CSC 131: Computer Software Engineering

Fall 2024 Software Design

Document

Project Overview:

The project's purpose is to develop a website for clinics. The goal of the project is to create a centralized system to manage patient records, appointment scheduling, treatment history and dental staff performance. The goal is for this website to be user friendly and easily navigable for clinical staff and patients, as well as having an easy access of communication and information for users. With the development of this project, the software would solve inaccuracies with patient care and create standardization with employee's services. Patients would benefit with a connection to staff provided through the website and possess a visual history of services. Employees would be able to track and maintain their own services, ensuring clear predictions of their pay and ease of management when providing services. Overall, the development of this software strives to establish a system that is beneficial to clinics and their patients.

Project Scope:

In Scope:

The scope of this project encompasses a web-based management system relating to information regarding patient appointments, dental treatment records, and billing.

The platform features account creation and the tracking of information associated with each account. New users will be able to input their information which will be passed through and stored in the database. Pre-existing users will be able to log into their account using their username and password. Their credentials will be encrypted and matched with the information in the database to verify user identity.

After logging in, all users will be directed to a dashboard page displaying relevant user information and serve as a hub to access other pages and features.

Different perspectives for the website and their features will be decided by the user's relationship with the website: Patient, Staff, or Admin. Staff view will allow the user to schedule appointments for patients, track upcoming appointments, and input information regarding a specific session. Similarly, Admin view will mirror Staff view with the exception of observing other staff profiles and pay distribution. Patient view will allow the user to schedule appointments, view treatment history, and pay billed amounts. All users have the ability to message other users using the integrated chat feature.

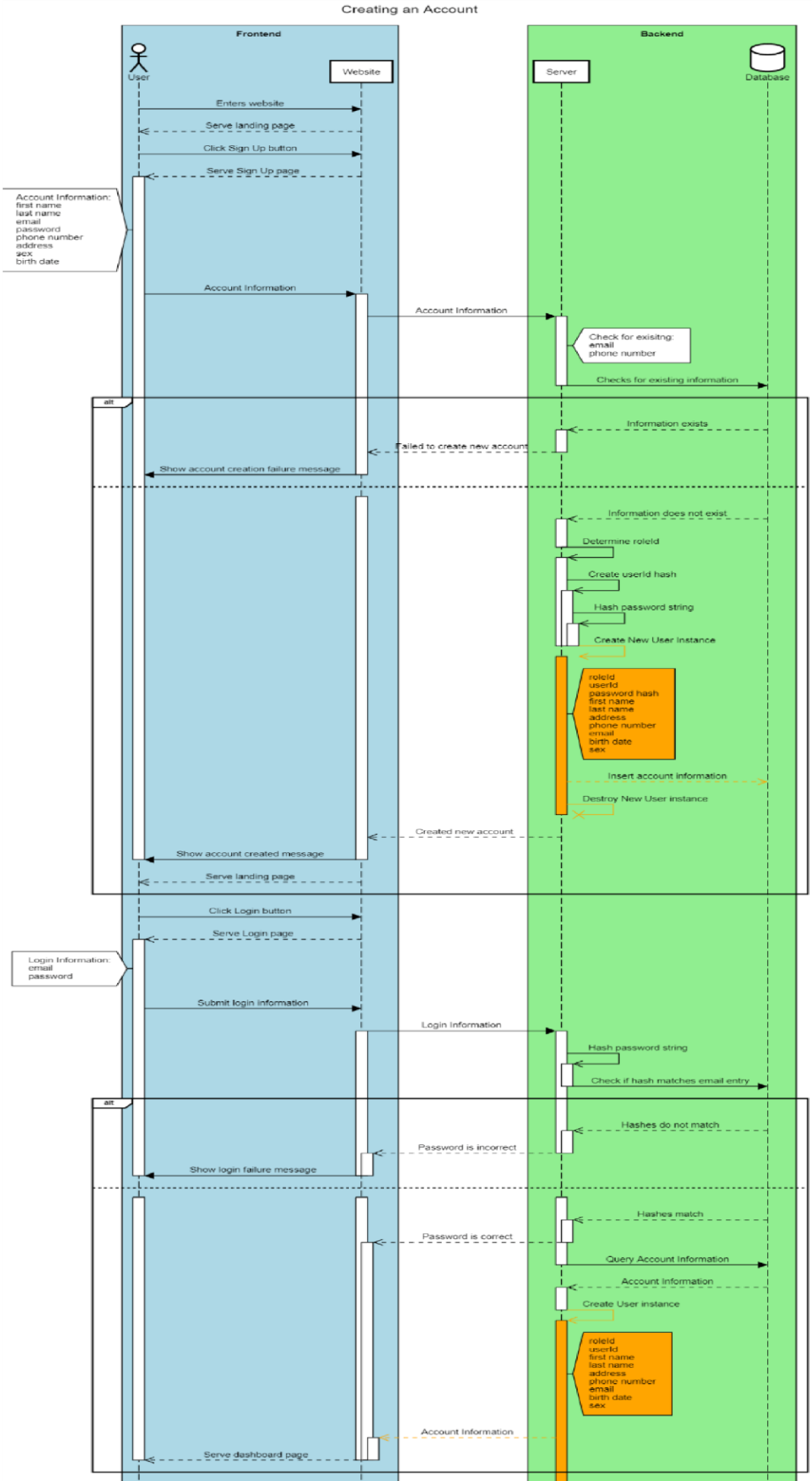
Out of Scope:

