Project Progress Report for Eevee’s Retreat

Practice Module for Certificate in Designing Modern Software Systems

14th March 2025 to 28th 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](#_Toc981665006)

[1.1 Project Name & Description 3](#_Toc324208255)

[1.2 Project Methodology 3](#_Toc1490271884)

[1.3 Project Summary 3](#_Toc237540155)

[2. Project Progress Report 4](#_Toc444116542)

[2.1 Reporting Period 5](#_Toc2061556931)

[2.2 Sprint Objectives 5](#_Toc1469052103)

[2.3 Sprint 4 Accomplishments (Planned vs. Actual) 5](#_Toc800991688)

[2.4 Sprint 4 Burndown Chart 7](#_Toc1692451405)

[2.5 Problems encountered, Action Plan, Status 7](#_Toc1072405911)

[2.6 Sprint Retrospective 7](#_Toc1077702730)

[2.6.1 What went well? 8](#_Toc527220822)

[2.6.2 What could have been done better? 8](#_Toc155492218)

[2.6.3 What will we try next? 9](#_Toc930743095)

[2.7 Sprint 4 Preview 9](#_Toc1114855435)

# 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
* Postman & Swagger – API testing and documentation

## **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**

This report reflects the progress of Sprint 4, which ran from **14th March 2025 to 28th March 2025**.

Sprint 4 focused on stabilizing the system by addressing **unit testing, bug fixing, API robustness, and UI/UX refinements**. Additionally, we implemented **role-based authentication and access control (RBAC),** ensuring secure user interactions.

Advanced backend functionalities, including **error handling improvements and booking cancellation policies**, were also developed to enhance reliability.

## **Sprint Objectives**

The primary goals of Sprint 4 were:

* Conduct **unit testing** for core functionalities (authentication, booking, room management).
* Implement **bug fixes** and **performance optimizations** based on integration feedback.
* Finalize **role-based access control (RBAC)** for different user roles (Admin, Customer).
* Enhance **UI/UX components** based on feedback from Sprint 3.
* Strengthen **error handling and API security** to improve system resilience.

## **Sprint 4 Accomplishments (Planned vs. Actual)**

During Sprint 4, the team focused on stabilizing the system by improving core functionalities, enhancing security, and refining the user experience. A major milestone achieved in this sprint was the successful implementation of unit testing and debugging for critical modules such as authentication, booking, and room management. These tests helped identify and resolve major issues, ensuring that the system functions reliably.

Additionally, Role-Based Access Control (RBAC) was finalized, enabling secure user authentication and permission management for different roles (Admin, Customer). This was a key security feature ensuring that hotel administrators have appropriate control over room management, while customers only access their own reservations.

To improve system usability and responsiveness, the team worked on UI/UX refinements, optimizing the booking flow and improving user validation messages. These changes resulted in a smoother and more intuitive experience for users navigating the platform.

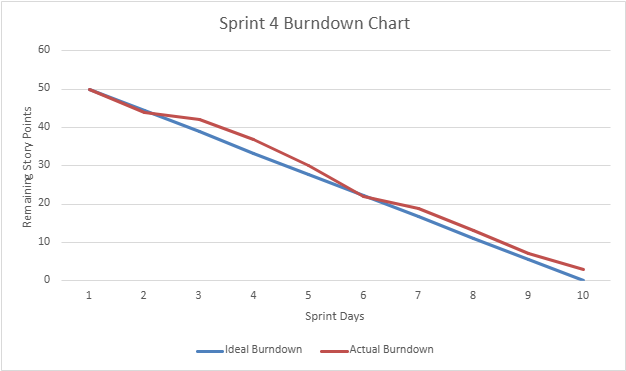
On the backend, API security was enhanced with structured error handling and validation mechanisms. This helped mitigate security risks such as unauthorized access and API misuse. Furthermore, a booking cancellation policy was implemented, allowing customers to cancel reservations within a specified timeframe, improving flexibility for users while ensuring proper room management for the hotel.

However, some challenges were encountered, particularly in ensuring seamless frontend-backend integration. Some inconsistencies in API responses led to unexpected UI behaviors, requiring additional debugging. Additionally, unit testing revealed areas that needed code optimization to enhance system performance. While these issues were successfully addressed, the extra time required for debugging delayed some of the planned testing activities.

Due to these constraints, some final performance optimizations and security audits were deferred to Sprint 4. The upcoming sprint will focus on User Acceptance Testing (UAT), final refinements, and deployment preparation, ensuring the system is stable and ready for production.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Planned Completion | Actual Completion | Remarks |
| Unit Testing (Authentication, Booking, Room Management) | ✅ Completed | ✅ Completed | Covered edge cases and common failure scenarios |
| Bug Fixes & Performance Optimization | ✅ Completed | ✅ Completed | Addressed API inconsistencies and UI responsiveness |
| **Implement Role-Based Access Control (RBAC)** | 🔄 In Progress | ✅ Completed | Admin and customer access permissions enforced |
| UI/UX Enhancements | 🔄 In Progress | ✅ Completed | Improved booking flow, validation messages, and dashboard UI |
| API Security & Error Handling Enhancements | 🔄 In Progress | ✅ Completed | Implemented structured error messages and security checks |
| Booking Cancellation Policy Implementation | ✅ Completed | ✅ Completed | Users can cancel bookings within a defined timeframe |

