



UM6P COLLEGE OF COMPUTING

Agile and Scrum Report

Student :

Khaoula Jellal
Ikram Benfellaah
Yassine Blali

Professor :

Pr. Mahmoud El Hamlaoui

January 9, 2025

Contents

1 Introduction 2

2 Product Backlog 2

2.1 Epic: User Authentication 2

2.2 Epic: Appointment Management 2

2.3 Epic: Prescription Management 3

2.4 Epic: Dashboard for Health Data 4

2.5 Epic: Health Staff Interface 4

2.6 Epic: Notifications 5

3 Sprints Overview 5

3.1 Sprint 1: Project Initialization 5

3.2 Sprint 2: Authentication and Role Management 5

3.3 Sprint 3: Appointment Management 6

3.4 Sprint 4: Treatment and Medicine Management 6

3.5 Sprint 5: Notification System 6

3.6 Sprint 6: User Experience Enhancements 7

3.7 Sprint 7: Prescription Upload and OCR Integration 7

3.8 Sprint 8: Finalization and Code Review 8

4 Agile Practices Followed 8

5 Results 8

6 Conclusion 9

University Health Center Management Application

January 9, 2025

1 Introduction

This report outlines the development process of the University Health Center Management Application using Agile methodology and Scrum framework. The project aimed to address key healthcare challenges for university students and staff, focusing on real-time appointment scheduling, treatment management, and prescription handling. By following iterative and incremental development practices, the team delivered a robust and user-centered solution while adapting to evolving requirements.

2 Product Backlog

The following table summarizes the Product Backlog, listing all user stories implemented in the project.

2.1 Epic: User Authentication

- **Story:** As a student, I want to register and log in using my university credentials so that I can securely access the application.
 - **Acceptance Criteria:** Students can register using valid credentials (email and password) and log in to their accounts.
 - **Priority:** High
- **Story:** As a user, I want to reset my password so that I can regain access if I forget my credentials.
 - **Acceptance Criteria:** Password recovery option via email verification.
 - **Priority:** Medium

2.2 Epic: Appointment Management

- **Story:** As a student, I want to view a list of available health professionals and their appointment slots so that I can select the best option.

- **Acceptance Criteria:** Display list of doctors and available time slots in real time.
 - **Priority:** High
- **Story:** As a student, I want to book an appointment with a doctor so that I can secure a time slot for my visit.
 - **Acceptance Criteria:** Students can book available time slots and receive confirmation via email.
 - **Priority:** High
- **Story:** As a student, I want to cancel or reschedule an appointment so that I can adjust to changes in my schedule.
 - **Acceptance Criteria:** Students can cancel or modify an appointment, and the changes are reflected in real time.
 - **Priority:** Medium
- **Story:** As a student, I want to receive appointment confirmation notifications so that I am aware of my booking status.
 - **Acceptance Criteria:** Notifications are sent immediately upon booking or rescheduling.
 - **Priority:** Medium

2.3 Epic: Prescription Management

- **Story:** As a student, I want to manually add medication details so that I can track my treatment and follow the prescription.
 - **Acceptance Criteria:** Students can input medicine name, dosage, frequency, start date, and end date.
 - **Priority:** High
- **Story:** As a student, I want to upload a PDF of my prescription so that the system can automatically extract my treatment details.
 - **Acceptance Criteria:** Students can upload a valid prescription PDF, and the extracted data populates the treatment fields.
 - **Priority:** High
- **Story:** As a student, I want to receive medication reminders so that I don't miss my scheduled doses.
 - **Acceptance Criteria:** Notifications are sent for each medication as per the prescribed schedule.
 - **Priority:** Medium

2.4 Epic: Dashboard for Health Data

- **Story:** As a student, I want to view my upcoming appointments so that I can manage my schedule effectively.
 - **Acceptance Criteria:** Dashboard displays appointments with details such as date, time, and doctor.
 - **Priority:** High
- **Story:** As a student, I want to view my medical history so that I can keep track of my previous treatments.
 - **Acceptance Criteria:** Dashboard includes a section for medical history, including past appointments and treatments.
 - **Priority:** Medium
- **Story:** As a student, I want to view my prescription details so that I can follow my treatment plan easily.
 - **Acceptance Criteria:** Prescription details (e.g., medication name, dosage) are displayed in the dashboard.
 - **Priority:** High

2.5 Epic: Health Staff Interface

- **Story:** As a health staff member, I want to view and manage my appointments so that I can provide timely care to students.
 - **Acceptance Criteria:** A dashboard displays all booked appointments with options to confirm, modify, or cancel.
 - **Priority:** High
- **Story:** As a health staff member, I want to access student medical histories so that I can provide informed treatment.
 - **Acceptance Criteria:** Medical records are accessible based on student profiles.
 - **Priority:** High
- **Story:** As a health staff member, I want to add notes to patient records after each appointment so that I can document important observations.
 - **Acceptance Criteria:** Post-appointment notes are saved securely and linked to the student's profile.
 - **Priority:** Medium

2.6 Epic: Notifications

- **Story:** As a student, I want to receive reminders for my appointments and medications so that I stay on schedule.
 - **Acceptance Criteria:** Notifications are sent for upcoming appointments and medication reminders.
 - **Priority:** Medium

3 Sprints Overview

3.1 Sprint 1: Project Initialization

- **Goal:** Set up the foundation of the project for development.
- **Tasks Completed:**
 - Initialized the Spring Boot backend and Vue.js frontend.
 - Set up Maven for dependency management and project configuration.
 - Created the base project structure, including packages for models, services, controllers, and repositories.
 - Configured database connections and set up a basic database schema.
 - Conducted individual tests to ensure the project setup was functioning as expected.
- **Testing:** Verified that the backend and frontend could connect successfully and ensured there were no errors during project initialization.

