
Design Document for Quiz Bowl Quizzer

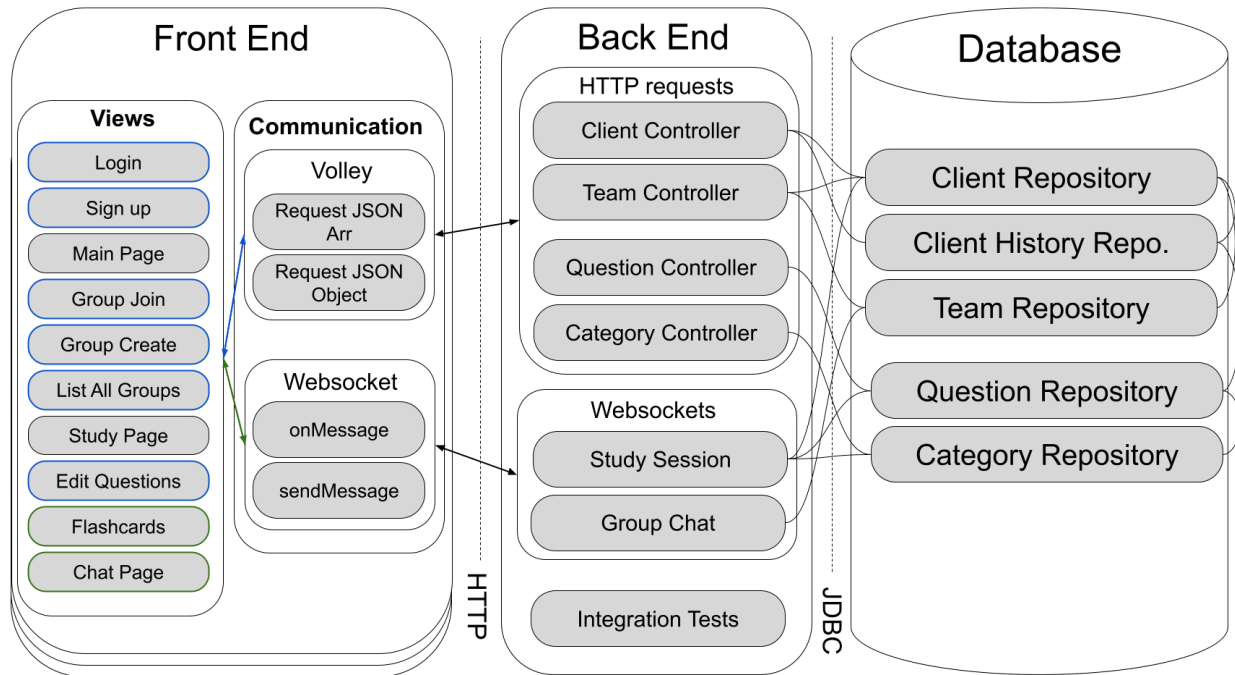
Group <MR2_7>

Oskar Niesen: 25 contribution

Sam Syrus: 25 contribution

Shreya Goel: 25 contribution

Rida Azam: 25 contribution



Back End:

- Teams:
 - Client has a many to many relationship with Team. A client can be on many teams and of course, a team can have many clients as members. One client on each team is a coach which is a unidirectional one to many relationship.
 - A team also has a group chat associated with it. This allows for members to send messages to the entire group.
- Question history persistence:
 - Client has a one to many relationship with ClientQuestionHistory to keep track of how many times a client got a question right or wrong. This can be used to serve only questions that the client hasn't seen yet or hasn't mastered.
 - ClientQuestionHistory has a unidirectional one to one relationship with Question. Question does not need to know which users has gotten it correct or not.
- Integration Tests:
 - We have a few tests to make sure our controllers are working correctly..
 - The tests are run using an in memory H2 database and a mock MVC so that we are not modifying our production data.
- Questions:
 - Questions has a one to many relationship with Category. A question can belong in only one category, but a category can have multiple questions in it.. Each question also has a point value associated with it, as well as an question and answer.
 - Questions is also associated with a live session, so that a user can have a practice or study session where they can answer questions and check if they were right or not

Front End:

- Create user
 - POST request to add user JSON object to Users table
- Login
 - GET request to check credentials with the Users table
 - If credentials are correct, then the user JSONObject is returned
 - Otherwise, nothing is returned
- Home page
 - Displays buttons for each page/activity that can be visited
- Flashcard creation
 - Allows the user to create and edit flashcards
 - These are stored and linked to the user in a table
- Groups page
 - Here it lists the members of one's team from a GET request
- Join group
 - Here it POSTs with a team id and adds a user to a team
- Create group
 - Sends a POST request to the groups table and creates one and makes the user who sent the request the coach
- Chat
 - Uses a websocket session with the server to initiate a live chat with a group
 - Allows the user to only communicate within one's group
 - Additionally, allows one to DM an individual user
- Study Session
 - Users create a websocket session with the server, which then gives them back a list of questions to answer. The user can then answer each question right or wrong and the result will be stored in ClientQuestionHistory.

Swagger Server UI Link: <http://coms-309-030.class.las.iastate.edu:8080/swagger-ui/index.html>

