Project Progress Report for Eevee’s Retreat

Practice Module for Certificate in Designing Modern Software Systems

28h February 2025 to 14h March 2025

**Group 7**

|  |  |
| --- | --- |
| **Name** | **Email** |
| Ho Zheng Wei | zhengwei.ho@ncs.com.sg |
| Izz Danial Bin Selamat | izzdanial.selamat@ncs.com.sg |
| James Nicolas Tan Cher Wei | james.tan@ncs.com.sg |
| Kahbelan Kalisalvam Kelaver | kahbelan.kelaver@ncs.com.sg |

**Contents**

[1. Introduction 2](#_Toc1274678564)

[1.1 Project Name & Description 3](#_Toc1259663411)

[1.2 Project Methodology 3](#_Toc1099279492)

[1.3 Project Summary 3](#_Toc69057653)

[2. Project Progress Report 4](#_Toc2145555847)

[2.1 Reporting Period 5](#_Toc1767682020)

[2.2 Sprint Objectives 5](#_Toc1824017095)

[2.3 Sprint 2 Accomplishments (Planned vs. Actual) 5](#_Toc446488155)

[2.4 Sprint 2 Burndown Chart 6](#_Toc302755160)

[2.5 Problems encountered, Action Plan, Status 6](#_Toc1380787574)

[2.6 Sprint Retrospective 6](#_Toc1338701157)

[2.6.1 What went well? 7](#_Toc1114131730)

[2.6.2 What could have been done better? 7](#_Toc809203946)

[2.6.3 What will we try next? 7](#_Toc1340298196)

[2.7 Sprint 3 Preview 7](#_Toc1010914305)

# Introduction

## **Project Name & Description**

**Eevee’s Retreat** is a web-based hotel booking system designed to make the reservation process simple, efficient, and hassle-free for customers. Guests can easily browse available rooms, check availability, book their stay, and manage their reservations all in one place.

The system also includes an admin dashboard, giving hotel staff the tools to manage room availability, pricing, reservations, and facility bookings with ease. With secure authentication, a user-friendly interface, and a streamlined booking engine, Eevee’s Retreat enhances both customer convenience and hotel operations.

## **Project Methodology**

The project follows an **Agile development methodology (SCRUM)**, ensuring an iterative and adaptive approach.

**Sprint Length:** 2 weeks per sprint

**Agile Artifacts:**

* **Sprint Planning:** Defining sprint goals and backlog prioritization
* **Daily Stand-ups:** Quick updates on progress and blockers
* **Sprint Review:** Demonstration of completed work
* **Sprint Retrospective:** Discussion on improvements for the next sprint

**Tracking & Tools:**

* GitHub Kanban Board for product backlog tracking
* GitHub for version control
* Microsoft Teams & Telegram Channel for team communication

## **Project Summary**

**Background:**

In the hospitality industry, providing a seamless and efficient booking experience is crucial for customer satisfaction and business success. Traditional hotel booking methods often lead to inefficiencies such as overbookings, manual errors, and lack of real-time availability updates. To address these challenges, there is a growing need for a modern, automated hotel booking and management system that enhances customer experience while improving hotel operations.

Eevee’s Retreat is designed as a web-based hotel management system that allows customers to browse available rooms, check availability, book their stay, and manage their reservations easily. At the same time, it provides hotel administrators with tools to efficiently manage room availability, pricing, bookings, and customer inquiries.

With an intuitive user interface, secure authentication, and a robust booking engine, the system aims to streamline hotel operations while providing a hassle-free experience for guests.

**Project Scope:**

**Deliverables**

1. A fully functional Eevee’s Retreat web application with core booking features.
2. Admin dashboard for hotel staff to manage rooms, bookings, and customer information.
3. User authentication system with secure role-based access control.
4. Database schemas & UML diagrams detailing system architecture.
5. Test cases & reports ensuring system functionality and reliability.
6. Comprehensive documentation, including user manuals and technical design specifications.

**Exclusions**

1. Integration with third-party payment gateways (payments will be manually processed in this version).
2. Mobile application development, as the focus is on a responsive web-based platform.
3. Multi-hotel chain support, as this version is tailored for a single-hotel system.

**Constraints**

1. Project timeline limitations, requiring us to focus on core booking and management features.
2. Limited familiarity with DevSecOps automation tools, requiring additional learning and setup time.
3. Resource constraints, as the team consists of a limited number of developers working within a fixed time frame.

# Project Progress Report

## **Reporting Period**

The project progress report reflects the team's progress at the end of Sprint 3, which ran from 1st March 2025 to 14th March 2025.

Sprint 3 focused primarily on implementing the core modules, such as the hotel booking system frontend UI, as well as backend implementation for the CRUD APIs required to support key functionality. The integration of basic interactions between the frontend and backend was also successfully completed. However, unit testing, bug fixing, and other advanced implementations and modules for both the frontend and backend were deferred to the next sprint to accommodate the prioritization of feature development.

