

**CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Project Management Plan

for

ALLY: Leveraging Digital Platforms to Enhance Access to Legal Aid

TABLE OF CONTENTS

1. OVERVIEW	4
1.1. PROJECT SUMMARY.....	4
1.1.1. Purpose, scope and objectives	4
1.1.2. Assumptions and constraints	5
1.1.3. Project deliverables.....	6
1.1.4. Schedule and budget summary	8
1.2. EVOLUTION OF PLAN	10
2. REFERENCES.....	11
3. DEFINITIONS	12
4. PROJECT ORGANIZATION	13
4.1. EXTERNAL STRUCTURE	13
4.2. INTERNAL STRUCTURE	14
4.3. ROLES AND RESPONSIBILITIES	14
5. MANAGERIAL PROCESS PLANS	15
5.1. START-UP PLAN	15
5.1.1. Estimation plan	15
5.1.2. Staffing plan	17
5.1.3. Resource acquisition plan.....	18
5.1.4. Project staff training plan	19
5.2. WORK PLAN.....	20
5.2.1. Work activities	20
5.2.2. Schedule allocation.....	21
5.2.3. Resource allocation	21
5.2.4. Budget allocation	21
5.3. CONTROL PLAN	22
5.3.1. Requirements control plan.....	22
5.3.2. Schedule control plan	22
5.3.3. Budget control plan	22
5.3.4. Quality control plan	22
5.3.5. Reporting plan.....	23
5.3.6. Metrics collection plan.....	23
5.3.7 Risk management plan	23
5.3.8 Project closeout plan.....	23
6. TECHNICAL PROCESS PLANS	24
6.1 PROCESS MODEL.....	24
6.2 METHODS, TOOLS, AND TECHNIQUES	24
6.3 INFRASTRUCTURE PLAN	25
6.3 PRODUCT ACCEPTANCE PLAN	25
7. SUPPORTING PROCESS PLANS	26
7.1. CONFIGURATION MANAGEMENT PLAN	26
7.2. VERIFICATION AND VALIDATION PLAN.....	26
7.3. DOCUMENTATION PLAN.....	26
7.4. QUALITY ASSURANCE PLAN	26
7.5. REVIEWS AND AUDITS	26
7.6. PROBLEM RESOLUTION PLAN	27
7.7. SUBCONTRACTOR MANAGEMENT PLAN	27
7.8. PROCESS IMPROVEMENT PLAN	27

8.0 ADDITIONAL PLANS.....	27
9. PLAN ANNEXES.....	29
10. INDEX.....	30

1. Overview

1.1. Project Summary

1.1.1. Purpose, scope and objectives

Purpose

The primary purpose of this project is to develop a secure, user-friendly, and anonymous digital platform designed to enhance access to legal aid services, particularly for underserved and vulnerable populations. The platform aims to address systemic barriers in the legal system—such as high costs, procedural complexity, limited availability of legal professionals, and concerns about privacy—by leveraging modern technologies including artificial intelligence, encrypted communication, and mobile accessibility. By streamlining legal processes and promoting inclusivity, the proposed solution seeks to empower individuals to confidently navigate legal challenges and increase overall access to justice.

Scope

This project focuses on the design and implementation of a mobile-based application that connects users to legal professionals and provides educational legal resources. Key features include anonymous AI-driven preliminary legal consultations, lawyer-client matching based on case relevance and availability, secure document uploads, encrypted messaging, real-time case tracking, and access to offline legal information. The scope also includes a lawyer dashboard for managing profiles and appointments, as well as integrated analytics for both users and legal professionals. While the platform facilitates initial consultations and document exchange, it does not replace formal legal representation or court proceedings.

Objectives

1. Main Objectives (SMART Goals):
 2. Increase platform adoption by achieving a 30% user registration growth within the first three months post-launch through an intuitive and efficient user interface.
 3. Ensure full compliance with data privacy regulations such as the Philippine Data Privacy Act of 2012 by integrating robust security protocols, including encryption and anonymization mechanisms.
 4. Achieve at least 70% user satisfaction in navigation and usability feedback surveys within the first month, guided by research-based UI/UX principles.
 5. Promote legal awareness, ensuring that over 50% of users access educational resources within four months of launch.
6. Specific Objectives (Key Deliverables):
 - Develop and deploy a functional user management system for both lawyers and clients.

- Implement AI-based case matching to connect clients with the most relevant legal professionals.
- Integrate end-to-end encrypted communication tools for confidential consultations.
- Provide personalized analytics and reports to assist users in tracking their legal concerns and interactions.
- Establish a library of legal educational resources, accessible both online and offline.

1.1.2. Assumptions and constraints

Assumptions

1. User Willingness to Adopt Digital Platforms

It is assumed that target users—including legal professionals and the general public—are willing to engage with a digital platform for legal aid services, provided it ensures security, ease of use, and anonymity. The platform's adoption is further assumed to be positively influenced by the increasing digital literacy and smartphone penetration in the Philippines.

2. Stable Internet Access in Urban Areas

The system is designed under the assumption that users, especially in urban settings, will have access to stable internet connectivity to utilize most of the platform's features. Offline access will be limited to specific functionalities, such as viewing downloaded educational materials.

3. Accurate and Honest User Input

Effective functioning of AI-powered lawyer-client matching and case assessments is assumed to rely on users providing accurate, truthful, and complete information during registration and legal queries.

4. Availability and Participation of Legal Professionals

It is assumed that a sufficient number of lawyers and legal professionals will register on the platform, maintain updated profiles, and actively respond to consultation requests, thereby ensuring timely and effective legal aid delivery.

5. Compliance with Local and International Data Privacy Laws

The system design assumes that data handling practices will comply with applicable data privacy regulations, including the Philippine Data Privacy Act of 2012 and international frameworks such as GDPR and CCPA where applicable.

Constraints

1. Functional Limitations of the Platform - The platform is constrained to provide initial consultations, resource access, and communication facilitation. It does not serve as a replacement for formal legal representation or adjudicative court processes.
2. Dependence on User-Provided Information - The system's ability to deliver accurate lawyer matching and legal guidance is limited by the completeness and reliability of the data provided by users. Inaccurate information may lead to inappropriate matches or legal misunderstandings.
3. Technical Dependence on Internet Connectivity - Most of the platform's features—including consultation, case updates, and document exchange—require active internet access. Offline functionality is restricted to static resources such as downloaded FAQs and legal guides.
4. Budgetary and Development Timeline Constraints - Project development is constrained by limited financial resources and a fixed academic timeline. As a result, some advanced features (e.g., multi-language support or extended AI capabilities) may be deferred to future phases beyond the initial launch.
5. Security and Privacy Considerations - Strict data privacy requirements may restrict certain functionalities such as long-term storage of conversation histories. Additionally, implementation of features like encrypted messaging and role-based access control must comply with existing cybersecurity standards, which may introduce development complexities.
6. Legal and Regulatory Barriers - Certain features may be restricted or unavailable due to jurisdictional limitations or evolving legal regulations concerning digital legal services, particularly in areas such as cross-border consultations and document validity.

