SOFTWARE ENGINEERING TEAM PROJECT

SSE 4301-2

TASK:

LAB 1: PLAN AND PREPARE PROJECT MANAGEMENT

PROJECT MANAGER:

MUHAMMAD RAFIQ BIN ABD RASHID

PREPARED FOR:

DR SALFARINA BINTI ABDULLAH

SUPERVISED BY:

EN. MD SHAHZAR MD. IDRIS

PREPARED BY:

NO.	STUDENT NAME	MATRIC NUMBER
1	IZZAT HAKIMI BIN IKMAL HISHAM	208335
2	NOOR SYAHIRAH BINTI MOHD SABRI	207542
3	SOFEA EZZATY BINTI MOHD KHUSRI	207084
4	DANISH AL-MUHAIMIN BIN MUSTAFFA	205647
5	JULIA NURFADHILAH BINTI MOHAMAD FAUZI	208256
6	MUHAMMAD HARITH THAQIF BIN MOHD HAIRUDDIN	205823
7	MUHAMMAD RAFIQ BIN ABD RASHID	206214

PJBUMI BERHAD

Software Project Management Plan for

LoRa Based Alert System for AquaTank Water Quality (Acidity) and Fish Measurement System

Izzat Hakimi Bin Ikmal Hisham
Sofea Ezzaty Binti Mohd Khusri
Noor Syahirah Binti Mohd Sabri
Danish Al-Muhaimin Bin Mustaffa
Muhammad Rafiq Bin Abd Rashid
Julia Nurfadhilah Binti Mohamad Fauzi
Muhammad Harith Thaqif Bin Mohd Hairuddin

Version 1.1.0

26/03/2023

Document Control

Version Release	Responsible Party	Major Changes	Date
1.0.0	Rafiq	Draft Plan	17/03/2023
1.1.0	Rafiq	Final version of SPMP document	26/03/2023

Table of Contents

Table of Contents **3**
1. Introduction **5**
1.1 Project Overview 5
1.2 Project Deliverables 7
1.3 Evolution of the SPMP 8
2. Project Organization **10**
2.1 Process Model 10
2.1.1 Agile Model 11
2.1.2 Waterfall Model 14
2.2 Organisational Structure 16
2.3 Organisational Interfaces 18
2.4 Project Responsibilities 19
3. Managerial Process **20**
3.1 Management Objectives and Priorities 20
3.2 Assumptions, Dependencies, and Constraints 21
3.2.1 Assumptions 21
3.2.2 External events dependencies 22
3.2.3 Constraints 22
3.3 Risk Management 22
3.3.1 Risk Tracking and Mitigation Plan 22
3.3.2 Risk Factors and Method 23
3.4 Monitoring and Controlling Mechanisms 24
3.5 Staffing Approach 25
4. Technical Process **27**
4.1 Methods, Tools, and Techniques 27
4.1.1 Methods 27
4.1.2 Tools 27
4.1.3 Techniques 28
4.2 Software Documentation 29
4.3 User Documentation 31
4.4 Project Support Functions 32
5. Work Packages, Schedule, and Budget **35**
5.1 Work Packages 35
5.2 Dependencies 35
5.2.1 Task dependencies 35
5.2.2 External dependencies 36
5.2.2 Gantt Chart 37
5.2.3 Critical Path Method 37
5.3 Resource Requirements 37

5.4 Budget and Resource Allocation 38
5.5 Schedule 38
6. Additional Components **39**
6.1 Index 39
6.2 Appendices 39

1. Introduction

This document acts as a road map for the project team to ensure that team members are aware of the tools and procedures that will be utilised to manage the project. A list of the tools to be used, together with explanations of each tool and their intended uses, are also included in the document.

Software Project Management Plan serves as a reference for team members who may not be familiar with the tools and as a means of ensuring that the project team is aware of the tools that will be used during the project. This to ensure that team members will collaborate skillfully, that the project will be finished on schedule, and that it will stay under budget.

