UNIVERSITI PENDIDIKAN SULTAN IDRIS

اونيۏرسيتي ڤنديديقن سلطان ادريس

SULTAN IDRIS EDUCATION UNIVERSITY

SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

STUDENT HOSTEL REGISTRATION

SOFTWARE ENGINEERING PROJECT (DES3073)

NO.	NAME	MATRIC NO.
1	MUHAMAD ALI HANAFIAH BIN SABARUDIN	D20221101859
2	MOHAMAD AMIRUL AIZAD BIN ROSLEE	D20221101863
3	IQQWAN JASMAN SU BIN AZLAN SU	D20221101830
4	MUHAMMAD AIZAD BIN ADZRAUS	D20221101813
5	LUQMANUL HAKEEM BIN ZULKARNAIN	D20221101871

LECTURER: DR. AHMAD WIRAPUTRA BIN SELAMAT

24 MAC 2025

REVISION HISTORY & DOCUMENT APPROVAL

Revision History

Version	**Author**	**Major Changes**	**Date**
1.0	MOHAMAD AMIRUL AIZAD BIN ROSLEE	Initial draft of SPMP created (Introduction, Overview, Team Structure).	23 March 2025
2.0	LUQMANUL HAKEEM BIN ZULKARNAIN	Updated Project Organization: Added team roles (Leader, Developer, Admin Database, Business Analyst, Designer/Tester)	25 March 2025
3.0	MUHAMMAD AIZAD BIN MOHD ADZRAUS	Planning and Managing the development and delivery of the System.	23 March 2025
4.0	MUHAMAD ALI HANAFIAH BIN SABARUDIN	Prepare System Development Activities,Software Documentation , Gantt Chart	24 April 2025
5.0	IQQWAN JASMAN SU BIN AZLAN SU	Complete Software documentation particularly on Software Requirement Specification and Software Design Description	25 April 2025
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28 April 2025

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Document Approval

The following Software Design Documentation has been accepted and approved by the following:

Signature	Name	Date
mAli	MUHAMAD ALI HANAFIAH BIN SABARUDIN	28/4/2025
	DR. AHMAD WIRAPUTRA BIN SELAMAT	

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

TABLE OF CONTENT

1.0 INTRODUCTION

1.1 Overview

1.2 Deliveries

1.3 Evolution of SPMP

1.4 Reference Materials

1.5 Definitions and Acronyms

2.0 PROJECT ORGANIZATION

2.1 Process Model

2.2 Organizational Structure

2.3 Organizational Interface

3.0 MANAGERIAL PROCESS

3.1 Management Objectives and Priorities

3.2 Assumptions, Dependencies, and Constraints

3.3 Risk Management

3.4 Monitoring and Controlling Mechanisms

3.5 Staffing Approach

4.0 TECHNICAL PROCESS

4.1 Methods, Tools, and Techniques

4.2 Software Documentation

4.2.1 Software Requirements Specification (SRS)

4.2.2 Software Design Description (SDD)

4.2.3 Software Test Plan

4.3 User Documentation

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

4.4 Project Support Functions

5.0 WORK PACKAGES, SCHEDULE, AND BUDGET

5.1 Work Packages

5.2 Dependencies

5.3 Resource Requirements

5.4 Budget and Resource Allocation

5.5 Schedule

6.0 ADDITIONAL COMPONENTS

6.1 Index

6.2 Appendices

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

1.0 INTRODUCTION

Student hostel registration is an important process that helps ensure students have proper and organized accommodation at the university. The Hostel Registration Portal for Kolej Harun Aminurrashid at Universiti Pendidikan Sultan Idris (UPSI) will provide an online system for students to apply for hostel rooms, check their application status, and manage their hostel information. The portal will also allow administrators to manage room availability, approve applications, and monitor hostel occupancy efficiently. By replacing the current manual process, the system aims to make hostel registration faster, more accurate, and more convenient for both students and staff.

1.1 Overview

This project aims to develop a hostel registration portal for Kolej Harun Aminurrashid at Universiti Pendidikan Sultan Idris (UPSI) to simplify and digitize the current manual registration process. The system will enable students to apply for hostel rooms, track their application status, and manage their profiles, while allowing administrators to handle room allocations and generate reports efficiently. The portal will be built using free development tools and frameworks such as Laravel and Firebase (free tier), with development scheduled from Mac to July 2025. The estimated timeline includes requirements gathering, system design, development, testing, and deployment, with a total estimated budget of RM300 for incidental expenses such as printing, training materials, and potential internet or testing-related costs.