1.1.3. Project deliverables

Project deliverables

The following are the major deliverables that will be produced throughout the development lifecycle of the project. These deliverables represent key outputs aligned with the project's objectives and scope and will serve as tangible milestones for tracking progress and quality assurance.

1. A. Documentation Deliverables
2. Software Project Management Plan (SPMP)
A comprehensive plan detailing project objectives, scope, work breakdown structure, schedule, risk management, and resource planning.
3. Software Requirements Specification (SRS)
A document outlining the functional and non-functional requirements of the system, including use cases, diagrams, and interface descriptions.
4. Software Design Description (SDD)

Detailed design documentation, including architectural design, data design, interface design, class and sequence diagrams, and module decomposition.

5. Test Plan and Test Cases

Includes test strategy, types of testing (e.g., unit, integration, system), and corresponding test cases for validating functional and non-functional requirements.

6. User Manual / Training Guide

A guide for end-users and legal professionals on how to navigate the platform, covering key features such as account creation, document upload, consultation scheduling, and legal resource access.

7. Final Capstone Report

A consolidated report summarizing all phases of the project: planning, research, design, development, evaluation, and conclusion, including results, limitations, and future work.

8. B. Software Deliverables

1. Mobile Application (Android-based)

- User Interface (Jetpack Compose)
- Anonymous user access with AI-powered legal consultation
- Secure client-lawyer messaging with end-to-end encryption
- Document submission and tracking
- Legal educational resources accessible offline
- Case ID tracking and update notifications

2. Backend API Services

- Developed using Spring Boot (Kotlin)
- RESTful API endpoints for user management, consultation handling, and analytics
- Integrated with Firebase Authentication and secure session management

3. Database Implementation

- MySQL schema with tables for users, consultations, legal documents, cases, and logs
- Encrypted data storage and access control

4. AI Case Matching System (Prototype)

- Rule-based or ML-powered logic for recommending suitable legal professionals based on user inputs (location, legal issue, urgency, etc.)

5. Admin & Lawyer Dashboard

- Profile management, case list, performance analytics, and appointment management

6. Analytics & Reporting Module

- Collects and displays user behavior and platform engagement metrics through Firebase Analytics
-

9. C. Deployment Deliverables

1. Working Application Package (.apk)

Ready-to-install APK file for Android smartphones, tested and optimized for performance and usability.

2. Deployment Plan

A document outlining the procedures and requirements for launching the platform on a cloud-based server, including final testing, configuration, and rollout.

3. Final Presentation and Demonstration

A presentation to stakeholders (advisers, panelists, end-users) showcasing the system features, architecture, and results of testing.

1.1.4. Schedule and budget summary

Phase	Description	Deliverables	Timeline
1. Requirement Analysis	Collect and document all system requirements through research and stakeholder interviews.	Software Requirements Specification (SRS)	Week 1
2. System Design	Translate requirements into high-level and detailed designs. This includes architecture, modules, and database structure.	Software Design Description (SDD), Diagrams	Week 2–3
3. Implementation	Actual development begins. Code is written per the designs. Each module is built in order.	Working software modules (e.g., Login, AI, Messaging)	Week 4–6
4. Integration and Testing	Combine modules and test the system as a whole. Bugs are identified and fixed.	Test Plan, Bug Reports, QA Logs	Week 7
5. Deployment	Deploy the working app to a cloud server and generate the .apk for Android.	APK file, deployment on Firebase/Cloud	Week 8 (start)
6. Maintenance (Post-Submission)	Handle minor bug fixes and feedback from defense panel.	Final report, defense improvements	After defense

Week	Activities	Output
Week 1	Requirements gathering, literature review, stakeholder interviews	Approved SRS
Week 2	Design system architecture, UI/UX wireframes, database ERD	Draft SDD, UML diagrams
Week 3	Complete detailed design, class diagrams, sequence diagrams	Final SDD
Week 4	Implement Module 1 (User Management), Module 2 (AI Consultation)	Authentication and AI working prototypes
Week 5	Implement Module 3 (Lawyer Matching), Module 4 (Document Upload)	Matching + upload module
Week 6	Implement Module 5 (Scheduling), Module 6 (Messaging), Module 7 (Analytics)	All major features developed
Week 7	Integrate modules, system testing, bug fixing	Final system build and test reports
Week 8	Deploy app, prepare documentation, rehearse for presentation	APK, final report, user manual, presentation

1.2. Evolution of plan

Version	Primary Author(s)	Description of Version	Date Expected
Draft	Enriquez, Piolo Frances	Initial draft created for distribution and review comments	13/03/2025
1.0	Enriquez, Piolo Frances	Initial version under configuration management	18/03/2025
1.1	Enriquez, Piolo Frances	Version after completion of Requirements and Planning phase	25/03/2025
1.2	Enriquez, Piolo Frances	Version after completion of User Management module	01/04/2025
1.3	Enriquez, Piolo Frances	Version after completion of Anonymous AI Consultation module	08/04/2025
1.4	Enriquez, Piolo Frances	Version after completion of Lawyer Matching and Document Submission	15/04/2025
1.5	Enriquez, Piolo Frances	Version after completion of Scheduling and Messaging module	25/04/2025
1.6	Enriquez, Piolo Frances	Version after integration of Analytics module and system-wide testing	02/05/2025
Final	Enriquez, Piolo Frances	Final version reflecting all project outcomes and post-deployment fixes	16/05/2025

