paying out spectator bets.

Weather contains basic get controllers to provide up-to-date API information that affects gameplay visuals in the frontend.

PUT THE TABLE RELATIONSHIPS DIAGRAM on this fourth page! (Create the picture using MySQLWorkbench)