This platform is not a system that requires extensive staff training and technical knowledge to navigate. The user interface should not be complicated to use nor result in an overall negative user experience.

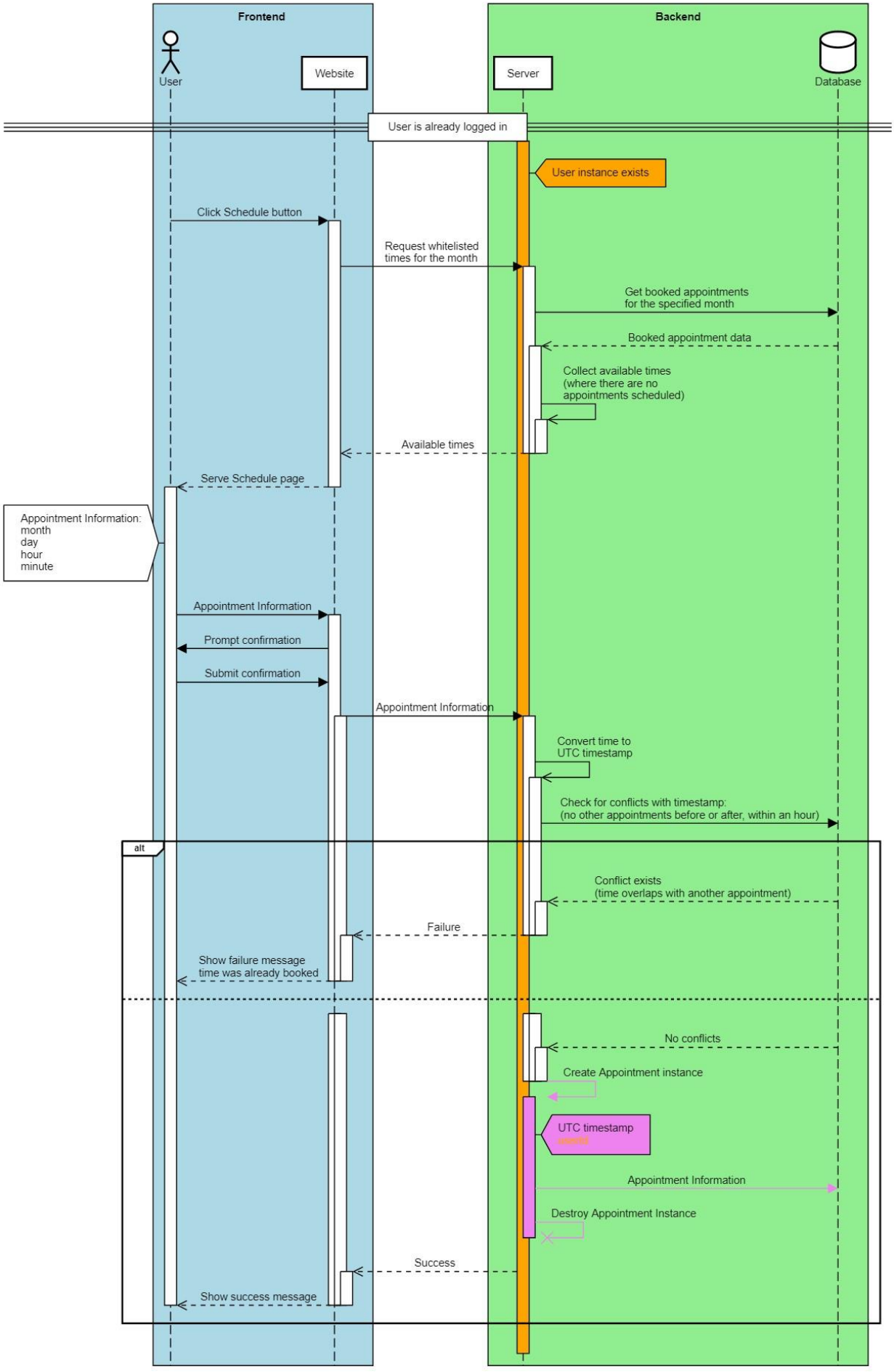
Target Audience:

The target users will be the patients, staff, and administrators of Scheduled Smiles seeking to access and record information online without visiting a physical desk. The patients of Scheduled Smiles will be able to conveniently access personal medical records, communicate with staff, and schedule appointments online without needing to visit the site directly. Staff will be able to manage their shifts, interact with patients, and view and record both future and past appointments through the website. Meanwhile, administrators will be able to view staff timesheets and pay online without direct interaction with staff.

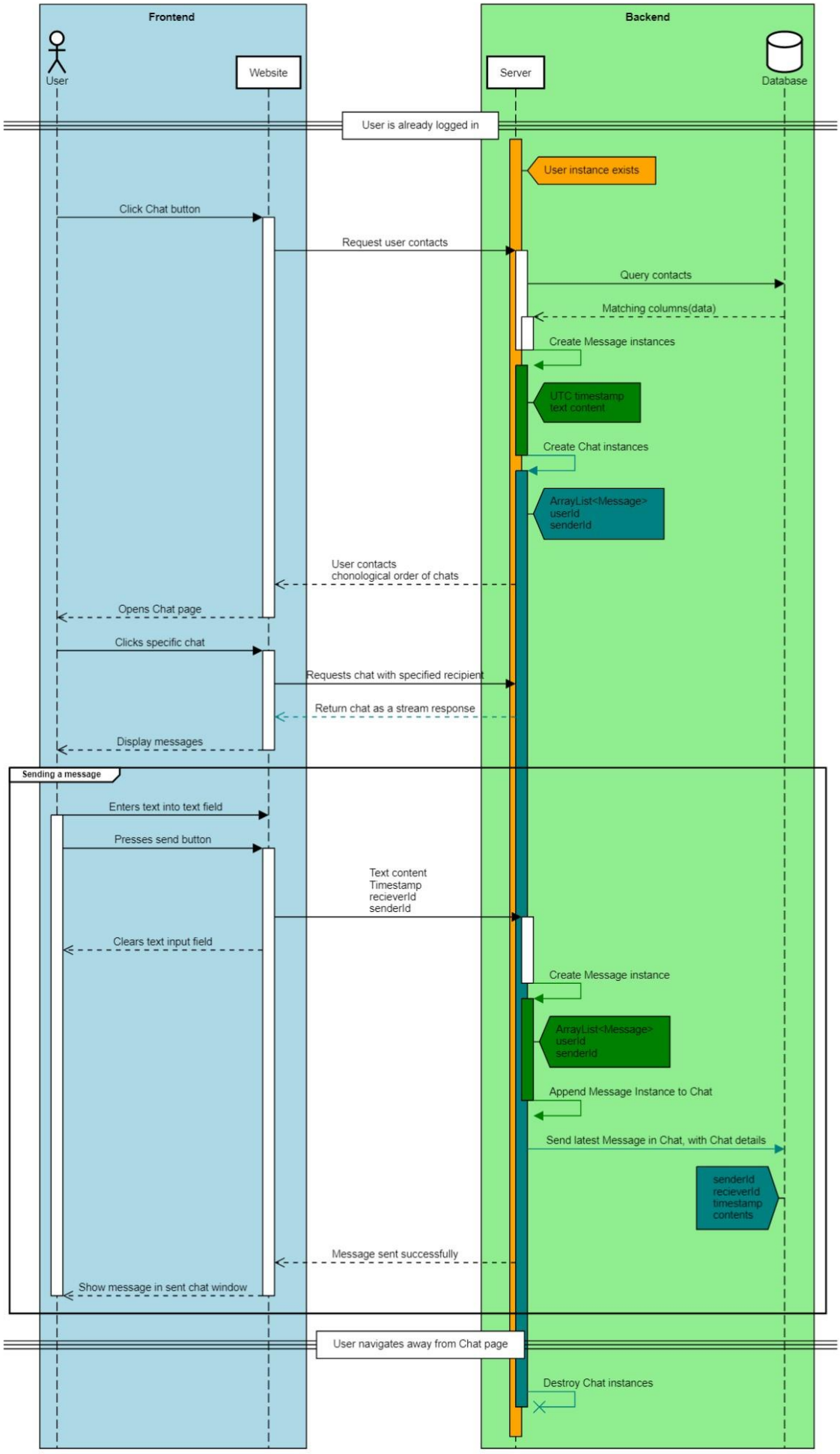
Sequence Diagrams:


Creating an Account
Frontend
Backend
User
Website
Server
Database
Enters website
Serve landing page
Click Sign Up button
Serve Sign Up page
Account Information:
first name
last name
email
password
phone number
address
sex
birth date
Account Information
Account Information
Check for exisitng:
email
phone number
Checks for existing information
alt
Information exists
Failed to create new account
Show account creation failure message
Information does not exist
Determine roleId
Create userId hash
Hash password string
Create New User Instance
roleId
userId
password hash
first name
last name
address
phone number
email
birth date
sex
Insert account information
Destroy New User instance
Created new account
Show account created message
Serve landing page
Click Login button
Serve Login page
Login Information:
email
password
Submit login information
Login Information
Hash password string
Check if hash matches email entry
alt
Hashes do not match
Password is incorrect
Show login failure message
Hashes match
Password is correct
Query Account Information
Account Information
Create User instance
roleId
userId
first name
last name
address
phone number
email
birth date
sex
Account Information
Serve dashboard page

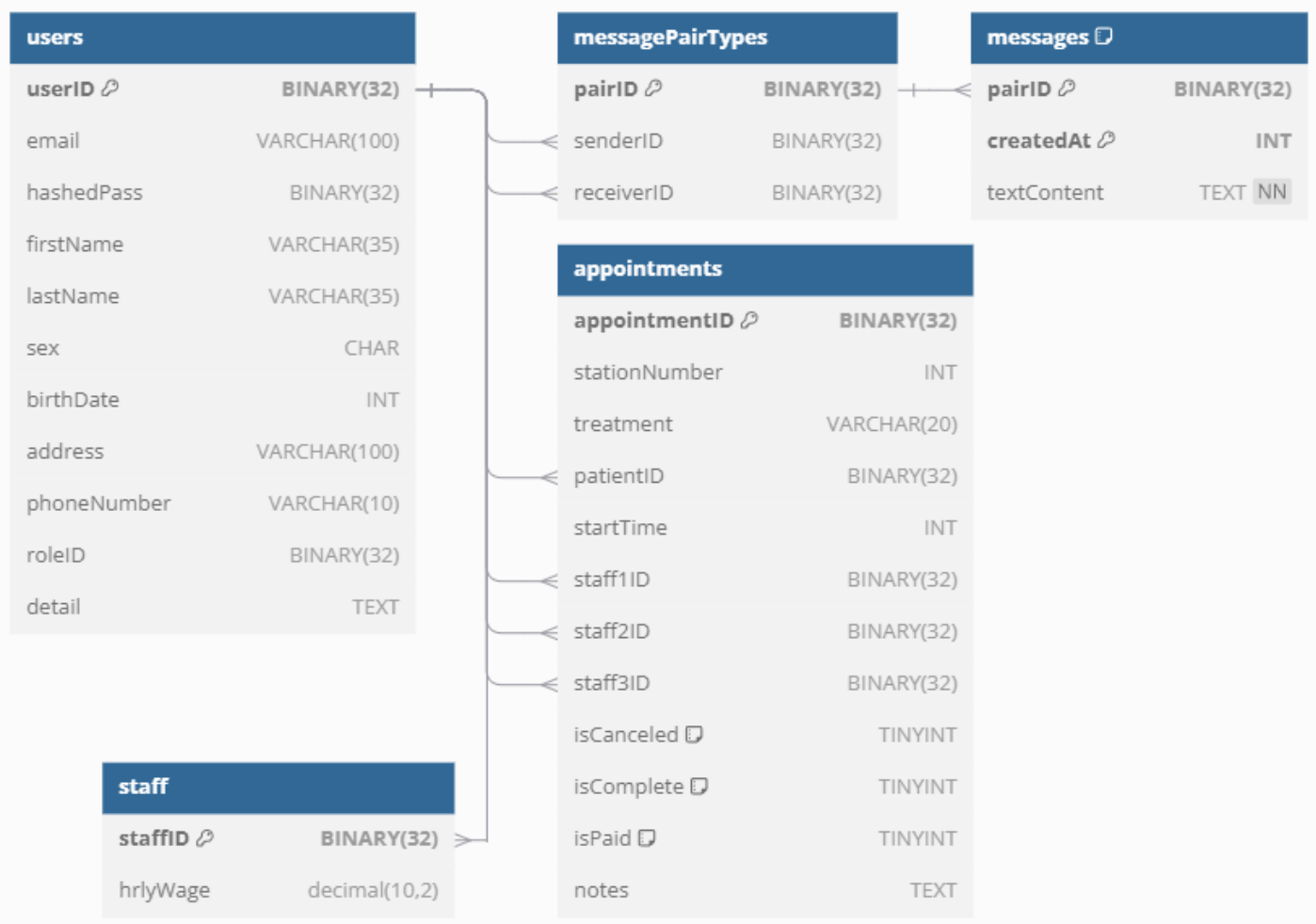
Scheduling an Appointment



Chat feature

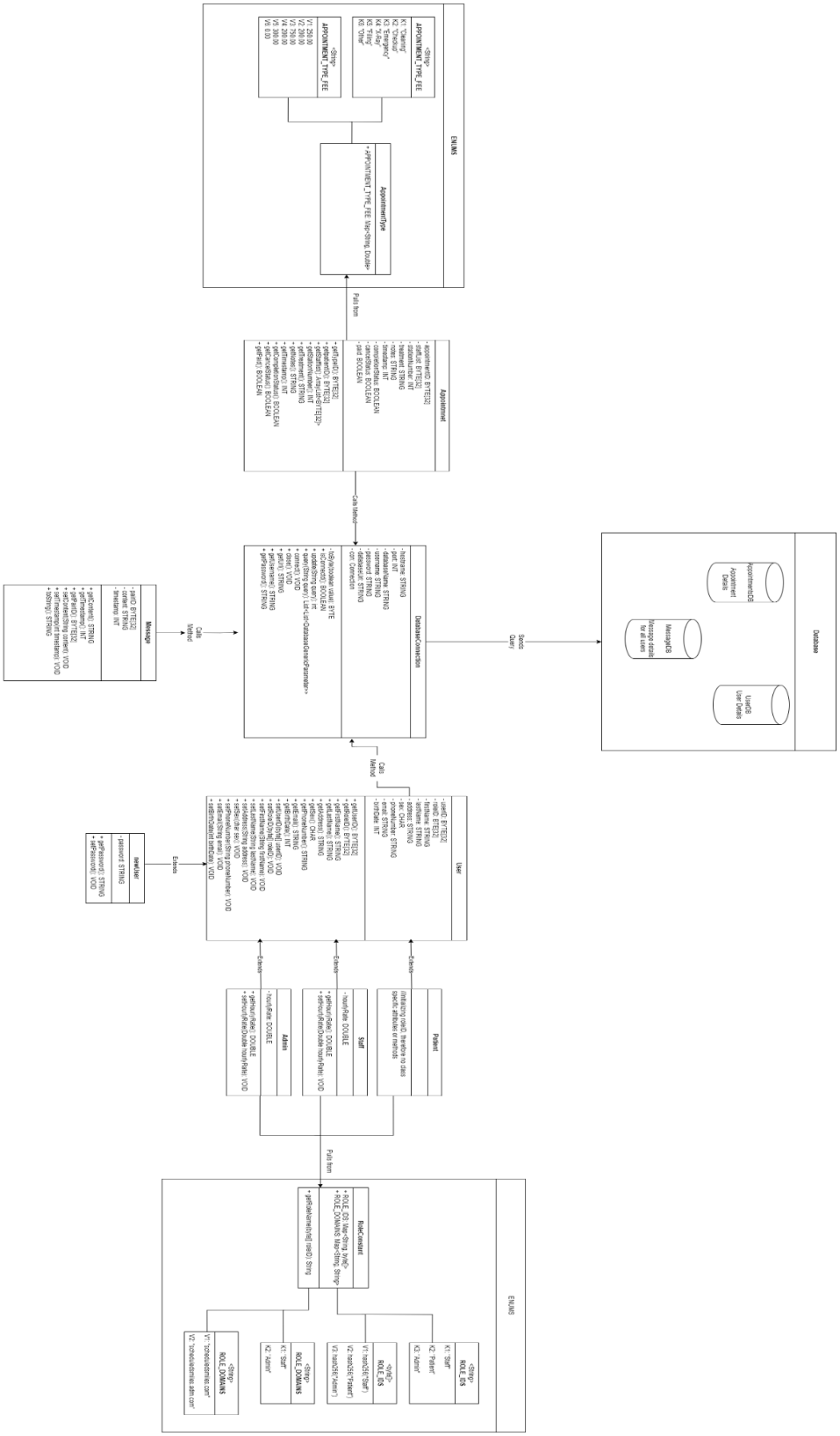


Database Schema:



UML Class Diagrams:

UML Diagram.drawio.png



Technologies and Tools:

- **Programming Languages used**:
- Javascript : For client-side interaction to the web page
- HTML : Standard markup language for web pages
- CSS : Used to style HTML how the elements will be displayed
- Java : Used for Object Oriented design and structure of our software.

- **Libraries used**:
- JDBC (Java Database Connectivity) : Used for connectivity between the server-side and MySQL database.