1.2 Deliveries

Deliverable	**Description**	**Delivery Date**	**Delivery Location**	**Quantity**
1.Requirements Specification Document	Detailed document outlining system features, user roles, and technical specifications.	15 May 2025	Kolej Harun Aminurrashid Office / Email Submission	1
2. UI/UX Design Mockups	High-fidelity design prototypes for both student and admin interfaces.	25 May 2025	Online via Figma link or PDF format	1 set (student & admin)
3. Portal Source	Complete beta version source	30 June	Submitted via	1

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Code (Beta Version)	code for internal testing and review.	2025	Google Drive	
4. Final Portal Source Code	Fully developed and tested system source code, ready for deployment by UPSI's IT department.	25 July 2025	Submitted via Google Drive	1
5. Final Project Report and Presentation	Summary of development process, features implemented, testing details, and future recommendations.	30 July 2025	Email submission (PDF) and real-time presentation	1

1.3 Evolution of the SPMP

The development of the hostel registration portal for Kolej Harun Aminurrashid will follow the Agile methodology, which emphasizes iterative progress, frequent feedback, and continuous improvement. The project will be divided into sprints, each lasting one weeks, with clear sprint goals and deliverables.

1.4 Reference Materials

Title	**Report/Document Number**	**Date**	**Author(s)**	**Publishing Organization**
IEEE Standard for Software Project Management Plans	IEEE Std 1058-1998	8 December 1998	IEEE Software Engineering Standards Committee	Institute of Electrical and Electronics Engineers (IEEE)

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

1.5 Definitions and Acronyms

1. **SPMP**: *Software Project Management Plan*. The controlling document for managing a software project, as defined in IEEE Std 1058-1998.
2. **UI/UX**: *User Interface / User Experience*. Refers to the design and usability aspects of the application interface.
3. **Firebase**: A platform developed by Google for creating mobile and web applications. Refer to: Firebase Documentation
4. **Flutter**: An open-source UI software development kit by Google used to develop cross-platform applications. Refer to: Flutter Documentation
5. **UPSI**: *Universiti Pendidikan Sultan Idris*, the institution under which this project is developed.
6. **KHAR**: *Kolej Harun Aminurrashid*, the target hostel for this registration system.
7. **Agile**: A software development methodology that emphasizes iterative development, customer collaboration, and responsiveness to change.
8. **Sprint**: A fixed time period during which specific work has to be completed and made ready for review (common in Agile).
9. **Github**: A web-based platform used for version control and collaborative software development.
10. **Beta Version**: A pre-release version of the software shared for testing and feedback purposes.
11. **Mockup**: A visual representation or prototype of the user interface, used for design approval and review.
12. **Bcrypt:** A cryptographic hash function designed for password hashing and safe storing in the backend of applications in a way that is less susceptible to dictionary-based cyberattacks.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

2.0 PROJECT ORGANIZATION

2.1 Process Model

For this project, we're using the **Agile development methodology**. This approach helps us stay flexible, work in short cycles (called sprints), and gather feedback regularly. Each sprint lasts about two weeks and focuses on building specific features like the login system, room booking, or admin dashboard.

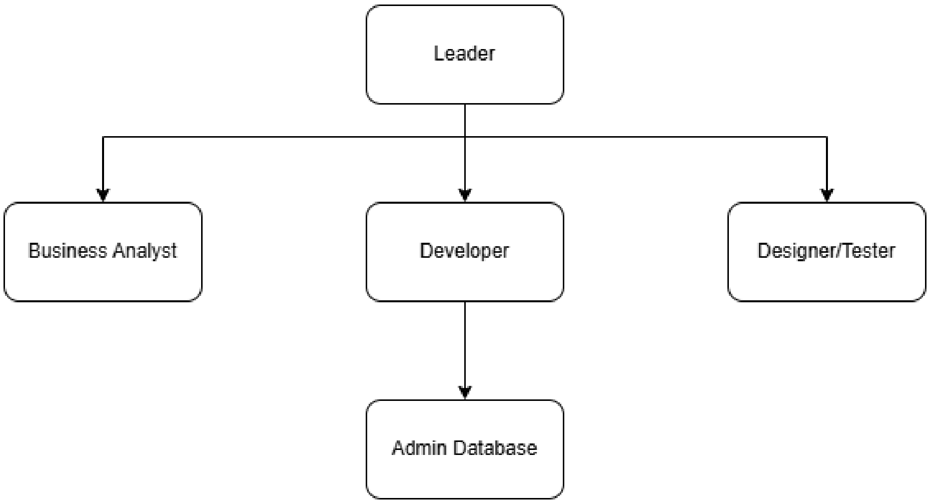
Here's how each sprint works:

- We start with **sprint planning**, where we pick which features to work on.
- Every day, we do quick check-ins to share updates and solve any issues.
- At the end of each sprint, we demo what we've built to get feedback.
- Then we reflect on what went well and what to improve for the next sprint.

This cycle repeats until the full system is complete by **July 2025**. It helps us deliver a better product step by step, instead of waiting until the very end.

