Sprint 1

Youssef Agiza - 900192237
Nada Moursi - 900191127
Rawan Sameh - 900192388
Abdelhalim Ali - 900193539
Mohamed Ibrahim - 900170035
Mina Mikhael - 900196040

The American University in Cairo

CSCE 3701, Dr. Sherif Aly

Github repo link: https://github.com/Youssef-Agiza/Medical-Health-Record-System/

1. Planning for the first sprint.

Database:

- MySQL database is used for the project
- Create models tables using MySQL Workbench
- Perform all possible relationship between models
- Normalize the models to avoid bad structure and redundancy

Backend:

- Use Node.js to create the API
- Initiate the server base code
- Use MVC architecture for developing the API
- Create authentication controllers and routes

Frontend:

- Use flutter to develop the frontend
- Create a welcome screen
- Create a login page
- Create a Signup page
- Frontend works for web, Android and IOS

2. The sprint backlog (the prioritized set of features you will work on during this sprint).

- Users can create an account with their national ID
- Users can login with their national ID

3. A sprint retrospective (what went well, what could be improved, what you will be doing for the next sprint):

Database:

What we did:

Using the diagrams created in the first milestone, we had a good idea about the models we will have in our project. We started designing the tables for these models with all their fields and

relationships with other tables. After that, we went through multiple processes of normalizing the tables to assure good structure for the database design and that we will not be having any redundancy in the future.

What we can improve:
We will need to work more on the tables to add more table fields that will be related to the upcoming features we are planning to have.

What we will be doing in the next sprints:
Applying and creating the new tables and fields of the planned upcoming features. Normalize the structure to make sure it can scale without having issues.

Backend:

What we did:
We developed the server code for the project using Node.js. The project will have mobile and web applications, so we planned to create an API that will ease the process of connecting to the backend from the applications. The architecture we followed while designing the api was Model-View-Controller (MVC). We developed the authentication controllers where the feature of logging users in and signing them up is working. We are using JWT tokens to authenticate the coming requests from users.

What we can improve:
We are sending the JWT token in the json response and that can be a security issue. We are planning to send the token as an http-only cookie so that we reduce its vulnerability. We will need to add validation for the sent request body to assure we get only the needed information and they are valid.

What we will be doing in the next sprints:
We will develop the authorization feature to allow only authorized requests for the protected routes. For example, an admin might request to see all users. We need to make sure that this type of user has the privilege to perform such type of action.

Front End:

During the development of the front end, we created a main welcome page, signup, and login page and successfully connected all the pages with each other.

What went wrong?

Forget password, we wanted to create a page connected to the backend to verify the user and let them create a new password. We didn't have the time to do so.

What can we improve?