## **Sprint 4 Burndown Chart**



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

|  |  |  |
| --- | --- | --- |
| Problem | Action Plan | Status |
| Role-Based Access Control complexity delayed implementation | Conducted additional debugging sessions and refined role permissions | ✅ Resolved |
| UI/UX inconsistencies between frontend & backend | Improved API contracts and adjusted UI components accordingly | ✅ Resolved |
| Some unit test cases failed due to API response mismatches | Fixed inconsistencies and re-ran test cases | ✅ Resolved |
| Slow response times for booking queries due to unoptimized database calls | Optimized SQL queries and added indexing for improved performance | ✅ Resolved |

## **Sprint Retrospective**

### **What went well?**

Sprint 4 marked a significant milestone in ensuring system stability and refining security. The successful implementation of **unit testing and debugging** allowed the team to identify and fix critical issues in authentication, booking, and room management. These tests ensured that core functionalities were reliable and operated as expected, improving the overall robustness of the system.

Another major achievement was the **completion of Role-Based Access Control (RBAC)**, which provided a structured authentication and authorization system. This enhancement allowed administrators to effectively manage hotel operations while ensuring that customers only had access to their own reservations. The implementation of RBAC added a vital security layer, protecting sensitive data and preventing unauthorized access.

User experience improvements were also a key focus in this sprint. The **UI/UX refinements** enhanced the booking flow, improved validation messages, and provided a more intuitive dashboard for both customers and administrators. These enhancements made the platform easier to navigate, ensuring a seamless booking experience.

On the backend, **API security and error handling mechanisms were strengthened** to prevent vulnerabilities and improve system reliability. Clear, structured error messages were introduced, making debugging more efficient and enhancing the user experience by providing better feedback when issues occurred. Additionally, the team implemented a **booking cancellation policy**, which allowed users to modify or cancel their reservations within a specified timeframe while ensuring that hotel administrators had proper control over room availability.

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

While the sprint was successful in many aspects, some areas could have been improved. One of the primary challenges was the **lack of structured testing early in the sprint**, which resulted in certain bugs and inconsistencies only being identified towards the later stages. Although unit testing helped resolve these issues, catching them earlier in the sprint could have reduced debugging time and improved overall efficiency.

Additionally, **frontend-backend integration was more complex than anticipated**. Some inconsistencies in API responses led to unexpected UI behaviors, requiring additional debugging and refinement. A more synchronized approach to frontend and backend development—such as defining API contracts earlier and aligning expectations—could have reduced these integration issues.

Another challenge was **time allocation for testing activities**. Debugging and optimization efforts took longer than expected, leading to some tests being rushed or postponed. While key objectives were completed, a more structured approach to test planning in future sprints would help ensure that testing efforts are evenly distributed throughout the sprint.

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

In Sprint 5, the focus will shift towards **User Acceptance Testing (UAT)** to validate the system with real users. This testing phase will help identify any remaining usability issues and ensure that the platform meets user expectations before deployment.

The team will also finalize **remaining UI refinements and optimizations** based on the feedback received during Sprint 4. These refinements will ensure a polished and user-friendly interface, further enhancing the booking experience.

Additionally, the system will be **deployed to a staging environment**, where final performance testing, security audits, and bug fixes will be carried out. The team will conduct thorough **performance testing and security validation** to ensure that the system is fully optimized and resilient before its final deployment.

By prioritizing these areas in Sprint 5, the team aims to ensure that Eevee’s Retreat is stable, secure, and fully ready for production release.

## **Sprint 4 Preview**

With Sprint 4 successfully addressing system stabilization, security enhancements, and UI/UX refinements, Sprint 5 will focus on **final testing, validation, and deployment preparations**. This sprint is critical in ensuring that the system meets user expectations, functions reliably under real-world conditions, and is fully prepared for deployment.

One of the primary objectives of Sprint 5 is to conduct **User Acceptance Testing (UAT)**, where real users will interact with the system to validate its usability, functionality, and overall performance. Any issues identified during UAT will be documented, addressed, and resolved promptly to ensure a seamless user experience.

Additionally, the team will conduct **final performance testing and security audits** to optimize system efficiency and mitigate potential vulnerabilities. This will involve **load testing** to assess the system’s response under concurrent user activity and **penetration testing** to evaluate its security resilience against unauthorized access attempts.

Sprint 5 will also involve **final bug fixes and refinements** based on feedback gathered from previous sprints and UAT. The team will address minor inconsistencies, improve error handling, and make last-minute UI/UX adjustments to enhance the booking experience.

As part of the deployment process, the system will be **deployed to a staging environment**, where final configurations, database migrations, and integration checks will be completed. Once validated, the system will be prepared for its **official deployment to the AWS Free Tier** for production use.

By the end of Sprint 5, the **Eevee’s Retreat** platform is expected to be fully functional, stable, and secure, marking the successful completion of the project