2.2 Team Structure

The project team is composed of five members, each assigned specific roles and responsibilities to ensure all aspects of the project are addressed effectively:



Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Roles	Responsibility
Leader	Keeps the team on track, handles communication, and makes sure deadlines are met.
Developer	Builds the core system features using Laravel, including the student and admin portals.
Admin Database	Manages how and where all the data is stored, ensuring everything is secure and organized.
Business Analyst	Talks to users, understands their needs, and turns that into clear requirements for the team.
Designer/Tester	Designs the look and feel of the portal and tests the system to make sure it works smoothly and is easy to use.

2.3 Organizational Interface

To ensure smooth workflow and effective communication, the following interfaces have been established:

I. **Internal Communication**
The team communicates using tools like **WhatsApp** and **Google Meet** for daily coordination, while **Jira** is used for task tracking and sprint planning.

II. **Version Control and Collaboration**
GitHub is used for collaborative coding and version control, ensuring that all developers and testers work on updated codebases without conflict.

III. **Design and Feedback Interface**
UI/UX mockups are shared via **Canva** and reviewed by both the team and stakeholders for feedback and approval.

IV. **Stakeholder Communication**
Regular progress updates and sprint demos are shared with hostel administrators from **Kolej Harun Aminurrashid**, who act as key stakeholders. Feedback is collected to refine requirements and prioritize development tasks.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

V. **Tool Integration**
Laravel (backend framework) is integrated with **Firebase** for user authentication and real-time data features, managed by the Developer and Admin Database roles.

3.0 MANAGERIAL PROCESS

3.1 Management Objectives and Priorities

The Student Hostel Registration system aims to streamline the registration and room allocation process for students while ensuring data consistency and minimizing administrative workload.

The following objectives will guide the management of this project:

- Deliver a user-friendly and secure platform for hostel registration.
- Complete the project within a clearly defined timeline.
- Maintain an adaptable scope that can evolve based on feedback.

The Flexibility Matrix outlines our project's priorities across three dimensions: cost, schedule, and scope. Each dimension is assigned one of the following flexibility levels:

- **Fixed**: Must be maintained at all costs.
- **Constrained**: Can tolerate some deviation, but limited.
- **Flexible**: Can be adjusted to support other priorities.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Project Dimension	Fixed	Constrain	Flexible
Cost		x	
Schedule	x		
Scope			x

Schedule is Fixed: The project must be completed on time, possibly due to academic deadlines or deployment timelines.

Cost is Constrained: Budget limits are important but can be slightly adjusted if justified.

Scope is Flexible: While core features must be delivered, non-essential features can be refined or deferred based on progress and user feedback.

3.2 Assumptions, Dependencies, and Constraints

Assumptions

- The institution has a reliable internet infrastructure and electricity supply to host the system.
- Target users (students and hostel administrators) will have basic digital literacy.
- Required hardware (e.g., computers, servers) and software platforms (e.g., databases, web hosting) will be available and functional.
- Stakeholders (e.g., admin staff, hostel managers) will be available for periodic feedback and testing.

Dependencies

- The project depends on access to student records from the institution's existing database or student information system.
- Timely input and validation from the hostel administration for rules (e.g., room assignment criteria, eligibility).

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

- Coordination with the IT department for deployment on university servers (or setup on cloud services if applicable).

Constraints

- The system must be completed and deployed before the next academic session's hostel allocation cycle begins.
- Budget is limited, so open-source technologies are preferred.
- System should support mobile access but must initially work reliably on desktop web browsers.
- The development team may consist of students with limited availability, possibly working part-time.

Relative Priorities

- **Schedule is the top priority** — the project must meet institutional deadlines.
- **Cost is constrained** — expenses must be kept low due to limited budget.
- **Functionality is flexible** — essential features will be prioritized, while advanced features (e.g., room change requests, analytics) may be deferred to later phases.

3.3 Risk Management

To ensure the successful completion of the Student Hostel Registration System, a structured risk management process will be followed. This includes identifying, analyzing, prioritizing, and responding to potential risks throughout the project lifecycle.

Risk Identification

Risks will be identified during the planning and development phases through team brainstorming, stakeholder input, and analysis of similar past projects. Common categories of risk include:

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

- **Technical Risks:** Bugs, integration failures, or compatibility issues across different platforms.

- **Schedule Risks:** Delays due to overlapping academic responsibilities, unforeseen technical obstacles, or waiting on inputs from external stakeholders.

- **Resource Risks:** Limited availability of team members or required tools (e.g., server access, testing environments).

- **Acceptance Risks:** User resistance, lack of proper feedback from users, or poor usability affecting adoption.

- **Security Risks:** Data protection concerns, especially related to personal student information.

Risk Analysis and Prioritization

Each identified risk will be assessed based on:

- **Probability (High / Medium / Low)**

- **Impact (High / Medium / Low)**

Risks with both high probability and high impact will be addressed with highest priority. A risk matrix may be used to visualize the severity and priority of risks.