2. References

- Domingo, R., & Reyes, L. (2021). *Poverty and Legal Exclusion in the Philippines*. Philippine Law Journal, 94(3), 45–67.
<https://law.upd.edu.ph/plj/>
- World Bank. (2022). *Philippines Economic Update: Overcoming Barriers to Inclusive Growth*.
<https://www.worldbank.org/en/country/philippines/publication/economic-update-april-2022>
- Philippine Judicial Academy (PHILJA). (2023). *Annual Report on Judicial Accessibility*.
<https://philja.judiciary.gov.ph/>
- Teufel, J., Renner, L., Gallo, M., & Hartley, C. (2021). *Income and poverty status among women experiencing intimate partner violence: A positive social return on investment from civil legal aid services*. Law & Society Review, 55(3), 405–428.
<https://doi.org/10.1111/lasr.12572>
- Goncearenco, L. (2024). *Legal aid delivery: Technology, pro bono services, and collaborative funding models*. Digital Rights Initiative.
<https://doi.org/10.53486/dri2023.01>
- Wahab, A., & Khairi, A. (2020). *Right to justice and legal aid barriers to the vulnerable non-citizens in Malaysia*. Malaysian Journal of Society and Space, 16(1).
<https://doi.org/10.17576/geo-2020-1601-02>
- Mania, K. (2022). *The Digital Transformation of Legal Industry: Management Challenges and Technological Opportunities*. DANUBE: Law and Economics Review, 13(3), 209–225.
<https://doi.org/10.2478/danb-2022-0013>
- Carter, R., & Wang, H. (2023). *The Role of Artificial Intelligence in Legal Aid: Efficiency and Ethical Challenges*. Journal of Legal Technology, 18(2), 143–160.
- Ellis, P. (2022). *Bias in AI-driven Legal Services: Risks and Mitigation Strategies*. Legal Ethics Review, 27(4), 89–107.
- Sourdin, T., & Zeleznikow, J. (2020). *Courts, Mediation and COVID-19*. SSRN Electronic Journal.
<https://doi.org/10.2139/ssrn.3595910>

Definitions

This clause of the SPMP shall define, or provide references to, documents containing the definition of all terms and acronyms required to properly understand the SPMP.

Acronym / Term	Definition
ALLY	Enhancing Access to Legal Aid Through a Secure, Anonymous, and User-Friendly Digital Platform
AI	Artificial Intelligence – used for automated legal consultations
APK	Android Package Kit – the file format used to distribute and install mobile apps
DB	Database – structured data storage used by the backend system
E2EE	End-to-End Encryption – ensures secure communication between client and lawyer
FAQ	Frequently Asked Questions – part of the legal resource library
Firebase	A platform by Google used for authentication, real-time database, and analytics
GUI	Graphical User Interface – user-facing interface design
MFA	Multi-Factor Authentication – security feature requiring two or more login factors
Kotlin	A modern programming language used for Android application development
PDF	Portable Document Format – common file format for uploading legal documents
REST API	Representational State Transfer – communication architecture used between frontend and backend
SDD	Software Design Description – outlines system architecture and design
SPMP	Software Project Management Plan – outlines project management strategies
SRS	Software Requirements Specification – formal documentation of system requirements
UI	User Interface – the visual component users interact with
UX	User Experience – the overall experience of users using the platform
JWT	JSON Web Token – used for secure user authentication
PHP	Personal History Profile – client's case background in legal form (if used)
SSL	Secure Sockets Layer – cryptographic protocol for secure internet communication

Project organization

The ALLY project will interface with several organizational entities external to the core project team. The primary external interface is with the **College of Computer Studies at Cebu Institute of Technology – University**, which serves as the academic authority overseeing the conduct of this capstone project. This includes setting academic standards, monitoring milestones, and evaluating deliverables in accordance with the Software Engineering curriculum.

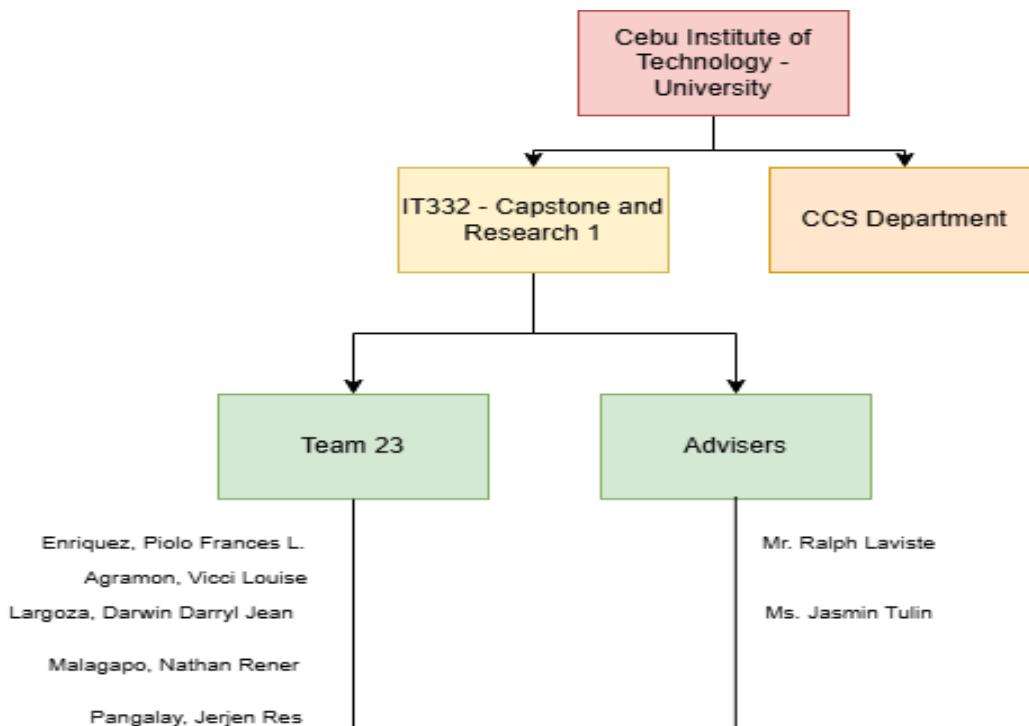
Guidance and oversight will be provided by the assigned Mrs. Jasmine Tulin, who ensures the team's adherence to best practices in software development and project management. The adviser offers academic mentorship, regularly reviews documentation, and assesses the team's performance throughout the project lifecycle. Additional oversight and evaluation will be conducted by a panel of Capstone Evaluators during formal project presentations and defenses.

Another key external interface is with the Legal Aid Community, including practicing lawyers, legal interns, and individuals with prior experience seeking legal assistance. While the platform is still in development, continuous feedback from prospective users—such as clients and legal professionals—will be actively sought. This input is critical to ensure that the ALLY platform effectively meets its intended goals of accessibility, security, and legal empowerment.

