
Block Diagram

Tarp

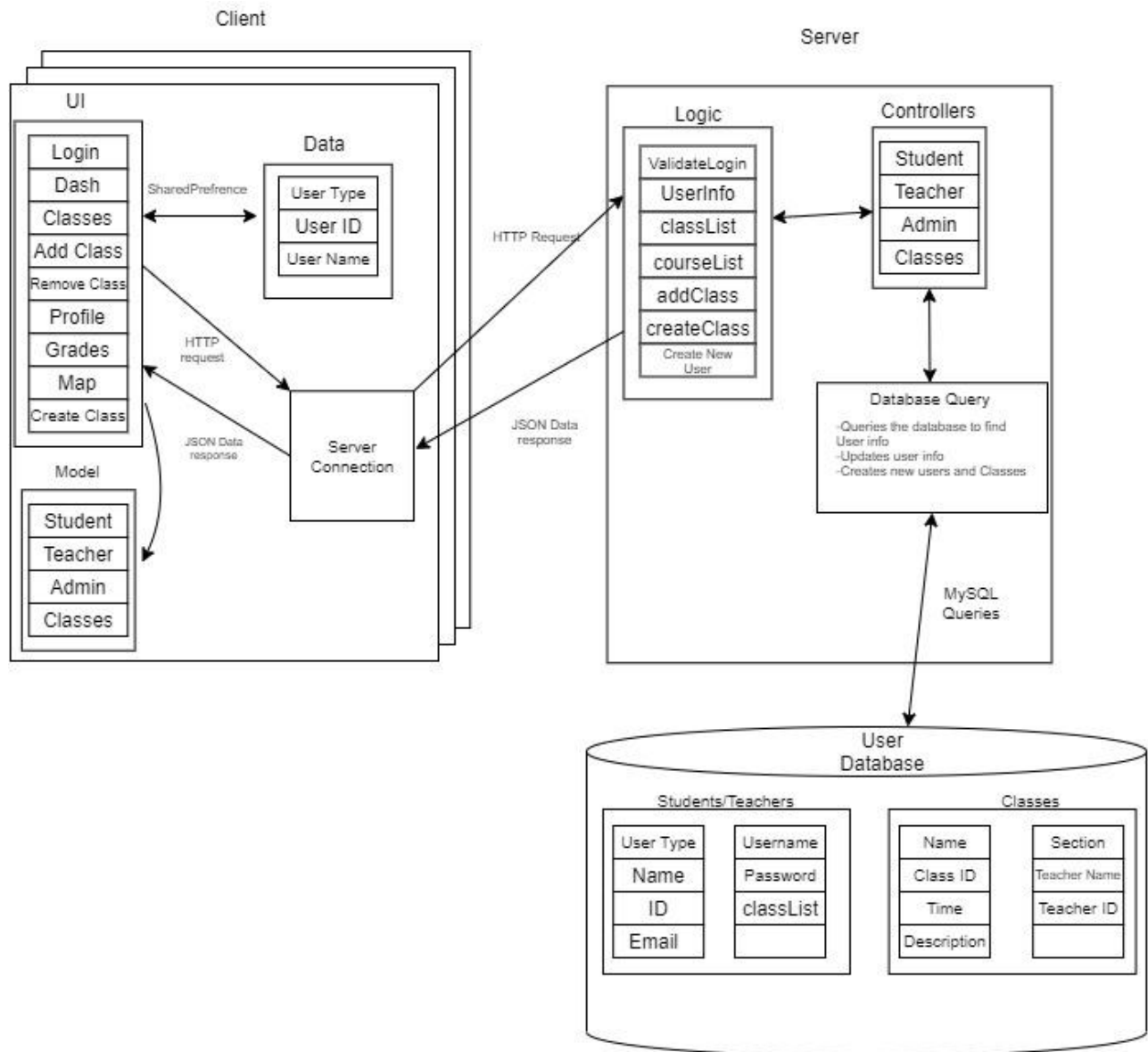
Group 2_HB_4

Austin Cichon

Brayton Rude

Greg Matson

Lincoln Khongmaly

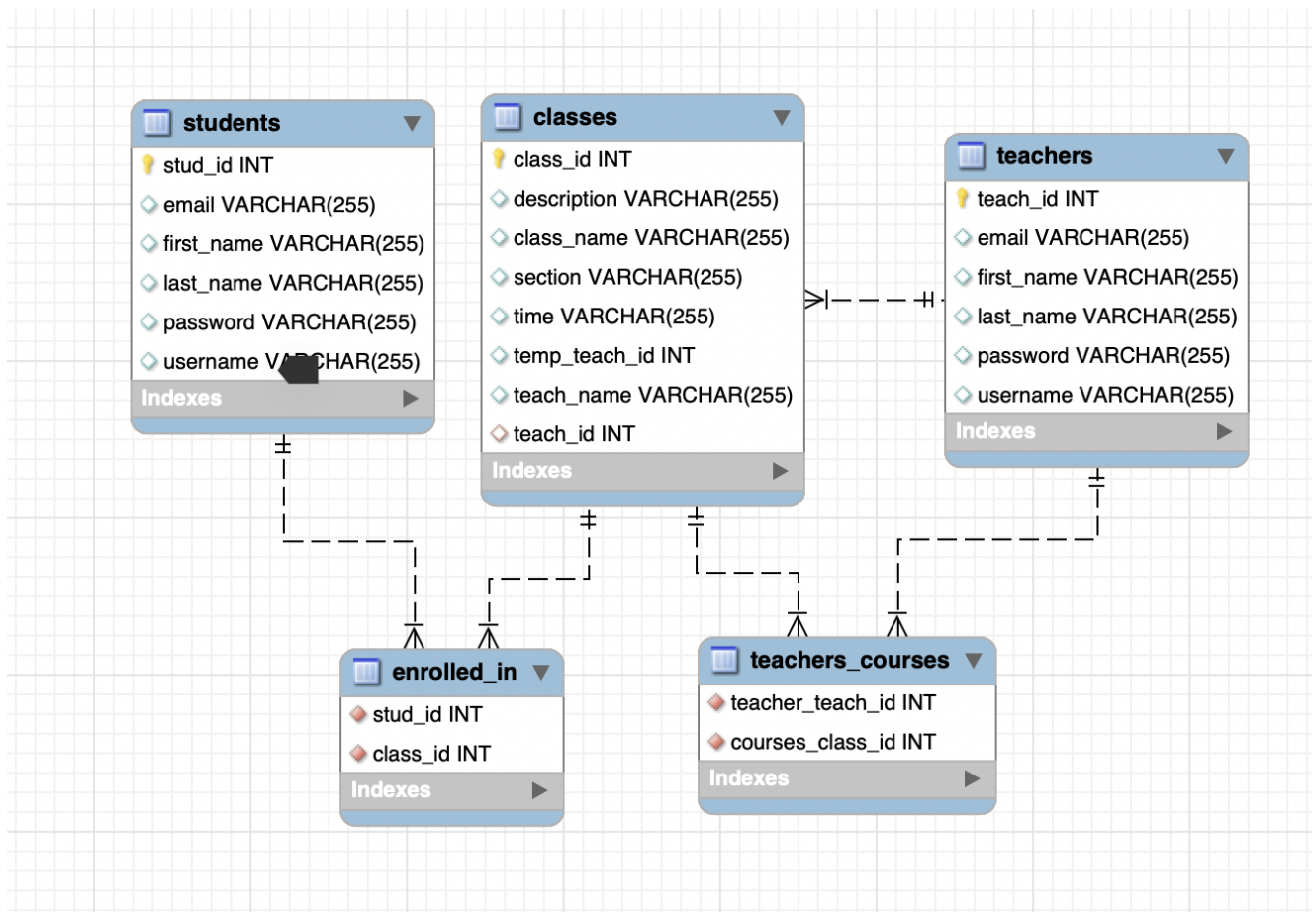


When the user boots up our app they either sign-up or login. When they login the app calls the database to verify the user and once verified, saves the users' name and ID in a SharedPreferences in the frontend. When the user signs up, they are added to the database using either the teacher or student model based on their choice, then they login.

Teacher users can create a new class. This calls the server to search the database to make sure the class name and section don't already exist. If the database does not find any class yet, it will create the class based on the model and return the class to the frontend in a JSON. This class is also added to the teachers' class list and the course list. Each class can only have one teacher, but each teacher can have multiple classes.

Student users can then search for classes the teachers have made by sending the name of the class to the database to receive a list of available classes, then they enter the section of the class they want. This will add the class to the students class list. Each student can join a class once, but each student can also join many classes. Each class can have many students.

Both users will be able to see their profile which will return the users Name, Email, and Username which they will be able to edit.



Tables and Fields (TARP Database)

students: student account information

- `stud_id` INT (Primary Key)
- `first_name` STRING
- `last_name` STRING
- `email` STRING
- `username` STRING (Unique)
- `password` STRING

classes: class object information

- `class_id` INT (Primary Key)
- `description` STRING
- `class_name` STRING

- section STRING
- time STRING
- teach_id INT (Foreign Key)
- temp_teach_id INT
- teach_name STRING

teachers: teacher account information

- teach_id INT (Primary Key)
- first_name STRING
- last_name STRING
- email STRING
- username STRING (Unique)
- password STRING

enrolled_in: many to many relationship table for students and classes that stores the related student id and class id

- stud_id INT (Foreign Key)
- class_id INT (Foreign Key)

teachers_courses: one to many relationship table for teachers and classes that stores the related teacher id and class id

- teacher_teach_id INT (Foreign Key)
- courses_class_id INT (Foreign Key)