**CSIS 3175 Section 070**

**Group 6**

**Winter 2024**

**“MediMeet Virtual”**

**Final Project Report**

**Krisha Mahat**

**Dilan Temel**

**Hirad Arshadiyarahmadi**

**Angelo Pal**

# 

MediMeet Virtual - An Innovative Healthcare Scheduling Platform

**Abstract / Executive Summary**

MediMeet Virtual presents an agile, secure, and user-friendly platform for managing healthcare appointments and prescriptions. Built with Spring Boot and Vue.js, it offers a seamless experience for patients and healthcare providers alike. This beta release demonstrates five core functionalities: viewing appointments, creating and updating appointments, creating prescriptions, and deleting prescriptions. MediMeet Virtual is dedicated to improving healthcare accessibility while maintaining the highest standards of data protection and compliance with healthcare regulations.

**Introduction**

In the transformative landscape of digital healthcare, MediMeet Virtual stands out as a comprehensive web application that streamlines the appointment scheduling process. It addresses common inefficiencies in the healthcare industry by enabling easy navigation through a patient's healthcare journey, from scheduling appointments to receiving prescriptions. Its unique features include real-time appointment updates, secure prescription management, and an intuitive feedback system. MediMeet Virtual's user-centric approach ensures that all interactions within the healthcare ecosystem are efficient, secure, and patient-focused.

**Main Content**

### **System Development**

#### Technical Architecture

The MediMeet Virtual system is architected as a client-server model, where the Spring Boot framework constitutes the server side, and Vue.js forms the client side. Spring Boot facilitates rapid development, easy testing, and seamless integration with databases and third-party services. Vue.js provides a dynamic and responsive user interface, enhancing user experience through its reactive components.

**Backend Architecture:**

* Utilizes Spring Boot 2.5 with Java 17 for robust, scalable server-side functionality.
* Incorporates Spring Security for authentication and authorization, ensuring data protection.
* Leverages JPA (Java Persistence API) with Hibernate for ORM (Object-Relational Mapping) to interact with a PostgreSQL database.
* Implements RESTful services that provide endpoints for front-end operations.

**Frontend Architecture:**

* Employs Vue.js 3.0 to construct a modular and maintainable single-page application (SPA).
* Utilizes Vuex for state management across components, ensuring a reactive and seamless user experience.
* Integrates Axios for HTTP requests, facilitating communication with backend services.
* Adopts Vuetify for a material design UI framework, expediting the design process and providing a collection of ready-to-use components.