User-centered insights will be gathered through surveys, informal interviews, and prototype testing to validate system functionality and user experience. This iterative feedback loop is crucial in ensuring that the solution is practical, user-friendly, and socially impactful.

4.1. External structure

The project team coordinates with external entities for academic oversight, consultation, and validation purposes. These include:



4.2. Internal structure

The internal structure of the project team follows a functional model. Each member is assigned to specific roles and responsibilities across the project lifecycle, covering analysis, design, development, testing, and documentation.

Name	Role	Responsibilities
Enriquez, Piolo Frances	Project Leader	Oversees the overall direction of the project; ensures timely delivery of tasks; leads sprint reviews and meetings. Prepares and maintains project documents including SRS, SDD, SPMP, and user manuals.
Agramon, Vicci Louise	Backend Developer	Develops and maintains server-side logic, APIs, and database integration; ensures secure and efficient backend processes.
Lagoza, Darwin Darryl Jean	UI/UX Designer	Designs user interfaces and user experience flows; creates wireframes, prototypes, and ensures usability and accessibility.
Malagapo, Nathan Rener	Backend Developer	Collaborates on backend logic and API development; ensures data processing and storage are optimized and secure.
Pangalay, Jerjen Res	Mobile / Frontend Developer	Implements mobile application features using Android/Kotlin; integrates frontend with backend APIs; ensures mobile performance and responsiveness.

4.3. Roles and responsibilities

This subclause of the SPMP shall identify and state the nature of each major work activity and supporting process and identify the organizational units that are responsible for those processes and activities. A matrix of work activities and supporting processes vs. organizational units may be used to depict project roles and responsibilities.

S

TEAM MEMBER	ROLE & RESPONSIBILITIES	
	SEM 1	SEM 2
Enriquez, Piolo Frances L.	Project Manager / Frontend Developer	
Agramon, Vicci Louise	Backend Developer / AI Developer	
Lagoza, Darwin Darryl Jean	UI/UX Designer / Frontend Developer	
Malagapo, Nathan Rener	AI Developer / Backend Developer	
Pangalay, Jerjen Res	Mobile Developer / Frontend Developer	

[Table 4.3.1] Role and Responsibilities

Managerial process plans

This clause of the SPMP shall specify the project management processes for the project. This clause shall be consistent with the statement of project scope and shall include the project start-up plan, risk management plan, project work plan, project control plan, and project closeout plan.

5.1. Start-up plan

The start-up plan refers to the initial activities required to initiate the project, establish its foundation, and ensure all team members are aligned with the objectives, tools, and methodologies.

The following steps were conducted at project inception:

1. Project Scope Definition

The team collaboratively defined the scope and objectives of the ALLY platform, including its target users, core functionalities, and desired social impact.

2. Role Assignment and Team Structuring

Roles and responsibilities were assigned to each team member based on individual strengths, specialization, and project needs (see Section 1.4).

3. Tool Setup and Environment Preparation

Development tools such as Android Studio, IntelliJ IDEA, Firebase Console, GitHub, and project management platforms (QueueIT) were configured.

4. Documentation Planning

A documentation structure was established, including templates for the SRS, SDD, SPMP, test plan, and user manual. Google Drive was used for collaborative writing and storage.

5. Initial Timeline and Milestones

A high-level Gantt chart and weekly work plan were created, allocating time for requirement gathering, design, development, testing, and documentation.

6. Communication Strategy

Weekly meetings and daily check-ins were scheduled to ensure effective coordination. Google Meet, Messenger, and email are used as the primary communication channels.

5.1.1. Estimation plan

• Schedule, Resource, and Effort Estimates

A detailed estimation chart showing activities, estimated durations, and resource assignments is included in **Appendix B**. Since this is a student capstone project, no monetary cost estimates are applied. Instead, **effort is measured in person-hours per task**, with estimates based on member availability (approximately 20 hours per week per person).

• Estimation Methods

Schedule and work estimations for each leaf activity in the **Work Breakdown Structure (WBS)** were performed using a combination of the following methods and data sources:

- **Resource Input**
- Assigned team members were asked to estimate the time required to complete their tasks.
- Each estimate included a breakdown into **sub-activity milestones** (e.g., UI design → implementation → integration → testing).

- These milestones are tied to “% complete” progress tracking metrics and are used to monitor and assess project status effectively.
- When more than one member was involved in an activity, **individual estimates were collected**. In cases of significant variance, a consensus was reached via a discussion moderated by the Project Leader. This process was **inspired by the Wideband Delphi technique**, adapted for a small academic team setting.
 - **Academic Project History**
 - Historical data and insights from previous capstone projects, particularly those using Android, Spring Boot, and Firebase, were used to guide reasonable duration and resource estimates.
 - Guidance from the capstone adviser and references to past student schedules informed and validated proposed timelines.
 - **Bottom-Up Estimation**
 - Each major module was broken down into finer tasks. Person-hour estimates were made at the task level and rolled up to the module level.
 - Tasks were also grouped into weekly deliverables to align with academic deadlines and sprint-style milestones.
- **Re-Estimation Methods**

When re-estimation is necessary due to unexpected delays or scope clarification, the following approach will be applied:

- **Resource Input**
- Team members involved in the affected task will provide updated estimates for **remaining effort**, broken down by sub-activities.
- These will be reviewed in team meetings to discuss dependencies, blockers, or technical difficulties. Updated “% complete” metrics and milestone revisions will be recorded.
- **Adviser Feedback**
- Upon significant changes, consultation with the project adviser will be sought for external validation of the updated timeline and to ensure the adjustments remain within the academic scope.
- **Criteria for Re-Baselining**

Re-baselining of the schedule will only occur in the following cases:

- A delay or advancement of **+/- 10% of the timeline** for any major module.
- The introduction of a **new system requirement** after the design phase.
- A **technical failure or limitation** that requires a redesign of a major component.

Update Type	Handled By	Recipients Notified
Resource Adjustment	Project Leader	Entire Team
Task Duration Change	Project Leader	Team + Adviser
Major Module Delay	Project Leader	Adviser + Documented in SPMP
Scope Expansion	Project Leader (requires adviser approval)	Team + Adviser

5.1.2. Staffing plan

This subclause of the SPMP specifies the number of personnel required by skill level, the project phases in which these skills are applied, and the duration for which each team member is needed. Since ALLY is an academic capstone project, all staff are student team members. No hiring or external contracting is involved.