## **Sprint Objectives**

The primary goals of Sprint 3 were:

* Implementing the hotel booking system frontend UI
* Developing and setting up CRUD APIs for backend functionality
* Integrating basic frontend-backend interactions
* Deferring advanced implementations and additional backend/frontend modules to Sprint 4
* Enhancing sprint workflow and backlog management

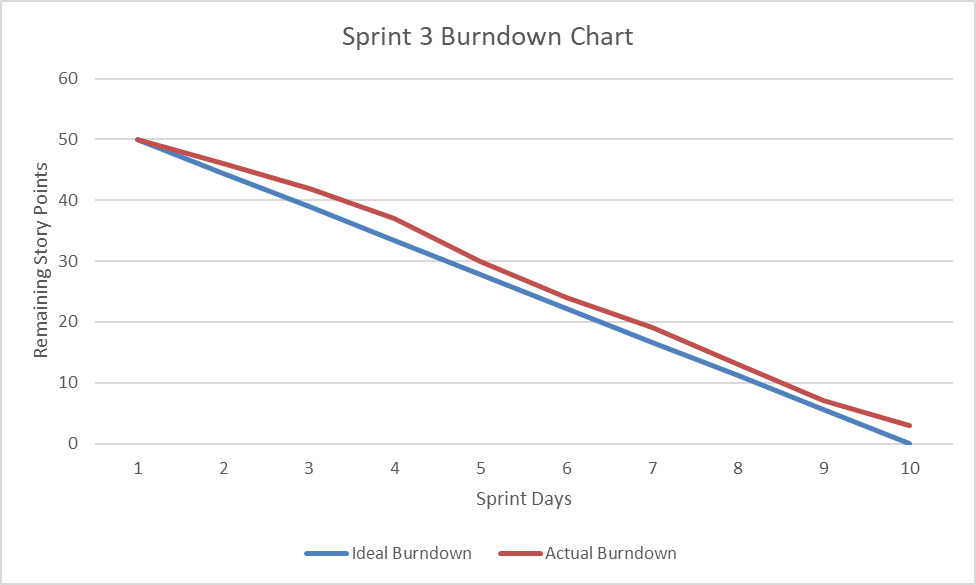
## **Sprint 3 Accomplishments (Planned vs. Actual)**

During Sprint 3, the team focused on building the fundamental components required for the hotel's booking system. The frontend UI for booking functionality was successfully implemented, providing an interactive interface for users to browse and reserve rooms. On the backend, the team developed essential CRUD APIs, ensuring robust functionality for room management, user authentication, and reservation handling. Basic frontend-backend integration was achieved, allowing for seamless interaction between the UI and the API services.

However, due to time constraints, advanced implementations, such as payment processing, enhanced UI refinements, and additional backend optimizations, were deferred to the next sprint. Similarly, unit testing and bug fixes were postponed to ensure the proper stability and reliability of the core modules before proceeding to testing.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Planned Completion | Actual Completion | Remarks |
| Implement Hotel Booking System Frontend UI | 🔄 In Progress | ✅ Completed | UI framework and layout implemented successfully |
| Develop CRUD APIs for Backend | 🔄 In Progress | ✅ Completed | Core APIs for reservations, user management, and rooms created |
| Integrate Frontend with Backend | ✅ Completed | ✅ Completed | Basic interactions between UI and API established |
| Unit Testing, Bug Fixing & Advanced Features | 🔄 In Progress | ❌ Not Started | Deferred to Sprint 4 for thorough testing and further module development |

## **Sprint 3 Burndown Chart**



## **Problems encountered, Action Plan, Status**

|  |  |  |
| --- | --- | --- |
| Problem | Action Plan | Status |
| Problem | Action Plan | Status |
| Complexities in frontend-backend integration | Conducted debugging sessions and refined API responses | ✅ Resolved |
| Limited time for unit testing, bug fixing, and advanced implementations | Deferred to Sprint 4 and prioritized for early completion | ❌ Not Started |

## **Sprint Retrospective**

### **What went well?**

The team successfully implemented the core hotel booking system UI, which provided an interactive and structured interface for users to browse and book rooms. The backend APIs were also developed and integrated with the frontend components, ensuring seamless data communication and user interaction. Additionally, backlog management and sprint planning were significantly improved, allowing for better prioritization and alignment of development tasks.

### **What could have been done better?**

A more structured approach to integrating the frontend and backend earlier in the sprint could have prevented some last-minute adjustments and integration issues. Additionally, better time allocation would have ensured that unit testing and bug fixing were not delayed to the next sprint. More collaboration between the frontend and backend teams could have helped resolve UI-related inconsistencies more proactively, reducing the need for additional fixes in the upcoming sprint.

### **What will we try next?**

In Sprint 4, the team will prioritize unit testing, bug fixing, and advanced feature implementation to ensure a stable and reliable system. Detailed UI/UX refinements and validation testing will be conducted to enhance the user experience. Additionally, backend API documentation will be improved to facilitate smoother development, debugging, and future enhancements.

## **Sprint 4 Preview**

In Sprint 4, the team will focus on completing unit testing and bug fixing to improve system stability and reliability. The remaining advanced backend and frontend modules will be implemented to enhance the functionality of the booking system. UI components will be refined and validated to ensure a smooth and user-friendly experience. Additionally, the team will work on improving error handling and API robustness to strengthen the system’s reliability.