- JDK.HTTP (Java Development Kit.HTTP): Used for connectivity between the client-side and server-side.

- **Additional Tools used**:
- MySQL : This will be our database for servers to store information.
- AWS: Remote platform to host MySQL database.
- Github : Used for hosting compilation of changes and to log the main development of programming changes.

Development Environment:

Version Control:

- **Git:** We will be using Git as the version control system to track changes to the codebase, enabling collaboration amongst the development team. This will ensure that code history is maintained, validated, and any changes can be reverted or merged as needed.

- **GitHub:** GitHub, a cloud-based platform, will be the hosting location for our Git repository. The utilization of a remote repository will allow individual members of the team to clone the repository for work wherever and whenever they need. GitHub also provides many collaborative features such as pull requests, code reviews, and issues tracking which are necessary for the quality assurance of the project's development.

Collaboration Tools:

- **Jira:** Jira will be our project management tool, allowing the team to keep track of current and upcoming tasks, bugs, and overall project progress. Team members are notified of any changes to the status of these tasks via email and are able to see in-depth information regarding tasks assigned to them as well as understanding project's progress towards completion. The Scrum Master will be able to create, update, and assign existing tasks to developers most-fitted based on the specific requirements of the task. The Scrum Master will also be the main point of contact between the different development teams to ensure compatibility, clarity, and obstacle clearance for a seamless workflow towards project completion.