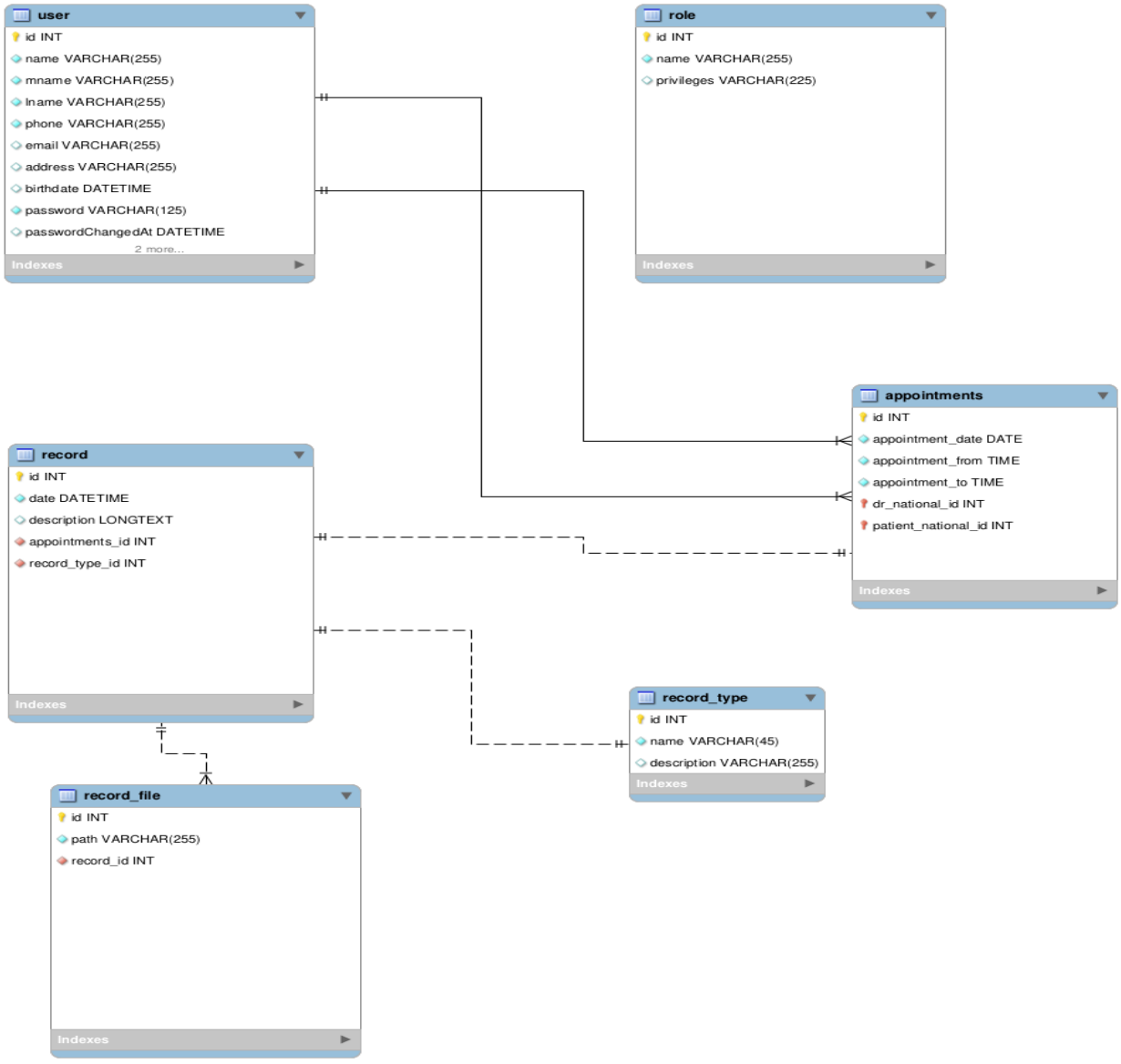
We want to create specific pages for the patient, admin, Doctor login where everyone has their own login page, because every single one of them will have their own authentication method. For example, the doctors will use his Hospital and professional ID to verify themselves, and the Admin will login using their own government authenticated credentials, and lastly the patient will login using their government issued ID. We can also create a new page for the patient for emergency use only in case of any emergencies.

What will we be doing for the next sprint?