The staffing model assumes part-time availability throughout the semester, with increased activity during the development, testing and finalization phases. The project draws on existing internal human resources, with skills distributed across backend, mobile development, UI/UX design, and project management.

Team Composition and Availability

Name	Affiliation to Project	March 2025	April 2025	May 2025
Enriquez, Piolo Frances	Project Leader / Full Stack Dev	Part-Time	Part-Time	Part-Time
Agramon, Vicci Louise	Backend Developer	Part-Time	Part-Time	Part-Time
Lagoza, Darwin Darryl Jean	UI/UX Designer	Part-Time	Part-Time	Part-Time
Malagapo, Nathan Rener	Backend Developer	Part-Time	Part-Time	Part-Time
Pangalay, Jerjen Res	Mobile Developer	Part-Time	Part-Time	Part-Time

Skills Needed by Project Phase

Project Phase	Required Skillsets	Assigned Members	Duration
Requirements & Planning	System analysis, documentation, communication	All	1 week
UI/UX Design	Wireframing, prototyping, user experience design	Lagoza,	1 week
Backend Development	API creation, database design, logic implementation	Agramon, Malagapo,	3 weeks
Mobile App Development	Android UI implementation, integration with backend	Pangalay	3 weeks

Project Phase	Required Skillsets	Assigned Members	Duration
AI Integration	Logic design, API usage, natural language processing setup	Enriquez, Agramon, Malagapo	1 week
Testing and Debugging	Quality assurance, bug fixing, user testing	All	1 week
Deployment & Documentation	Final code integration, system deployment, report writing	All	1 week

5.1.3. Resource acquisition plan

All **human resources** required for the successful development and deployment of the **ALLY** project will be sourced internally from the capstone team. The **Project Leader**, Enriquez, Piolo Frances, is responsible for coordinating the assignment of roles and responsibilities across team members based on individual skillsets and academic schedules.

The acquisition and use of **non-human resources** will rely solely on freely available, open-source, or educational-use tools. No monetary expenses are required. However, this section outlines the necessary non-human resources, their functions, timelines, and access procedures for effective project execution.

Human Resources

Resource Type	Source	Acquisition Method
Project Team Members	BSIT Capstone Team, CIT University	Assigned based on project proposal and internal team agreement
Technical Guidance	Capstone Adviser and Instructors (CIT-U Faculty)	Provided through scheduled consultations and regular project reviews
User Feedback	Legal professionals, mock users (voluntary)	Identified through academic network, informal interviews, and test cases

Non Human Resources

- **A. Development Tools**
 - Required Tools: Android Studio, IntelliJ IDEA (Community Edition), Visual Studio Code
 - Required Dates: Continuous (Mar–May 2025)
 - Source: Free download from official websites
 - Acquisition: Installed individually by each team member
 - Remarks: All tools are free and open-source or available under student licenses.
- **B. Version Control and Repository**
 - Required tool: Github
 - Required dates: February 10, 2025 – May 17, 2025
 - Source: Github
 - Request by: Already acquired prior to development phase
 - Remarks: Provides unlimited private repositories and collaborative features.
- **C. Cloud Services**
 - Required tool: Firebase, Firestore, Cloud Storage
 - Required dates: February 10, 2025 – May 17, 2025

- Source: Google Firebase Free Tier
- Request by: Already acquired prior to development phase
- Remarks: Sufficient for development, testing, and academic presentation. May require optimization for production-scale use in the future.
- **D. Artificial Intelligence Services**
 - Required Tools: OpenAI GPT API (Free-tier/trial credits), or rule-based fallback
 - Required Dates: 25/03/2025 – 25/04/2025 (for AI consultation module)
 - Request By: 20/03/2025
 - Source: OpenAI (API access via trial)
 - Remarks: Only used for prototype-level integration; logic is fallback to rule-based if free credits are insufficient.
- **E. Documentation and Productivity Tools**
 - Required Tools: Google Docs, Google Slides, Canva, Lucidchart
 - Required Dates: Throughout the project (Mar–May 2025)
 - Source: Free accounts or education plans
 - Remarks: Used for collaborative documentation, diagrams, reports, and final presentation materials.

5.1.4. Project staff training plan

No formal training program will be conducted for the project participants of the ALLY Capstone Project. All team members have acquired the foundational knowledge and technical competencies required through their academic coursework as Bachelor of Science in Information Technology (BSIT) students at **Cebu Institute of Technology – University**. Each member has undergone prior training and practical exercises in software development subjects, including:

- Mobile Application Development (Kotlin/Android Studio)
- Web Development and API Integration (Spring Boot)
- Human-Computer Interaction (UI/UX Design)
- Software Engineering and Project Management
- Database Management Systems (MySQL, Firebase)

In addition, the team members are familiar with the **Waterfall software development life cycle (SDLC)** and its associated phases, as applied in previous coursework and academic projects.

To supplement any minor knowledge gaps, the team will rely on **self-directed learning**, such as:

- Official documentation of Android, Kotlin, Spring Boot, and Firebase
- Online courses, video tutorials, and forums (e.g., Stack Overflow, Google Developer)
- Peer-to-peer mentoring and collaborative debugging during weekly meetings

Domain-Specific Familiarization

While the team does not have direct professional experience in the legal aid domain, this limitation is addressed through:

- **Adviser consultations:** Regular meetings with the assigned capstone adviser to review system scope, legal context, and privacy requirements.
- **User research and interviews:** Informal interactions with law students and legal practitioners to

understand the processes and pain points in accessing legal services.

- **Reference materials:** Reviewing legal aid frameworks, access to justice studies, and government legal service platforms (e.g., PAO, DOJ).

This approach ensures that the system is designed with consideration for both technical feasibility and user relevance.

5.2. **Work plan**

This clause of the SPMP shall specify the work activities, schedule, resources, and budget details for the software project.

5.2.1. **Work activities**

This subclause of the SPMP shall specify the various work activities to be performed in the software project. A work breakdown structure shall be used to depict the work activities and the relationships among work activities. Work activities should be decomposed to a level that exposes all project risk factors and allows accurate estimate of resource requirements and schedule duration for each work activity. Work packages should be used to specify, for each work activity, factors such as the necessary resources, estimated duration, work products to be produced, acceptance criteria for the work products, and predecessor and successor work activities. The level of decomposition for different work activities in the work breakdown structure may be different depending on factors such as the quality of the requirements, familiarity of the work, and novelty of the technology to be used.