- **Discord:** The primary form of communication between the team will be through Discord. Messages, announcements, project content, calls, and meetings will take place on the social networking platform. Team members can provide their current progress status and request for assistance from other members as well as directly connect to other role-oriented teams for clarity and assistance on specialized issues.

Testing Framework:

- **Java SDK:** Testing will be done locally on developer run systems. Site server will be hosted and accessed locally to test, verify, validate, and eliminate issues that arise. To ensure that outputs are identical to one another, developers will deliver implementations using the same software and library versions.

Design Prototype:

1. Figma design prototype:

https://www.figma.com/proto/q30rwMn9oEnSBc3fst7kFy/Scheduled-Smiles-Site?node-id=0-1&t=B50lojxWnFYkh2JB-1

2. Design Patterns:

The design of our prototype revolves around maintainability and usability. We achieve this by focusing on a standard language that is feasible and usable for the team. The direction we moved towards was minimizing the cost value towards actual development,

while pushing for clear and simplistic functionality. Given the scale of the project, commercial use for clinics, we are able to focus on small scale operations of our developing software. The functionality of our design does not have to accommodate large interactions between applications and servers of scope. Thus, our focus was for interactions between a smaller user base and our software. The design of our website uses structured and interactive buttons for friendly and clear instructions for end users and allows our team to comprehensively adjust for changes.