3.2 Sprint 2: Authentication and Role Management

- **Goal:** Implement user authentication and role-based access.
- **Tasks Completed:**
 - Developed user registration and login functionality for students and health staff.
 - Implemented role-based authentication to restrict features based on user roles (e.g., students, doctors).
 - Created REST APIs for authentication and session management.
 - Configured JWT-based token authentication for secure communication.
 - Designed and implemented corresponding frontend components for login and registration.
 - Conducted unit and integration testing to ensure all authentication-related features were functional and secure.
- **Testing:** Validated the login/logout process, ensured tokens were correctly issued, and tested restricted access for different roles.

3.3 Sprint 3: Appointment Management

- **Goal:** Enable appointment scheduling and management.
- **Tasks Completed:**
 - Designed the database schema for appointments and doctor availability.
 - Implemented the ability for students to view available appointment slots and book, reschedule, or cancel appointments.
 - Developed REST APIs for appointment management, including viewing, creating, rescheduling, and canceling appointments.
 - Created a doctor's view to manage availability and view upcoming appointments.
 - Built corresponding frontend components for appointment scheduling and management.
 - Conducted unit and end-to-end testing to ensure the booking and rescheduling process worked without errors.
- **Testing:** Ensured that appointment data was stored correctly, APIs were functional, and the user interface aligned with the backend logic.

3.4 Sprint 4: Treatment and Medicine Management

- **Goal:** Introduce treatment and medicine tracking features.
- **Tasks Completed:**
 - Designed the database schema for treatments and medicines, including fields like name, dosage frequency, and start/end dates.
 - Developed the ability for students to manually add treatments and medicines.
 - Created REST APIs for managing treatments and medicines (CRUD operations).
 - Built corresponding frontend components for students to add and manage their treatments.
 - Conducted individual testing to ensure treatments and medicines were saved and displayed correctly.
- **Testing:** Verified the accuracy of treatment data, ensured proper associations with students, and conducted UI/UX usability testing.

3.5 Sprint 5: Notification System

- **Goal:** Implement notifications for appointments and medicine schedules.
- **Tasks Completed:**
 - Developed notification services for reminding students of upcoming appointments and rescheduled or canceled appointments.

- Implemented medication reminders based on prescription schedules.
 - Integrated backend notification logic with frontend display components.
 - Designed and implemented an intuitive notification center for students to view reminders.
 - Conducted unit and integration testing to ensure notifications were sent at the right time.
- **Testing:** Validated that notifications were accurate, timely, and displayed appropriately in the frontend.

3.6 Sprint 6: User Experience Enhancements

- **Goal:** Enhance the application by adding filtering options and medicine tracking.
- **Tasks Completed:**
 - Implemented filters for appointments, allowing users to view appointments by date (e.g., today, month) or category (e.g., past, upcoming).
 - Developed features for tracking daily medicine dosages based on the prescription schedule.
 - Enhanced the user interface to provide a more seamless experience for filtering and tracking.
 - Conducted unit testing and usability testing to ensure the enhancements were intuitive and error-free.
- **Testing:** Verified that filtering logic worked correctly and daily dosage tracking aligned with prescription schedules.

3.7 Sprint 7: Prescription Upload and OCR Integration

- **Goal:** Enable students to upload prescriptions and automatically update treatments.
- **Tasks Completed:**
 - Integrated Apache PDFBox for extracting text from uploaded PDF prescriptions.
 - Built a file upload API to process and validate uploaded prescriptions.
 - Implemented OCR functionality to extract treatment and medicine details from the uploaded prescriptions.
 - Linked extracted prescription data with the treatment management system, allowing automatic updates.
 - Designed a seamless frontend flow for uploading prescriptions and previewing extracted data.
 - Conducted unit, integration, and end-to-end testing to ensure accurate extraction and treatment updates.

- **Testing:** Ensured that PDFs were processed accurately and updates to treatments were reflected correctly in the UI.

3.8 Sprint 8: Finalization and Code Review

- **Goal:** Prepare the application for deployment by optimizing and finalizing code.
- **Tasks Completed:**
 - Reviewed all modules to ensure code readability, maintainability, and adherence to best practices.
 - Optimized backend performance, including caching frequently accessed data.
 - Conducted comprehensive testing of all features, including edge cases, to ensure robustness.
 - Prepared final documentation, including API references, user guides, and deployment scripts.
 - Deployed the application and conducted live testing to confirm functionality in the production environment.
- **Testing:** Ensured no regressions occurred during optimization, validated deployment scripts, and conducted a final review of all features.

4 Agile Practices Followed

- **Sprint Planning:** Each sprint began with a planning meeting to define the scope, prioritize user stories, and establish the timeline for development.
- **Backlog Grooming:** The product backlog was continuously updated throughout the project to reflect changes in priorities and evolving requirements.
- **Sprint Review:** At the end of each sprint, the team reviewed the completed features and assessed progress, ensuring alignment with the project goals.
- **Retrospective Meetings:** After each sprint, the team conducted retrospectives to evaluate what went well and identify areas for improvement in future iterations.

5 Results

- All planned features were successfully implemented within the project timeline.
- A functional, user-friendly application was developed, addressing the core requirements and challenges outlined at the beginning of the project.
- The project was completed on schedule, and feedback from peer reviews confirmed that it met the specified objectives.

6 Conclusion

By following Agile and Scrum practices, the project advanced through iterative cycles, allowing the team to efficiently deliver high-quality results while adapting to changing requirements. The University Health Center Management Application demonstrates the effectiveness of these methodologies in driving focused, incremental progress. The project successfully met its goals by maintaining flexibility, fostering collaboration, and ensuring continuous improvement, ultimately delivering a robust solution tailored to the outlined objectives.