Risk Monitoring

Risks will be documented in a **Risk Register**. This document will include:

- Description of the risk

- Probability and impact ratings

- Mitigation strategy

- Responsible team member

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

- Status (Open, Monitored, Resolved)

This register will be reviewed in weekly progress meetings, and updated as the project evolves.

Risk Mitigation Strategies

Some example strategies include:

- **Technical Risks:** Use proven technologies; early prototyping and regular testing.
- **Schedule Risks:** Build in buffer time; use agile practices to adapt to changes.
- **Resource Risks:** Assign clear roles; cross-train team members where possible.
- **Acceptance Risks:** Involve users early in UI/UX testing; provide training and documentation.
- **Security Risks:** Follow best practices in data storage and encryption; restrict data access.

Contingency Planning

For high-risk scenarios, contingency plans will be prepared. For example, if the main developer becomes unavailable, another team member will be briefed regularly to take over if needed.

3.4 Monitoring and Controlling Mechanisms

To ensure that the project progresses according to the Software Project Management Plan (SPMP), systematic monitoring and control mechanisms will be implemented. These will cover progress tracking, status reporting, quality assurance, and communication across the team and stakeholders.

Monitoring Approach

Project monitoring will be conducted at the **work package level**, where each major task or module (e.g., user registration, room allocation, admin dashboard) will be assigned a responsible team member and tracked individually.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Progress will be reviewed using:

- **Task tracking boards** (e.g., Trello or Jira)
- **Weekly team meetings**
- **Milestone reviews** after key phases like design, implementation, and testing

Reporting Mechanisms

Reports will be used to provide transparency and accountability. These will include:

Report Type	Frequency	Content	Audience
Weekly Status Report	weekly	Progress, blockers, completed tasks, next steps	Project Team
Stakeholder Review	Bi-weekly	Feature demos, milestone checks, feedback	Faculty/stakeholders
Quality Assurance Report	Monthly or per build	Test results, bug reports, quality metrics	QA Team, Developers
Final Project Review	End of project	Summary of accomplishments, lessons learned	All stakeholders

Review and Audit Mechanisms

- **Code Reviews**: Regular peer reviews will ensure code quality and adherence to standards.
- **Milestone Reviews**: Conducted at the end of each phase (e.g., requirements, design, testing).
- **Configuration Audits**: To ensure correct versioning and consistency of deliverables.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Support Function Monitoring

- **Quality Assurance**: Managed through automated testing (if available) and manual test cases, tracked using a bug tracking tool.

- **Configuration Management**: Use of version control systems like Git to manage source code, track changes, and handle collaboration.

- **Documentation**: Maintained in shared drives or wikis, updated regularly as features evolve.

- **Training**: Conducted for users/admins before deployment with user manuals or short demos.

Communication Plan

A structured communication plan will ensure timely flow of information:

Communication Type	Medium	Frequency	Participants
Team Sync Meeting	In-person/Online	Weekly	Development Team
Stakeholder Check-in	Email	Bi-weekly	Project Lead, Faculty
Issue Escalation	Direct Call/Email	As Needed	Team Lead, Stakeholders
Progress Updates	Group Call	Continous	All Team Members

3.5 Staffing Approach

The success of the Student Hostel Registration System depends heavily on assembling a team with the right mix of technical and management skills. This section outlines the required skills, recruitment strategy, and necessary training for the team.

Required Skills

The project team will need the following skill sets:

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Skill Area	Description
Frontend Development	Proficiency in HTML, CSS, JavaScript, and a framework like React or Vue.js
Backend Development	Knowledge of server-side scripting (e.g., PHP, Node.js, or Python with Flask/Django)
Database Management	Experience with MySQL, PostgreSQL, or another relational database system
UI/UX Design	Ability to design user-friendly and accessible interfaces
Project Coordination	Task tracking, scheduling, and ensuring deadlines are met
Quality Assurance (QA)	Ability to design test cases and conduct both manual and automated testing
Documentation	Writing clear user manuals, technical documentation, and system guides

Recruitment Plan

Given this is an academic or student-driven project:

- Team members will be selected from students with relevant academic background or coursework (e.g., software engineering, computer science).
- Roles will be assigned based on individual strengths and interests after a brief assessment or discussion.
- Faculty or project mentors may assist in guiding team formation.

Training Requirements

Since team members may have varying levels of experience:

- Short training sessions or tutorials will be conducted on:
    - Version control tools like Git/GitHub

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

    - Chosen frameworks and libraries (e.g., Laravel, Django, React)
    - Agile methodologies or task tracking tools (e.g., Trello, Jira)
    - Secure coding and privacy practices for handling student data
- Peer learning will be encouraged — experienced members will help train others in specific tasks.