Risks and Mitigations:

Inevitably, there will be potential risks our team may face that could delay the project's progress. One such issue is the lack of a testing environment or inability to host a server for the back-end team. In order to minimize this, our team plans to establish a protocol where two other team members must approve each developer's work before merging commits into the main project. Another risk that could occur is if the current sprint's scope exceeds the initial plan. Our strategy for this scenario would be to redefine current requirements, leveraging the flexibility provided by the Agile framework. Miscommunication can also play a large role in impacting the progress of the project, leading to misunderstandings and errors. To avoid these potential setbacks, the team will regularly meet and establish clear protocols similar to what was mentioned earlier for our testing environment. Furthermore, we will track each member's progress and updates through GitHub to prevent hesitancy and uncertainty among the team. However, if a member cannot meet yet works on the project, they must communicate their changes on the team's Discord server to avoid potential confusion. A lack of communication with the stakeholder is another significant risk. Our plan to mitigate this is to communicate frequently with the stakeholder and utilize their feedback to make informed decisions. We will also use Jira to document to share with the stakeholder what has been completed, currently in review, in progress, and what is left to do.

Conclusion:

The document outlines critical components necessary for this website development project. It incorporates details that focus on creating a robust, scalable, maintainable, and user-friendly website with server-side and database integration. Key aspects of our project include a well-defined frontend utilizing HTML, CSS, and JavaScript to structure, style, and perform user-sequenced interactions. The backend of our website will be developed with Java and tied to the database layer built through MySQL. Connections between the frontend, backend, and database will be through the Java libraries Java Database Connectivity (JDBC) and JDK HTTP Client (JDK.http).

Our team's design decision to select Java as the backend programming language primarily stems from the entire team's large familiarity with the programming language. In addition, Java offers an extensive library which we will be utilizing to connect with the database and frontend. Furthermore, the decision to use Git and GitHub as the version control and collaboration tools is inspired from industry standards. These tools allow for seamless communication and coordination during our developmental process and a necessity for progress tracking.

In conclusion, each design decision was carefully made to reflect the project objective of delivering a user-friendly website to meet the needs of the stakeholders while maintaining scalability, efficiency, and a quality user experience.

Team members Contribution:

- **Sammy Wong:**
    - Presenter for Demo
    - Figma Development Contributions
        - Design choices
        - Quality Control
        - General assets Creation
        - Chat / Communications Page
        - Patient Payment Page
        - Landing Page
    - SDD Contributions
        - Quality Control
        - Database Development consultation
        - UML Diagram consultation
        - Sequence Diagram consultation
        - SDD Documentation: Development Environment
        - SDD Documentation: Conclusion
        - SDD Documentation: Project Scope
- **Keav'n Lor:**
    - Presenter for Demo
    - Figma Development Contributions
        - Design choices
        - Quality Control
        - General assets Creation
        - Sign-in / Register page
        - Dashboards (Patient/Staff)

        - Session Tracking page
        - Appointment & Scheduling page
        - Employee Payroll page
    - SDD Contributions
        - SDD Documentation: Project overview
        - SDD Documentation: Design Prototype
        - Sequence Diagram - Chat features ●

Kyle Tran:

- SDD Contributions
    - Quality Control
    - UML
    - Database Design
    - Sequence diagram
    - SDD Documentation: Scope/Target/Tools ●

Dann Manganti:

- Figma Development Contributions
    - Design choices
    - Quality Control
    - Appointment & Scheduling page
    - Sign-in page
- SDD Contributions
    - Rewording to Target Audience
    - Techstack & Tools ●

John Vue:

- SDD Contributions
    - UML diagram composition
    - UML note taking
    - UML quality corrections
    - UML documentation
    - SDD: Scope Review ●

Kaylina Kwong:

- Figma Development Contributions
    - Design choices
    - General assets Creation
    - About us page
    - Sign-in page
    - Chat / Communications
- SDD Contributions
    - UML
    - Database Design

- SDD Documentation: Proofread●

Brandon Casey:

- Presenter for Demo
- Figma Development Contributions
  - Quality Control
  - General assets Creation ■ About us page
  - Chat / Communications
- SDD Contributions
  - UML
  - Design Prototype ●

Erds Ferdi Mabilog:

- Figma Development Contributions
  - Session’s page
  - Sign-in page
- SDD Contributions
  - Database Schema
  - Database design quality Control
  - Database Design
  - UML Design tool