Activity	Start Date	End Date
1. Requirements Gathering & Analysis	2025-03-01	2025-03-10
2. System Design	2025-03-11	2025-03-25
2.1 Architectural Design	2025-03-11	2025-03-15
2.2 Detailed Module Design	2025-03-16	2025-03-25
3. Development	2025-03-26	2025-05-10
3.1 Frontend Development	2025-03-26	2025-04-20
3.2 Backend Development	2025-03-26	2025-04-30
3.3 AI Module Integration	2025-04-15	2025-05-10
4. Testing & Quality Assurance	2025-05-11	2025-05-20
4.1 Unit Testing	2025-05-11	2025-05-14
4.2 Integration Testing	2025-05-15	2025-05-17
4.3 User Acceptance Testing	2025-05-18	2025-05-20
5. Deployment & Release	2025-05-21	2025-05-25
6. Documentation & Training	2025-04-25	2025-05-25
7. Project Review & Closure	2025-05-26	2025-05-30

5.2.2. Schedule allocation

The project schedule is divided into three main cycles to ensure a clear and manageable progression from inception to completion. The first cycle focuses on requirements gathering and system design, which are crucial for establishing a solid understanding of project objectives and technical specifications. This phase runs from March 1 to March 25, allowing sufficient time to engage stakeholders and finalize design documents. Proper planning at this stage reduces the risk of major changes during development and helps in setting realistic expectations.

The second cycle encompasses the development phase, starting on March 26 and concluding on May 10. During this period, the project team undertakes frontend and backend development along with AI module integration. The extended timeline for this phase accounts for the complexity involved in coding, integration, and initial testing. Buffer periods are also incorporated to accommodate any unforeseen technical challenges or rework, ensuring that quality is not compromised in favor of speed.

The final cycle, from May 11 to May 30, is dedicated to testing, deployment, documentation, and project closure. Multiple testing layers, including unit, integration, and user acceptance testing, are conducted to ensure the system meets all functional and quality requirements. Deployment is planned immediately after testing to facilitate a smooth transition to production. The project closes with comprehensive documentation and review sessions that capture lessons learned and prepare the team for maintenance and future enhancements. Weekly progress meetings throughout all cycles help to monitor adherence to the schedule and adjust plans proactively.

5.2.3. Resource allocation

Resource allocation is carefully designed to align with the project phases and to leverage team members' expertise optimally. The Project Manager assumes overall responsibility throughout all cycles, coordinating activities, managing risks, and ensuring communication among stakeholders. During the initial cycle, the Business Analyst and System Architect focus on eliciting detailed requirements and designing the system architecture, which are foundational to subsequent work.

Development roles are concentrated in the second cycle, with Frontend and Backend Developers tasked with building the user interface and server-side logic, respectively. The AI Specialist focuses on integrating and fine-tuning the AI modules, a task that requires specialized skills and careful coordination with other developers. This clear role delineation supports parallel work streams and maximizes productivity, while frequent integration checkpoints reduce the risk of component mismatches.

The final cycle involves quality assurance and deployment specialists who lead the testing, system validation, and rollout activities. The QA/Test Engineer ensures the system meets functional and non-functional requirements through rigorous testing processes. Meanwhile, the Technical Writer documents system functionalities and creates user training materials to support end-user adoption. The Deployment Engineer manages the release process, ensuring a smooth transition to the live environment. This structured allocation helps prevent resource conflicts and ensures timely completion of each phase.

5.2.4. Budget allocation

The project budget is primarily allocated to personnel costs, reflecting the human capital necessary for successful project execution. Given the multidisciplinary nature of the team, salaries for roles such as project management, development, AI integration, testing, and documentation represent the largest expenditure. This investment ensures that skilled professionals are dedicated to delivering a high-quality product within the project timeframe.

Software licenses and technology-related expenses form the second major budget category. These funds cover essential development tools, AI APIs, and testing frameworks that provide the technological backbone for the project. Investment in up-to-date software and licenses minimizes development delays caused by inadequate tooling and facilitates efficient collaboration among team members.

Additionally, the budget accounts for hardware and infrastructure costs necessary to support development and deployment environments. Training and documentation expenses are allocated to prepare users and stakeholders, ensuring effective system adoption and long-term sustainability. A contingency fund is reserved to handle unexpected expenses, providing financial flexibility and reducing the risk of budget overruns. Miscellaneous expenses such as communication and administrative costs are also included to cover the everyday operational needs of the project team.

5.3. Control plan

The Control Plan outlines the procedures and mechanisms established to monitor, regulate, and guide the project activities to ensure objectives are met within defined constraints. It serves as the overarching framework that integrates all control activities across requirements, schedule, budget, quality, reporting, metrics collection, and risk management. Control measures include periodic reviews, audits, status meetings, and the use of project management tools to track progress. Deviations from the project baseline are identified early through continuous monitoring, enabling corrective actions to be implemented promptly. This plan ensures transparency, accountability, and alignment among all stakeholders throughout the project lifecycle.

5.3.1. Requirements control plan

The Requirements Control Plan specifies how the project requirements will be managed, tracked, and verified to prevent scope creep and ensure completeness and clarity. All requirements will be documented in a centralized repository and subjected to version control. Changes to requirements must follow a formal change control process, including impact analysis and approval by the project steering committee. Traceability matrices will be employed to map requirements to design, development, and testing activities, facilitating verification and validation. Regular requirements reviews with stakeholders will confirm that the project remains aligned with user needs and expectations.

5.3.2. Schedule control plan

The Schedule Control Plan defines the methods and tools used to monitor project progress against the established timeline. Project activities will be tracked using a detailed work breakdown structure integrated into a project management software, enabling real-time updates. Weekly status meetings will be held to review completed and pending tasks, identify schedule variances, and assess risks to the timeline. Any delay or bottleneck will trigger a corrective action plan, including resource reallocation or scope adjustment, to realign the schedule with project objectives. Key milestones and deliverables will be closely monitored to maintain overall project momentum.

5.3.3. Budget control plan

The Budget Control Plan governs the monitoring and management of project costs to ensure expenditures do not exceed approved allocations. All budget items will be tracked using financial management software with regular reconciliation of actual versus planned expenses. Monthly budget reviews will involve the project manager and finance officer to identify variances and forecast future spending. Any anticipated cost overruns will require a formal justification and approval before additional funds are allocated. The plan includes contingency management to address unforeseen expenses while maintaining fiscal discipline.