Each team member will be assigned specific roles, with opportunities for overlapping or collaborative work as needed. The general structure may include:

- **Project Manager** – responsible for scheduling, communication, and coordination
- **Frontend Developer(s)** – build the user interface
- **Backend Developer(s)** – handle logic and database interaction
- **QA Tester** – perform testing and report bugs
- **Documentation Lead** – handle user manuals and reporting

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

4.0 Technical Process

4.1 Methods, Tools, and Techniques

Computing Systems:

- **Development System:**
    - Personal computers (Windows/macOS/Linux) with PHP, Composer, and Laravel installed.

- **Target Environment:**
    - Web browsers (Chrome, Edge, Firefox) for portal access.

Development Methods:

- **Agile Development Methodology** with a **5-week sprint** structure:
    - Sprint 1: Requirements gathering and system design.
    - Sprint 2: User authentication module (student and admin login).
    - Sprint 3: Hostel registration module and room management.
    - Sprint 4: Admin dashboard and reporting.
    - Sprint 5: Testing, final adjustments, and documentation.

Standards:

- IEEE Std 1058-1998 – *IEEE Standard for Software Project Management Plans*.
- IEEE Std 610.12-1990 – *IEEE Standard Glossary of Software Engineering Terminology*.

Policies and Procedures:

- **Version Control Policy:**
    - All source code will be managed via Git and GitHub.

- **Coding Standards:**

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

    - Follow Laravel best practices (Controller-Service-Repository pattern if needed).

- **Documentation Standards:**
    - Technical and user documentation using Course Project guidelines.

- **Testing Procedure:**
    - Conduct unit testing and functional testing after each sprint using Laravel's built-in testing features (PHPUnit).

Team Structure:

Name	Role	Responsibility
Muhamad Ali Hanafiah Bin Sabarudin	Leader	Oversee project progress, conduct sprint planning, handle team coordination, manage deadlines.
Muhammad Aizad Bin Mohd Adzraus	Developer	Implement backend logic (Laravel controllers, routes, models) and API integrations.
Mohamad Amirul Aizad Bin Roslee	Admin DB	Design and manage the database (MySQL), create migrations, optimize queries, handle data integrity.
Iqqwan Jasman Su Bin Azlan Su	Business Analyst	Gather system requirements, analyze user needs, validate features, and ensure project aligns with user goals.
Luqmanul Hakeem Bin Zulkarnain	Designer / Tester	Design UI/UX (frontend with Blade and TailwindCSS), conduct manual and functional testing of all modules.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Programming Languages and Notations:

- **Backend:**
  - PHP (Laravel Framework).

- **Frontend:**
  - Blade (Laravel’s templating engine), HTML, CSS, JavaScript.

- **Database:**
  - MySQL (using Laravel's Eloquent ORM).

- **Other Tools:**
  - Laravel Migrations for database structure setup.
  - Laravel Artisan commands for maintenance and project setup.

Development Tools:

- **IDE:**
  - Visual Studio Code (with PHP, Laravel, and Blade syntax extensions).

- **Version Control:**
  - Git and GitHub.

- **Database Tools:**
  phpMyAdmin / MySQL Workbench for database management.

- **Other Tools:**
  Laravel Valet / XAMPP / Laragon (for local development server).

Techniques and Methods:

- **Design Technique:**
  - MVC (Model-View-Controller) architecture following Laravel’s structure.

- **Testing Method:**
  - PHPUnit for automated unit testing.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

    - Manual testing for user flow and usability.

- **Build/Deployment Preparation:**
    - Application will be developed for local hosting environment (without live server deployment).

4.2 Software Documentation

To ensure that the implementation of the Hostel Registration Portal satisfies the specified requirements, the following documentation will be developed:

Work Product	Description	Peer Review Type	Reviewer(s)
Software Requirements Specification (SRS)	Detailed listing of all functional and non-functional requirements for the system.	Requirements Review	KHAR
Software Design Document (SDD)	Describes the system architecture, database schema, modules, user flow diagrams, and interfaces.	Design Review	Team Leader
Software Test Plan	Outlines the testing strategies, detailed test cases, and expected results for each functionality.	Test Plan Review	Team Leader
User Manual and Developer Guide	Instructions for using the system (students and admins) and for setting up and maintaining the system (developers).	Usability Review, Documentation Review	Team Leader

- **Naming Conventions**:
  - Code: camelCase for variables and PascalCase for class names.
  - Database: snake_case for table and column names.
  - Document Sections: Numbered hierarchical format (e.g., 1.0, 1.1, 2.0).
- **Formats**:
  - Working drafts: Google Docs format.
  - Final documents: PDF format.
  - Diagrams: Created with Draw.io.
  - Version control: Managed through GitHub repositories.

The documentation effort will be planned and executed alongside the system development activities.

