System Design

CSC301

February 14, 2020

Table of Contents

CRC Cards

Class name: Enrollment
Responsibilities:
- Allow students to enroll into classrooms using the unique ids of the classroom
- Add student into classroom

Collaborators:
- Classroom

Class name: Sign-in
Responsibilities:
- Allow users to sign into their accounts
- Bring students to the webpage which displays the classes they’re enrolled in

Class name: Sign-up
Responsibilities:
- Allow users to create an account
- Add account into database

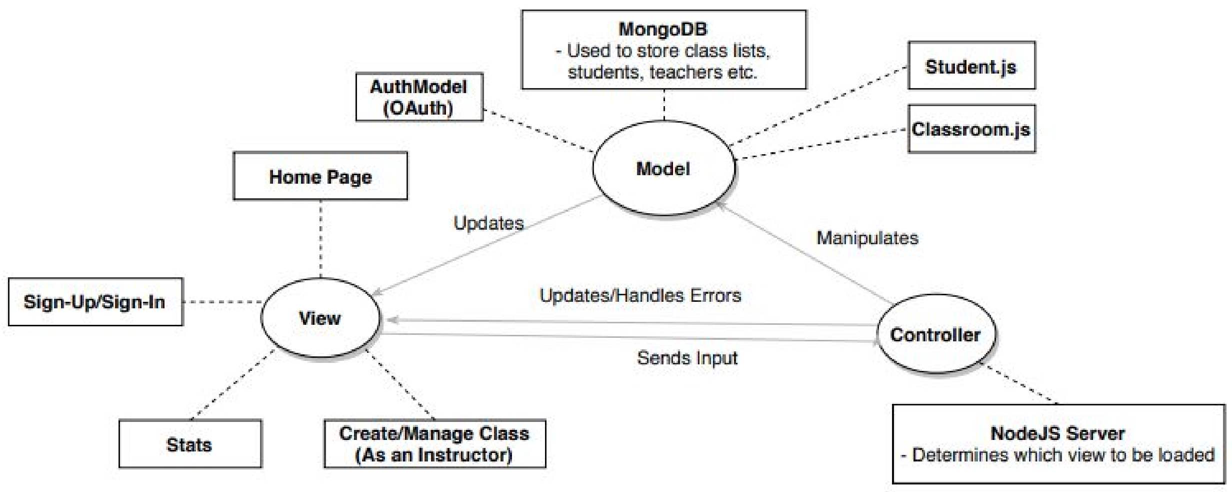
Class name: Classroom
Responsibilities:
- Adds classroom information into the database
- Knows its unique class code
- Knows what to display

Class name: App
Responsibilities:
- Create and connect to database
- Schemas for the database

System Interaction with Environment

- OS used: Windows 10, Linux
- MongoDB is the database that is used to store information

Software Architecture Diagram



System Decomposition

- Controller:
    - The server (Node.js)'s App.js will control the REST requests and what data gets served for each endpoint. This is the front controller that dictates what data is shown. Note: the node.js does not control the format of the data being shown, that is up to react-router-dom.
    - Additionally, we have react-router-dom, that controls the different views of our site, as this is a single page webapp.
    - Note: the 2 controllers both have access to the same models
- Model
    - Database and Classroom.js, Student.js, and other entities that may be added.
    - For Database, we have the classroom document, student document, grades document, quiz document, student_history document, and a document to store the additional files a teacher may add.
- View:
    - The front-end web pages make up the view of the model.
    - These would be the React components that contain both the html being displayed, and apply the materialize theme on it.

Error and Exception Handling

- If users try to sign in with an account that doesn’t exist, a message should display telling them that their username or password is incorrect. OAuth will handle authentication errors, along with the MongoDbClient in node.js.

- If users try to sign up with a username that already exists, a message should be displayed telling them to change their username. OAuth will handle authentication errors, along with the MongoDbClient in node.js.

- Some page’s urls cannot be seen without authentication first. For this, each of the subsequent “SignedInViews' will have checks to make sure that the user is in fact signed in. We will use some sort of token to keep track of this, or some react library.

- Database: should not allow for the addition of invalid data, for example, one that does not conform to our “schema”. As a design decision, we pick a loose “schema” for our MongoDb documents, so that the backend can be organized.

- In the case of a 404 error, the required data cannot be retrieved, instead of showing a data-less React view, we show a special 404 error page, and prompt the user to go back and try again.