5.3.4. Quality control plan

The Quality Control Plan establishes the quality standards and procedures to ensure the project

deliverables meet specified requirements and stakeholder expectations. Quality assurance activities include peer reviews, code inspections, and formal testing phases such as unit, integration, and user acceptance testing. Defects and non-conformances will be logged and tracked until resolved. The plan also defines acceptance criteria for each deliverable and incorporates continuous improvement through lessons learned sessions. Adherence to industry standards and best practices will guide all quality-related activities.

5.3.5. Reporting plan

The Reporting Plan describes the communication strategy for disseminating project status, issues, and achievements to stakeholders. Regular reports will be generated, including weekly progress summaries, monthly status reports, and milestone completion updates. These reports will include key performance indicators such as schedule adherence, budget status, risk exposure, and quality metrics. Reports will be distributed via email and made accessible through a shared project portal. Additionally, ad hoc reports will be prepared as needed to address urgent concerns or requests from the project sponsor or steering committee.

5.3.6. Metrics collection plan

The Metrics Collection Plan details the specific performance metrics to be gathered, the methods of collection, and their frequency to provide objective data for project evaluation. Metrics will cover scope, schedule, cost, quality, and risk factors, such as requirements volatility, task completion rates, budget variance, defect density, and risk occurrence rates. Data will be collected through automated project management tools and manual reporting. The metrics will be analyzed periodically to identify trends and inform decision-making, enabling proactive management and continuous improvement.

5.3.7 Risk management plan

The Risk Management Plan outlines the approach to identifying, analyzing, mitigating, and monitoring risks throughout the project. Risks will be identified during project planning and continuously reviewed during execution. Each risk will be documented with an assigned owner, likelihood and impact assessment, and a mitigation strategy. The project team will maintain a risk register that is updated regularly and reviewed in status meetings. Contingency plans will be developed for high-priority risks to reduce their potential impact. The plan fosters a risk-aware culture, encouraging early detection and prompt response to issues.

5.3.8 Project closeout plan

The Project Close-Out Plan defines the activities necessary to formally complete the project and transition deliverables to operational status. This includes finalizing all work products, conducting acceptance reviews with stakeholders, documenting lessons learned, and releasing project resources. A close-out report will summarize project performance against objectives, including schedule, budget, quality, and risk management outcomes. All project documentation will be archived for future reference. Post-implementation support arrangements will be clarified to ensure smooth ongoing operations and maintenance. The plan ensures an orderly closure that maximizes organizational learning and stakeholder satisfaction.

6. Technical process plans

This clause of the SPMP shall specify the development process model, the technical methods, tools, and techniques to be used to develop the various work products; plans for establishing and maintaining the project infrastructure; and the product acceptance plan.

6.1 Process Model

The development of the ALLY platform adopts an **Agile software development methodology**, characterized by iterative and incremental processes to accommodate evolving requirements and user feedback. Development activities are structured around sprint cycles, each encompassing planning, implementation, testing, and review phases. This cyclical approach enables continuous delivery of functional components and fosters responsiveness to stakeholder input.

Each sprint commences with a backlog review and planning session, wherein tasks are prioritized based on user stories and technical dependencies. The team engages in daily stand-up meetings to ensure transparency and facilitate early identification of impediments. At the conclusion of each sprint, a review session is conducted to assess deliverables against predefined acceptance criteria, followed by a retrospective to evaluate process efficiency and identify areas for improvement.

This iterative framework aligns with best practices for user-centered system development and is well-suited to the dynamic nature of the legal aid landscape addressed by the ALLY platform.

6.2 Methods, tools, and techniques

The ALLY project employs a comprehensive set of technical methods, tools, and techniques across the software development life cycle (SDLC):

- **Requirements Elicitation:** Structured interviews, stakeholder surveys, and document analysis are conducted to gather functional and non-functional requirements. User stories and use-case diagrams facilitate a shared understanding of system behavior.
- **Design Methodologies:** UML diagrams—including class, sequence, and activity diagrams—are used to model system structure and behavior. Design decisions adhere to principles of modularity, scalability, and maintainability.
- **Programming Languages and Frameworks:**
 - *Frontend:* Kotlin with Jetpack Compose for Android-based user interfaces.
 - *Backend:* Spring Boot (Kotlin) for RESTful API development.
 - *Database:* MySQL, with secure schema design and encryption protocols.
 - *AI Module:* OpenAI API integration (prototypical), with fallback to rule-based logic.
- **Development Tools:**
 - Android Studio, IntelliJ IDEA, and Visual Studio Code for integrated development.
 - GitHub for version control and collaboration.
 - Firebase for authentication, analytics, and real-time database support.
- **Testing Tools:** JUnit for unit testing, with manual and automated test scripts to support integration and system-level validation.

6.3 Infrastructure Plan

The project leverages a distributed infrastructure wherein each team member utilizes locally installed development environments (Android Studio, IntelliJ IDEA) and cloud-based tools (e.g., Firebase, GitHub). All infrastructure components adhere to accessibility, security, and cost-efficiency criteria appropriate for an academic setting. Version control is managed through GitHub repositories, which support collaborative development and continuous integration.

Development, staging, and testing environments are hosted on the Firebase platform, ensuring seamless deployment and scalability. Internal team coordination is supported through productivity tools such as Google Workspace and Draw IO in google for collaborative documentation and visualization.

6.3 Product Acceptance Plan

The product acceptance process follows a structured validation protocol to ensure deliverables meet specified functional, usability, and performance criteria. Acceptance criteria include successful execution of test cases derived from user stories, satisfaction of security and privacy benchmarks (e.g., data encryption, access control), and favorable feedback during user acceptance testing (UAT).

Formal acceptance shall be contingent upon successful deployment of the platform in a controlled environment, satisfactory demonstration of all critical features (e.g., anonymous legal consultation, document upload, AI matching), and approval by the academic adviser and evaluation panel.

Documentation of acceptance results and any identified issues will be appended to the final project report.

7. Supporting process plans

7.1. Configuration management plan

Configuration management is conducted through GitHub, which serves as the centralized repository for source code, documentation, and configuration files. A branching strategy (feature → develop → main) is employed to manage parallel development streams and minimize integration conflicts.

Change control is enforced via pull requests and code reviews. Each baseline version is tagged and documented. Configuration audits are conducted at each project milestone to ensure consistency and traceability. The project team utilizes GitHub Issues for tracking change requests and defect reports, with the Project Leader overseeing configuration control decisions.