Task	**Start Date**	**End Date**	**Responsible Party**	**Estimated Effort**
Drafting Software Requirements Specification (SRS)	Mac 2025	Early April 2025	Project Lead	20 hours
Peer Review for SRS	Early April 2025	Mid-April 2025	All Team Members	5 hours
Drafting Software Design Document (SDD)	Mid-April 2025	Early May 2025	Developer	25 hours

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Peer Review for SDD	Early May 2025	Mid-May 2025	All Team Members	5 hours
Drafting Test Plan and Test Cases	Mid-May 2025	Early June 2025	Software Tester	20 hours
Peer Review for Test Plan	Early June 2025	Mid-June 2025	Developers	5 hours
Drafting User Manual and Developer Guide	Mid-June 2025	Early July 2025	Documentation Team	15 hours
Final Project Report Preparation	Mid-July 2025	End of July 2025	Project Lead, Team Members	15 hours

4.2.1 Software Requirement Specification (SRS)

Project Title: Student Hostel Registration

The Objective: To create a web-based system that allows users to register in groups of 4, 8, or 12 and receive room assignments based on availability. Admins manage registrations, assign rooms, and track occupancy for efficient space utilization.

Functional Requirements:

- Users can register in groups of 4, 8, or 12 and submit their group details for room assignment.
- Form fields include group name, group size, members' details, group status, and desired house/room preferences.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

- Admin users can:
    - View a list of pending group registrations.
    - Review group details and their member list.
    - Assign rooms based on group size and available space in houses.
- The system sends real-time notifications upon successful group registration and room assignment.
- Admin users can view the groups and room availability.

Non-Functional Requirements:

- The system must be **responsive** and accessible on both desktop and mobile devices.
- Data (group details, room assignments) must be securely stored in the database with proper **input validation**.
- Form submission must be **reliable**, and users will receive a **success confirmation** once their group is registered.

User Roles:

- **User (Student representative)**: Submits group registration forms and tracks group status.
- **Admin**: Manages, views and assigns rooms.

4.2.2 Software Design Description (SDD)

System Architecture: The system follows a **Model-View-Controller (MVC)** design pattern using the Laravel Framework.

- Model: Handles database interactions. For the hostel registration system, models like Group (for group details) and Room (for room assignments) will manage data operations.
- View: Blade templates are used to render forms for group registration and display the admin dashboard.
- Controller: Manages group registration submissions, validation, data storage, and admin functionalities.
    - GroupController: Handles the submission and management of group registrations.
    - AdminController: Manages admin actions such as room assignments, group list views, and filtering.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Interface Design:

- **User Interface**:
    - Clean, simple **form layout** for group registration.
    - **Validation** on input fields to ensure data accuracy (e.g., group size must be 4, 8, or 12).
    - **Real-time notifications** for success or errors during group registration.

- **Admin Interface**:
    - **Admin dashboard**: Displays a **searchable and filterable list of group registrations**.
    - **List view**: Displays pending groups with relevant details like group name, size, and status.
    - **Detail view**: Allows viewing full group details.
    - **Room assignment view**: Displays the list of rooms in each house and shows the assigned group.

Security Design:

- **Form data validation** on both the **frontend** (JavaScript validation) and **backend** (Laravel form requests) to ensure proper data input.
- **CSRF protection**: Laravel automatically handles **Cross-Site Request Forgery (CSRF)** protection with CSRF tokens to protect the system from malicious attacks.
- **User authentication**: Laravel's built-in **authentication** system to securely manage admin and user logins.
- **Password hashing**: Admin user passwords will be securely hashed using Laravel’s default *bcrypt* hashing.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

4.2.3 Software Test Plan

- Objective

  To test and validate the core functionalities of the hostel registration system:
  - User registration and group creation.
  - Admin room assignment.
  - Group status tracking.
  - Real-time notifications.

- Functional Testing

1. User Registration:
   a. Test Case 1: Verify users can register and create groups of size 4, 8, or 12.
      i. Steps: Submit registration form with valid inputs.
      ii. Expected Result: Group should be created successfully with correct group size.
2. Group Status Tracking:
   a. Test Case 2: Verify that users can see the status of their group (pending, assigned).
      i. Steps: View group status after registration.
      ii. Expected Result: Group status should reflect the correct current status (pending/assigned).
3. Room Assignment:
   a. Test Case 3: Verify admins can assign rooms to groups based on group size.
      i. Steps: Admin assigns rooms to groups of 4, 8, or 12.
      ii. Expected Result: Rooms should be assigned based on group size (4 per room).
4. Notifications:
   a. Test Case 4: Verify that users receive notifications upon room assignment.
      i. Steps: Assign a group to rooms.
      ii. Expected Result: User should receive a notification confirming room assignment.

- Pass/Fail Criteria
  - All critical functional tests pass.
  - No major defects remain unresolved.
  - System meets the required performance standards.
  - Security vulnerabilities are addressed.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

4.3 User Documentation

User documentation will include:

- **Student User Manual**
  - Explains how to register, apply for rooms, check status, and update profiles.
- **Administrator Guide**
  - Instructions for managing room allocations, viewing applications, and generating reports.
- **Online Help**
  - Context-sensitive help accessible within the portal.
- **Training Materials**
  - Quick start guides and FAQs for workshops.
- **Formats**
  - Available online (embedded within the portal).

