

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

Backend Design & Implementation

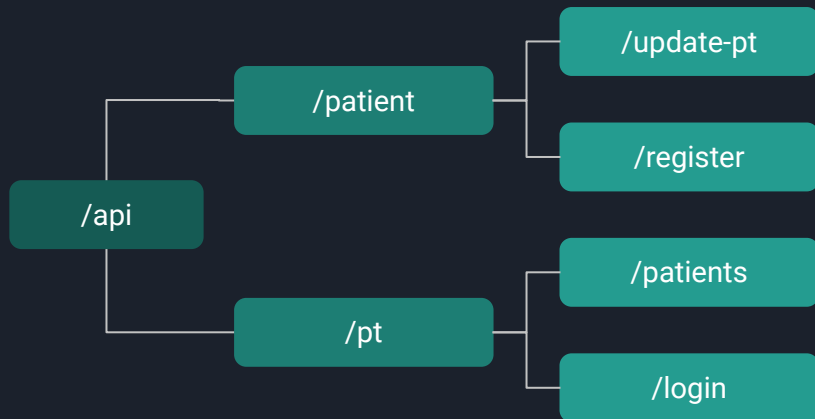
Brooke Porter -- Team 02 Backend Lead

Routing

In order for frontend to communicate with backend, we have to include paths to be called upon.

`/api/patient/register?email=test@mail.com&f_name=John&l_name=Doe&company=Test`

`/api/pt/login?email=test@mail.com&password=test`





Routing

It is important to note that each route uses specific HTTP methods to communicate.

GET /api/pt/patients.

POST /api/patient/register

PUT /api/patient/entry/comment

All of this information must be documented, as it acts as a contract between frontend and backend.



Database Communication

```
String query = "SELECT * FROM user u JOIN pt p " + "ON u.user_id = p.user WHERE u.email = ?";
```

```
Connection c = DriverManager.getConnection( Server.path, Server.username, Server.password );
```

```
PreparedStatement ps = c.prepareStatement(query);
```

```
ps.setString( 1, this.getEmail() );
```

```
ResultSet rs = ps.executeQuery();
```

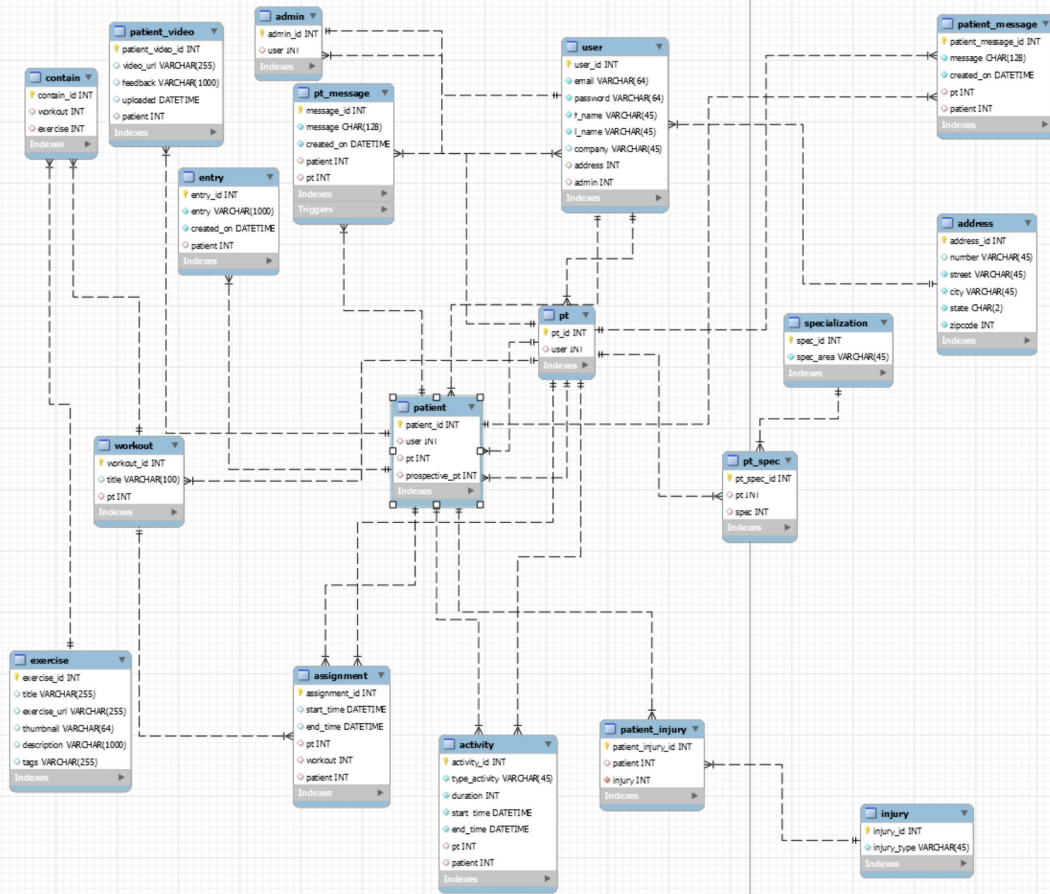


CRUD Operations

Create, Read, Update, Delete -- These methods need to be implemented for every model.

This translates to MySQL statements INSERT, SELECT, UPDATE, and DELETE, and are linked with the HTTP Methods Post, Get, Put, and Delete.

To create a PT or Patient, we first must create a User and then supply the ID from that user as a foreign key to the PT or Patient.





Challenges

Messaging -- Getting all messages between a PT and Patient meant that we lost who the sender was.

Select all PT messages from this PT to their Patient, setting “sender” as the email of the pt, and save these messages as pt_messages. Then, select all Patient messages from this Patient to the PT, setting “sender” as the email of the patient, and save these messages as patient_messages.

Finally, select all messages between the pt_messages and patient_messages lists and order by date.

When frontend gets this list of messages, the email of the sender is attached to each message.