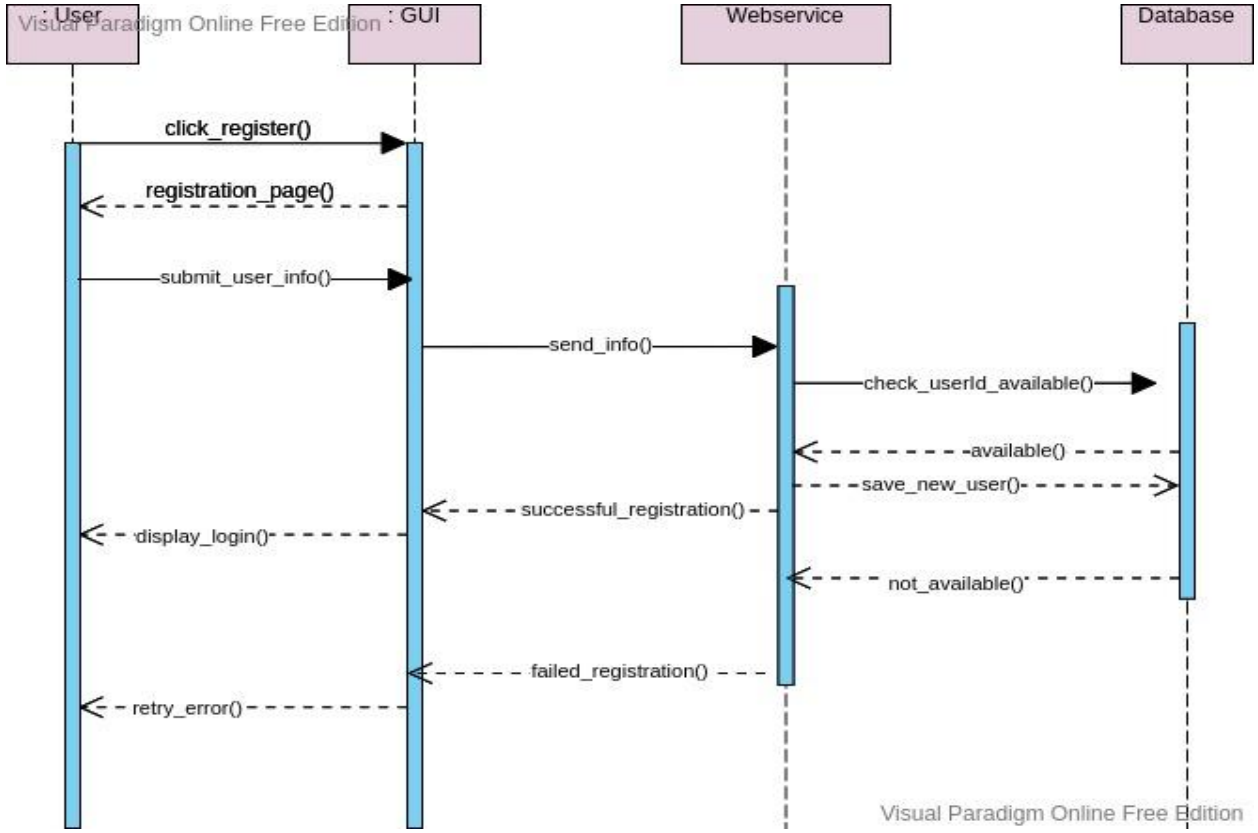
We will do the improvements that were mentioned above, and we also will create a profile page for each user including their personal information. For the patient, we will add a place so they can access all their records. For the doctors, we will give them access to patients data, and limited access to edit it and cancel appointments. For the admin, they will have full access to the system and will be able to edit, view, and comment on any data or information.