Development of the user documentation will be planned in parallel with system testing, ensuring documentation reflects the final system.

4.4 Project Support Function

Purpose: This section describes the infrastructure, support activities, and resources required to ensure the continuous development, maintenance, and operation of the Student hostel registration.

A. Configuration Management

  a. Version Control:

  - All source code is managed using Git.
  - Github is used as the remote repository for backup and collaboration.
  - Each new feature, fix, or change will be developed in a dedicated branch before merging into the main branch.

  b. Version Naming:

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

        - ■ Versioning follows the format vMajor.Minor.Patch (e.g., v1.0.0).

B. Backup and Recovery
    a. Database Backups:
        - ■ The database will be backed up manually once a week during the development phase.
    b. Source Code Backups:
        - ■ The Github repository acts as the main backup.
        - ■ Local copies are also maintained on team members' machines.

C. Training and Knowledge Transfer
    a. User Training:
        - ■ Basic training sessions or documentation will be provided to admin users covering system usage, form management, and data exports.
    b. Developer Handover:
        - ■ Full technical documentation will be provided for future maintenance teams.

D. Maintenance and Support Plan
    a. Disclaimer: No maintenance, support plan, or continuation for this project is **intended only** for assignment prior subject DES3073(Project Management).

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

5.0 Work packages, schedule, and budget

5.1 Work packages

This section breaks down the project into smaller work packages, where each package groups related tasks together, assigns responsibility, and defines clear goals.

Work Package	Title	Description	Person-in-Charge
WP1	Requirement Analysis	Gather system requirements, define user stories, create initial draft specs.	Business Analyst (Jasman)
WP2	UI/UX Mockup Design	Design portal interface prototypes (student and admin views) for approval.	Designer/Tester (Luqman)
WP3	Student Module Development	Build and test student features (registration form, profile management).	Developer (Amirul)
WP4	Admin Module Development	Build and test admin features (room allocation, reports generation).	Developer (Amirul)+ Admin Database (Aizad)
WP5	Firebase Database Setup	Configure authentication and data storage structures on Firebase.	Admin Database (Aizad)
WP6	System Testing	Conduct functional, usability, and acceptance testing.	Designer/Tester (Luqman)
WP7	Final Documentation	Prepare user manuals, installation guides, final reports.	All team members
WP8	Sprint and Progress Reviews	Plan sprints, monitor task progress, update backlog, manage risks.	Leader(Ali Hanafiah)

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

5.2 Dependencies

To ensure all tasks are completed in the correct order, reducing risks of delays and confusion.

I. **WP1 (Requirements Analysis)** must be completed first.

   ○ All other work depends on the finalized system requirements.

II. **WP2 (UI/UX Mockup Design)** depends on WP1.

   ○ The design must align with the confirmed system functions and user needs.

III. **WP3 (Student Module Development)** depends on WP1 and WP2.

   ○ Development follows the confirmed requirements and approved UI designs.

IV. **WP4 (Admin Module Development)** depends on WP1, WP2, and WP5.

   ○ Admin features need both the database structure and the UI designs to be ready.

V. **WP5 (Firebase Database Setup)** depends on WP1.

   ○ Database structure is based on system field specifications.

VI. **WP6 (System Testing)** depends on WP3, WP4, and WP5.

   ○ Testing can only start after both student and admin modules, and the database, are functioning.

VII. **WP7 (Final Documentation)** depends on WP6.

   ○ Documentation must reflect the final tested and approved system.

VIII. **WP8 (Sprint and Progress Reviews)** is ongoing throughout the project.

   ○ Supports and monitors the progress of all work packages.

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

5.3 Resources requirement

Resources Types	Details
IDE tools	Visual Studio Code, Cannva, Laragon, GitHub
Backend technologies	Laravel 11, PHP 8.3
Database	Firebase Authentication and Firestore Database (Free Tier)
Testing tools	Browser testing, PHP unit testing
Hardware	Laptops, stable internet connection
Communication Tools	WhatsApp, Google Meet, Jira
Team members	5 Students (Leader, Developer, Admin DB, Designer/Tester, Analyst)