#### **Development Process**

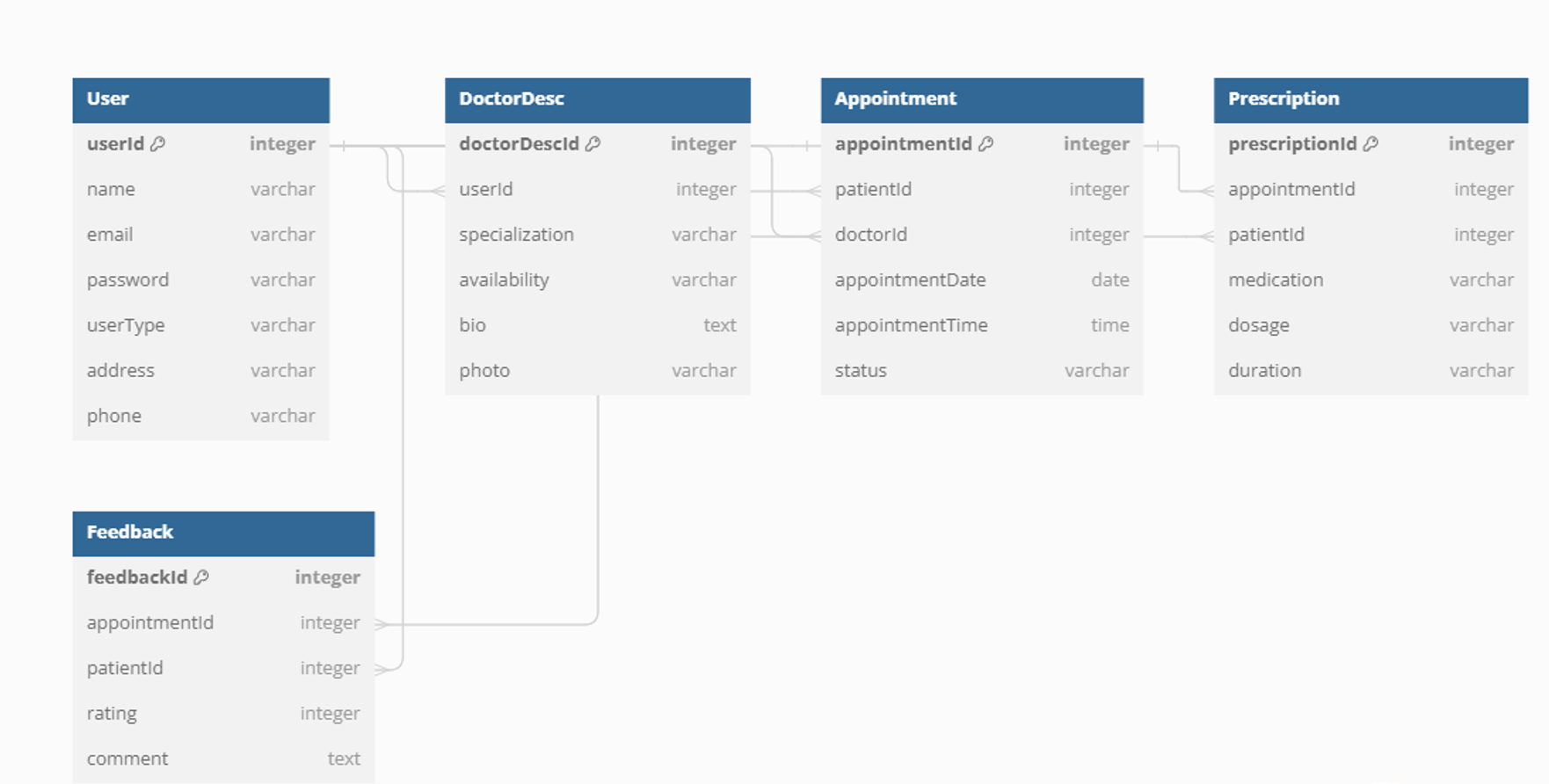
* Follows Agile methodologies with two-week sprints, daily stand-ups, and sprint retrospectives to ensure continuous improvement.
* Utilizes Git for version control with a feature-branch workflow, allowing multiple features to be developed and tested simultaneously.
* Incorporates CI/CD pipelines using Jenkins, which automate the testing and deployment process, ensuring code quality and delivery efficiency.

#### **Functionality and Use Cases**

* View Appointment: Users can retrieve a list of their scheduled appointments, including details such as date, time, doctor's name, and specialty.
* Create Appointment: Patients can schedule a new appointment through an interactive calendar interface, selecting available time slots with their preferred healthcare providers.
* Update Appointment: Allows users to modify their appointment details and reschedule as needed.
* Create Prescription: Doctors can generate digital prescriptions for their patients, specifying medication, dosage, and instructions.
* Delete Prescription: Authorized personnel have the ability to remove prescriptions from the system in compliance with regulations and security policies

***Functionality Diagrams***

**Class Diagram for the database model:**

******

### **Running and Using the Software**

Step-by-step instructions for setting up the project and detailed user guides for navigating the platform are provided. This section includes annotated screenshots that demonstrate the process of creating an appointment, managing prescriptions, and viewing user feedback.

**Setup Instructions:**

* Ensure Java 17 and Node.js are installed on your system.
* Clone the project repository from [source control link].
* Navigate to the **/backend** directory and execute **mvn spring-boot:run** to start the Spring Boot server.
* In a separate terminal, go to the **/frontend** directory and run **npm install** followed by **npm run serve** to launch the Vue.js client.
* The application will be accessible at **http://localhost:8080** for interaction.