4. Refined Design:

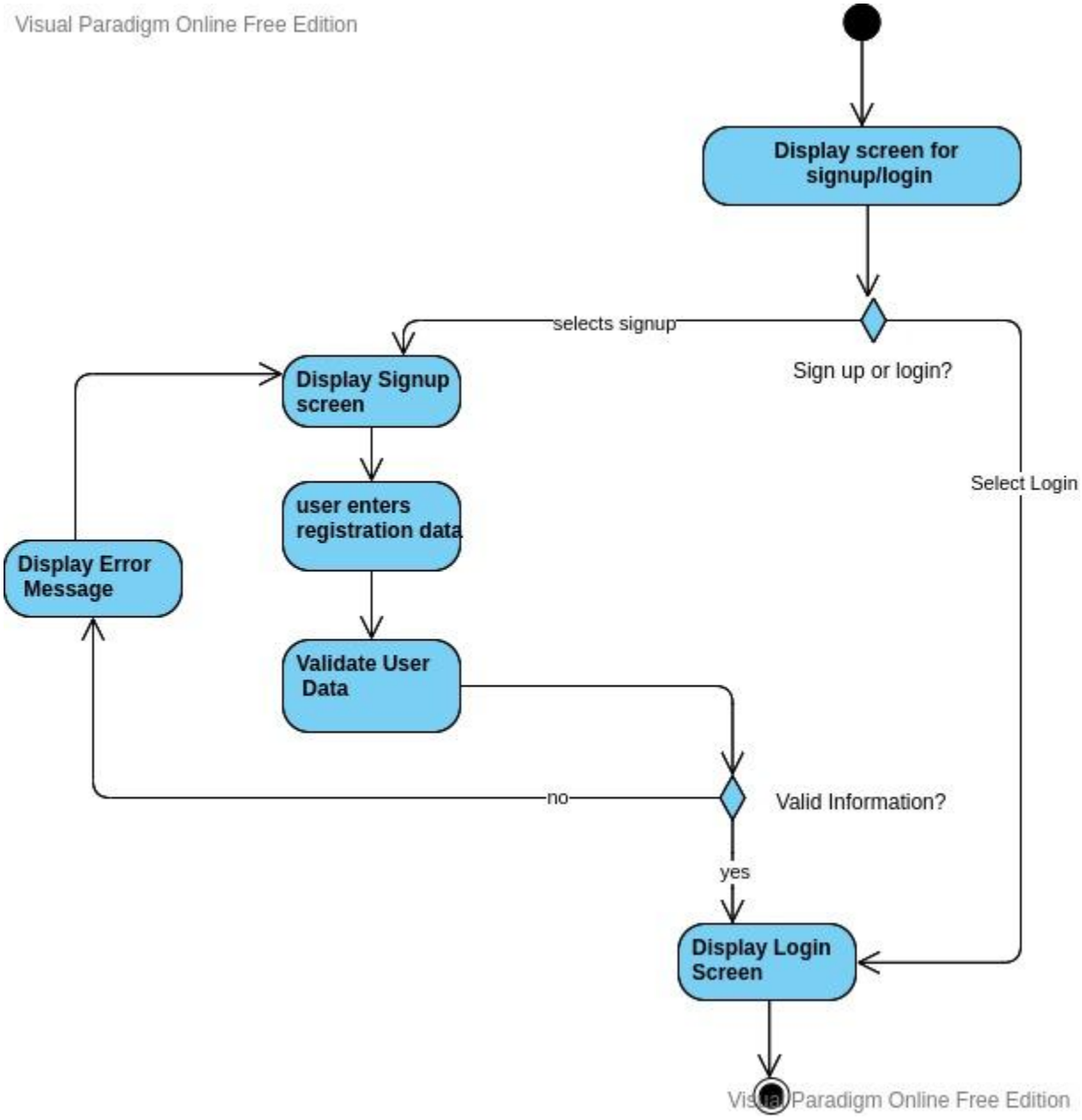
SQL Database Schema Diagram



Registration Sequence Diagram

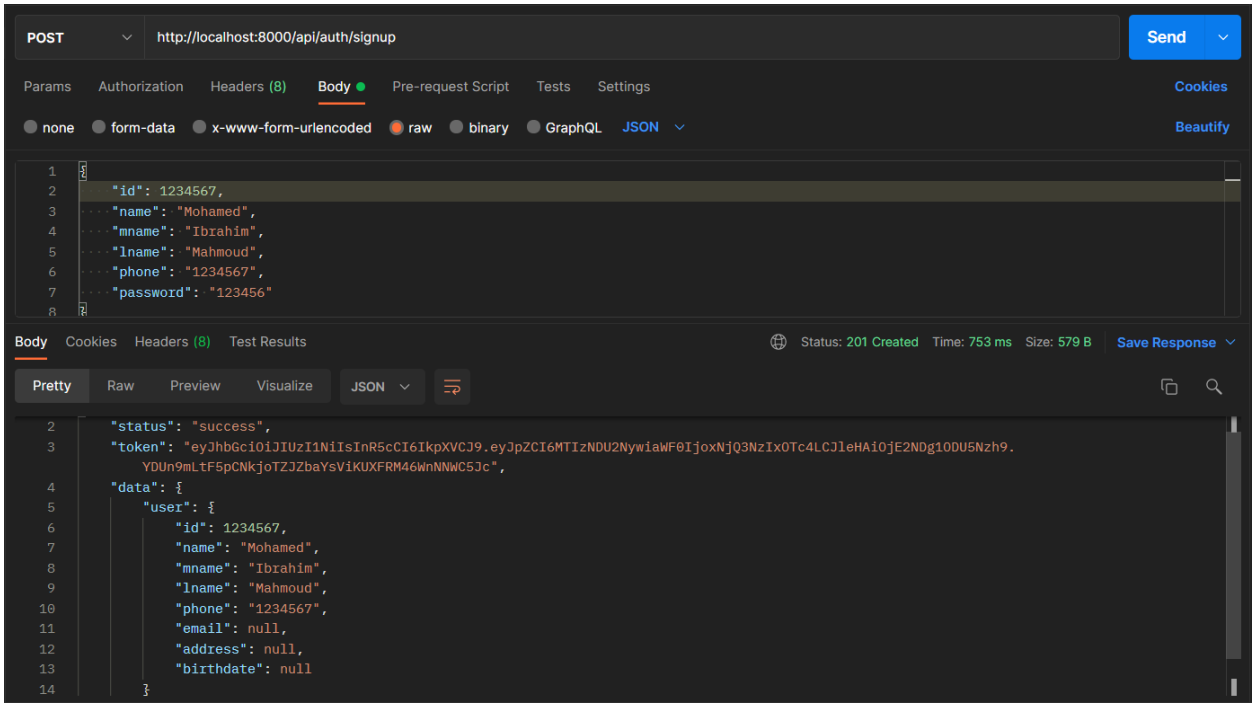