5.4 Budget and Resource Allocation

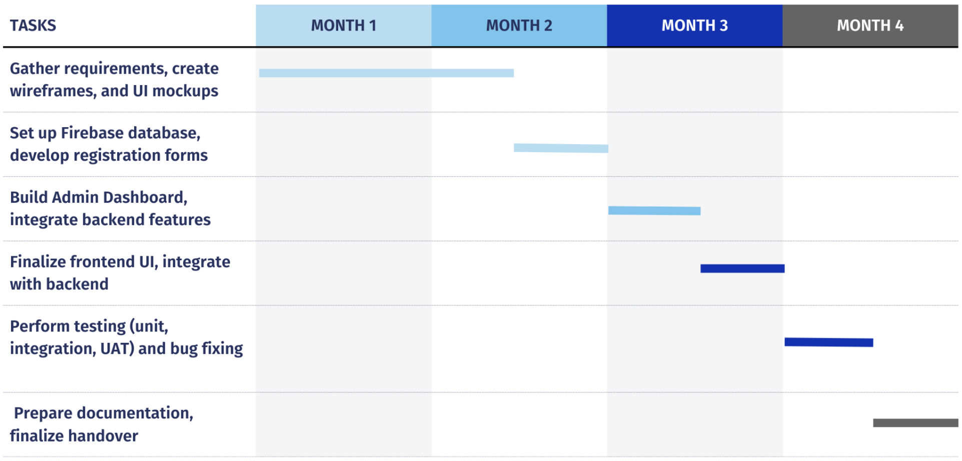
Item	Estimated cost (RM)	Purpose
Printing (reports, user guide)	50	Printing important documents for submission or backup
Design Tools (Canva free plan)	0	Using free tier
Hosting, deployment	200	Firebase Hosting upgrade or external temporary hosting
Miscellaneous	50	Unexpected minor expenses during system testing
Total	**RM300**	

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

5.5 Schedule

Sprint	Activities	Purpose
Sprint 1	Week 1-6: Gather requirements, create wireframes, and UI mockups	Finalize system requirements and design concepts
Sprint 2	Week 7-8: Set up Firebase database, develop registration forms	Build backend structure and student form functionality
Sprint 3	Week 9-10: Build Admin Dashboard, integrate backend features	Develop Admin dashboard and room allocation functionality
Sprint 4	Week 11-12: Finalize frontend UI, integrate with backend	Complete system interface and full functionality
Sprint 5	Week 13-14: Perform testing (unit, integration, UAT) and bug fixing	Ensure the system is functional, stable, and user-friendly
Sprint 6	Week 15-16: Prepare documentation, finalize handover	Complete project documentation and deliver final report

Gantt Chart



Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

6.0 ADDITIONAL COMPONENTS

Certain additional plans and supporting materials have been identified to improve the overall management and documentation quality of the Hostel Registration Portal project. These components are included either within this document or as appendices for clarity and ease of reference.

6.1 Index

An index to the key terms and acronyms used throughout the SPMP is optional, but recommended to improve usability of the SPMP.

6.2 Appendices

A. Current Top 10 Risk Chart

Risk	**Impact**	**Likelihood**	**Mitigation Strategy**
Delay in requirement gathering	High	Medium	Frequent communication with stakeholders
Misunderstanding of user needs	High	Medium	Conduct user feedback sessions
Team member availability issues	Medium	Low	Assign backup responsibilities
Data loss during development	High	Low	Regular GitHub backups
Security vulnerabilities	High	Low	Follow Laravel security best practices
Integration bugs	Medium	Medium	Continuous testing after each sprint
Time underestimation	High	Medium	Add buffer time into sprint planning

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

Internet connectivity issues	Low	Medium	Prepare offline development environment
Learning curve with Laravel	Medium	Medium	Early training and tutorials
Hardware/software failure	Medium	Low	Use redundant systems and local backups

B. Current Project Work Breakdown Structure

WBS ID	**Task**	**Team Member Responsible**
1.0	Requirements gathering and analysis	Business Analyst
2.0	System design (UI/UX, DB design)	Developer, Admin DB, Designer
3.0	Backend development (Laravel setup, authentication)	Developer
4.0	Frontend development (Blade templates)	Developer, Designer
5.0	Database creation and integration (MySQL)	Admin DB
6.0	System testing and debugging	Designer/Tester
7.0	Documentation (SRS, SDD, User Manual, Reports)	All Team Members
8.0	Final presentation and project handover	Leader

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025

C. Current Detailed Project Schedule

Sprint	**Duration**	**Main Activities**	**Output**
Sprint 1	6 May 2025 – 10 May 2025	Requirements gathering, SRS writing	SRS Document
Sprint 2	11 May 2025 – 17 May 2025	Database design, UI/UX mockup, system setup	SDD, Mockups
Sprint 3	18 May 2025 – 24 May 2025	Backend development: authentication, student registration	Partial backend system
Sprint 4	25 May 2025 – 1 June 2025	Room booking module, admin dashboard	Full system core features
Sprint 5	2 June 2025 – 8 June 2025	Testing, bug fixing, documentation finalization	User Manual, Final Report

Version	Author	Major Changes	Date
6.0	IQQWAN JASMAN SU BIN AZLAN SU	Completed Appendices (Risk Chart, WBS, Detailed Schedule), and polished document formatting.	28/4/2025