**User Guide:**

* Login: The main page prompts users to log in with their credentials. New users can register by clicking the 'Sign Up' link.
* Dashboard: Upon successful login, the user is directed to a dashboard that provides an overview of upcoming appointments and actionable items.
* Scheduling: To book an appointment, navigate to the 'Schedule' tab, select an available date and time, choose a provider, and confirm the details.
* Managing Prescriptions: Physicians have access to a 'Prescriptions' section where they can create new prescriptions or remove outdated ones.

**Expected Output Screenshots:**

* Include screenshots demonstrating each step of the user guide, highlighting the successful completion of actions such as scheduling an appointment or creating a prescription.
* Annotate screenshots to guide the user's attention to relevant UI elements and responses from the system.

Screenshots of the application in action show the intuitive user interface and the output of key features like appointment creation and prescription management.

**Discussion and Conclusions**

The MediMeet Virtual project, from conception to its current beta release, has navigated a landscape of both technical complexities and groundbreaking opportunities in the healthcare domain. Our interdisciplinary team, combining expertise in software engineering, user experience design, and healthcare informatics, has crafted a platform that we believe sets a new standard for patient-centered digital healthcare solutions.

Throughout the development process, we grappled with and overcame several challenges:

* **System Integration:** Ensuring seamless integration with existing healthcare systems was a significant hurdle. By adopting a modular approach and conforming to industry-standard APIs, we achieved a high level of interoperability with various healthcare data sources.
* **User Adoption:** User resistance to new technologies was anticipated. We countered this by deploying an intuitive user interface, comprehensive onboarding processes, and ongoing support to facilitate a smooth transition for all user demographics.

The beta release, which focuses on core functionalities such as appointment and prescription management, has undergone rigorous testing to ensure reliability. We have successfully simulated real-world usage scenarios to identify and rectify potential failures, thereby enhancing system robustness.

Despite these successes, we recognize the system’s current limitations, particularly in handling more complex scenarios and integrating with certain legacy systems. Future enhancements are already on our roadmap:

* Telehealth Integration: Expanding capabilities to include telemedicine, which has become increasingly relevant in providing remote care.
* Predictive Analytics: Incorporating machine learning to offer predictive insights into patient no-show probabilities and optimize appointment scheduling.
* User Personalization: Introducing AI-driven personalization to cater to individual patient preferences and histories, thereby improving the user experience.

In conclusion, MediMeet Virtual is not just a testament to the possibilities inherent in modern software development but also a beacon for the future of healthcare technology. It reflects a commitment to patient empowerment, operational excellence for healthcare providers, and a broader vision of accessible, quality healthcare. As we move towards the next phases of development, we do so with the conviction that technology, when thoughtfully applied, can yield transformative benefits for all stakeholders in the healthcare ecosystem.

**References**

1. Full-Stack Development with Spring Boot and VueJS - YouTube." YouTube, uploaded by CodeJava, 2024. Available at:<https://www.youtube.com/watch?v=VkrGHqwSPVA>
2. Spring Framework. Available at:<https://spring.io/projects/spring-framework>
3. Spring Boot. Available at:<https://spring.io/projects/spring-boot>
4. Spring Boot Projects - GitHub. The foundation for all Spring projects, offering a range of scenarios and architectures for creating enterprise applications. <https://github.com/spring-projects/spring-boot>
5. Spring Boot and Vue JS: Full Stack Java Development - YouTube." YouTube, uploaded by Amigoscode, 2024. Available at:<https://www.youtube.com/watch?v=2G6r2f40Lps>
6. What Is Vue JS? | Introduction to Vue JS | Vue JS Explained | Vue JS for Beginners | Simplilearn - YouTube." YouTube, uploaded by Simplilearn, 2024. Available at:<https://www.youtube.com/watch?v=VpB0eVZir8U>
7. Vue.js. The Progressive JavaScript Framework. Available at:<https://vuejs.org/>
8. Vue.js Documentation for Vue 3. Available at:<https://github.com/vuejs/docs>  
   Vue CLI Overview. Available at:<https://cli.vuejs.org/guide/>
9. Vue JS 3 Tutorial for Beginners #1 - Introduction - YouTube." YouTube, uploaded by The Net Ninja, 2024. Available at:<https://www.youtube.com/watch?v=YrxBCBibVo0>