Registration Activity Diagram

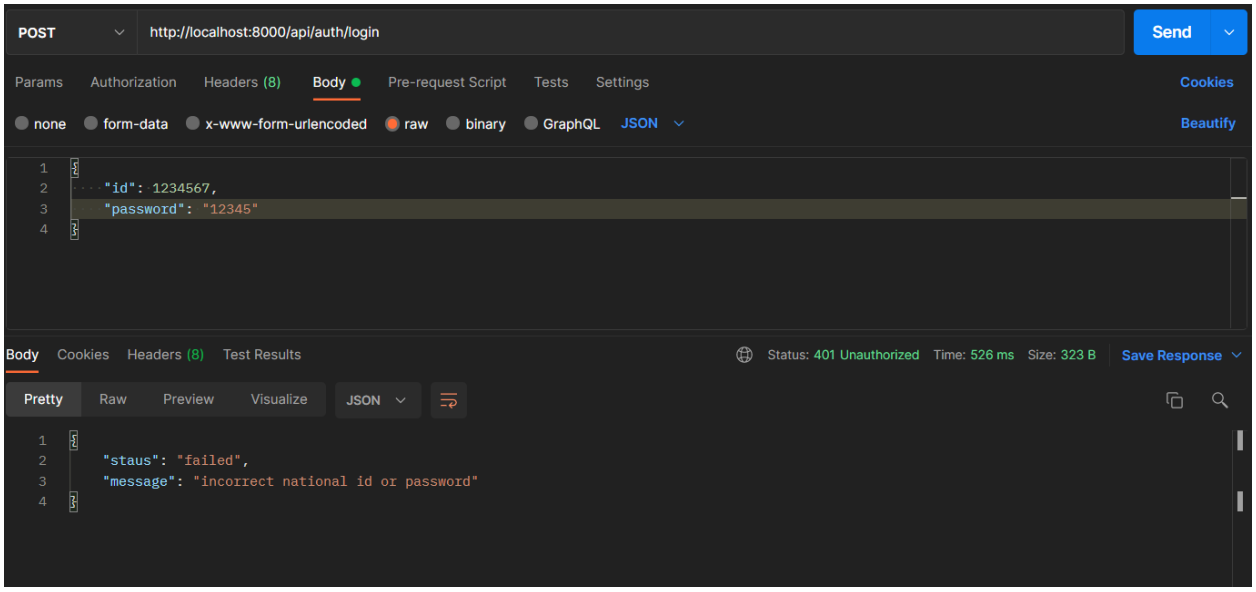


Screenshots

User signup:



User login (incorrect data):



User login (correct data):