7.2. Verification and validation plan

The verification and validation (V&V) framework ensures that the system conforms to its specifications and fulfills user requirements. Verification activities include peer reviews, code inspections, and unit testing, conducted at each phase of development. Validation is accomplished through integration testing, system testing, and user acceptance testing.

A traceability matrix links requirements to design artifacts, code modules, and test cases to ensure comprehensive coverage. V&V outputs include test plans, test reports, defect logs, and phase completion checklists, all of which are version-controlled and reviewed prior to milestone acceptance.

7.3. Documentation plan

Project documentation encompasses both deliverable and non-deliverable artifacts. Key deliverables include the Software Requirements Specification (SRS), Software Design Description (SDD), Test Plan, and User Manual. These are maintained in shared cloud storage and version-controlled through GitHub.

Each document follows a predefined template consistent with IEEE standards. The documentation process includes peer reviews and adviser validation. Roles and responsibilities for documentation are assigned based on individual expertise, with the Project Leader coordinating integration and final submission.

7.4. Quality assurance plan

The quality assurance (QA) plan outlines procedures for ensuring that software products meet predefined standards for performance, usability, and reliability. QA activities include adherence to coding standards, conduct of regular code reviews, test coverage monitoring, and regression testing.

Quality objectives are defined as measurable targets (e.g., 90% unit test coverage, 95% user satisfaction in usability surveys). Non-compliance with quality benchmarks prompts immediate corrective action and re-testing. QA results are documented in testing logs and reviewed during weekly sprint retrospectives..

7.5. Reviews and audits

Periodic reviews and audits are conducted to evaluate progress, verify compliance, and identify improvement areas. Internal peer reviews are held for each module prior to integration. Formal milestone reviews are conducted with the adviser to validate alignment with academic requirements.

Audit activities include configuration audits, code inspections, and document reviews. Outcomes are recorded in inspection reports, and corrective measures are documented and tracked through resolution logs.

7.6. Problem resolution plan

The problem resolution process is designed to systematically capture, analyze, prioritize, and resolve issues encountered during development. Issues are logged using GitHub Issues and categorized based on severity and impact.

Each issue is assigned to a team member, with resolution progress tracked through tags and comments. Critical bugs are prioritized during sprint planning sessions. A structured root cause analysis is conducted for recurring issues, and solutions are incorporated into process improvement activities.

7.7. Subcontractor management plan

As an academic capstone project, no external subcontractors are engaged. All development, testing, and documentation activities are performed in-house by student team members. In the event that external tools or services are temporarily utilized (e.g., AI APIs), usage is limited to free-tier access and does not involve formal contracts or SLAs.

7.8. Process improvement plan

Process improvement is addressed through continuous feedback loops embedded within the Agile framework. Sprint retrospectives serve as the primary mechanism for identifying inefficiencies and proposing corrective actions. Lessons learned are documented and incorporated into subsequent sprint planning.

A final process evaluation is conducted at the project's conclusion to inform future academic and professional development. Recommendations are synthesized in a "Lessons Learned" section of the final report and may inform broader process improvements at the program or institutional level.

8.0 Additional Plans

This section outlines supplementary plans that address critical considerations beyond the standard project management and technical development processes. These include provisions for privacy and security, user training, system transition, and product maintenance—each of which supports the sustainability and ethical deployment of the ALLY platform.

Privacy and Security Plan

Given the sensitive nature of legal information, robust security measures are integral to the ALLY platform. The system adheres to the **Philippine Data Privacy Act of 2012** and relevant international frameworks such as the **General Data Protection Regulation (GDPR)**.

Security features include:

- **End-to-End Encryption (E2EE)** for client-lawyer messaging.
- **Secure Socket Layer (SSL)** protocols for data transmission.
- **Role-based access control** to restrict unauthorized data access.
- **Anonymized AI interaction** to preserve user identity during consultations.

User Training Plan

To ensure effective adoption of the platform, a **comprehensive user training initiative** is included. A digital **User Manual** will be provided, containing step-by-step guides on account registration, document upload, legal consultation booking, and dashboard navigation.

Additionally:

- Legal professionals will receive instructions on managing profiles and interacting with clients.
- Clients will have access to tutorials and FAQs addressing common platform tasks.
- Training materials will be designed for both online and offline access to ensure inclusivity.

System Transition Plan

Upon completion of development and testing, the system will undergo a structured transition to the live production environment hosted on Firebase. This transition includes:

- Migration of test data to a clean production database.
- Finalization of configuration settings.
- End-user access enablement with appropriate disclaimers and support channels.

Any legacy data used during development will be purged to maintain integrity and security.

Product Maintenance Plan

Following deployment, the project team will enter a **post-submission maintenance phase** wherein minor bugs, usability concerns, and enhancement requests will be addressed. A ticketing system via GitHub will be used to log issues.

Maintenance activities include:

- **Bug fixes** based on panel or user feedback.
- **Performance optimization** to ensure scalability.
- **Documentation updates** reflecting any post-deployment changes.

9. Plan Annexes

The annexes contain supplemental artifacts that support the implementation and evaluation of the SPMP. These documents provide additional detail and are referenced throughout the main body of the plan.

- Annex A: Work Breakdown Structure (WBS)
- Hierarchical task decomposition including estimation metrics, task dependencies, and assigned team members.
- Annex B: Project Gantt Chart
- Visual timeline of key activities from requirements gathering through system deployment and closure.
- Annex C: Risk Register
- Identified project risks, severity assessment, mitigation strategies, and monitoring responsibilities.
- Annex D: Communication and Meeting Logs
- Record of formal and informal meetings, decisions made, action points, and stakeholder feedback.
- Annex E: Technical Specifications Summary
- Consolidated summary of system architecture, database schema, API routes, and data flow diagrams.

10. Index

Term	Section Reference
Agile Process Model	Section 6.1
AI Case Matching	Section 1.1.3, Section 6.2
Annexes	Section 9
Budget Allocation	Section 5.2.4
Configuration Management	Section 7.1
Deployment Plan	Section 8.3
Documentation Plan	Section 7.3
Firebase Integration	Section 6.2, 6.3
Legal Data Privacy	Section 1.1.2, Section 8.1
Product Acceptance Criteria	Section 6.4
Quality Assurance	Section 7.4
Security Protocols	Section 6.2, Section 8.1
System Transition	Section 8.3
Testing Strategies	Section 7.2
User Manual	Section 1.1.3, Section 8.2
Version Control	Section 7.1