1.1 Project Overview

Project Name: LoRa Based Alert System for AquaTank Water Quality (Acidity) and Fish Measurement System

Project Description: The goal of this project is to design, develop, and implement a LoRa based software solution for monitoring the acidity level of aquatank water and creating an observation of the water acidity level and fish size and weight management. The software will use LoRa technology for long-range communication, and it will alert the user if the acidity level is too low or too high. The software will also create analytics based on reading the value of acidity monthly and yearly to know the pattern of acidity value. It will also capture the profile of each fish.

Objectives:

- Develop dashboard to estimate the acidity of aquatank water accurately using LoRa technology
- Create a user-friendly interface to visualise and monitor the measurement of content of each AquaTank (water acidity, marine life) level in real-time
- Alert the user if the acidity level is too low or too high
- Create analytics based on reading the value of acidity monthly and yearly to know the pattern of acidity value

- To analyse water acidity on weekly, monthly, quarterly and yearly basis
- Ensure that the dashboard is reliable, secure, and scalable

Major Work Activities:

- Design and development of the dashboard solution
- Integration of LoRa technology for long-range communication
- Development of the user interface for data visualisation and monitoring
- Implementation of the alert system for abnormal readings
- Creation of analytics based on monthly and yearly readings
- Testing and validation of the software solution

Major Milestones:

Phases	Major Milestones	Duration	Description
P1	• User Requirement • System Design (Frontend & Backend)	Week 1-8	• Identify goals and objectives of the system • Gather functional and non-functional requirements • Develop models and diagrams (use case diagrams, flowcharts, sequence diagrams) • Visualize system functionality
P2	• System Development • System Testing	Week 5-14	• Develop software code that meets design specifications • Conduct code reviews and testing • Develop test plan • Identify test cases and test data • Evaluate testing process and results
P3	System Deployment	Week 14	• Deploy system • Provide ongoing maintenance and support

Table 1.1: Project Milestones

Required Resources:

- LoRa network expertise for designing and implementing the system
- Sensor node hardware and software development resources
- Database management and analysis resources
- Web-based or mobile-based interface design and development resources
- Project management resources to oversee the project and ensure timely completion

Relationship to other Projects:

- This project is a standalone project with no direct relationship to other projects.

Reference to Official Statement of Product Requirements:

- The official statement of product requirements can be found at PJBUMI Heavy Engineering & Services Sdn Bhd.

1.2 Project Deliverables

Project deliverables for a software development project may include items such as a data visualization dashboard that is provided on an optical disk, a database integration solution that seamlessly integrates with existing systems, and an operating manual that provides live documentation and can be accessed through a power bi button. Each of these deliverables plays a critical role in the overall success of the project, ensuring that stakeholders have access to the necessary tools and resources to achieve their goals.

Primary Deliverables:

- Software solution for estimating the acidity level of aquatank water accurately using LoRa technology
- User-friendly interface for visualising and monitoring the acidity level in real-time
- Alert system for notifying the user if the acidity level is too low or too high
- Monthly and yearly analytics based on reading the value of acidity to know the pattern of acidity value

Delivery Dates:

- Software solution for estimating the acidity level of aquatank water accurately using LoRa technology: Week 4
- User-friendly interface for visualizing and monitoring the acidity level in real-time: Week 6
- Alert system for notifying the user if the acidity level is too low or too high: Week 8
- Monthly and yearly analytics based on reading the value of acidity to know the pattern of acidity value: Week 12

1.3 Evolution of the SPMP

Describe how this plan will be completed, disseminated, and put under change control. Describe how both scheduled and unscheduled updates will be handled.

Completion:

- The initial version of the SPMP will be created and approved by the project manager before the start of the project.
- The SPMP will be updated and reviewed regularly to ensure it reflects the project's current status and aligns with the project's goals.
- The SPMP will be completed once the project is successfully delivered, and all project documentation is reviewed and approved.

Dissemination:

- The SPMP will be shared with all stakeholders, including the project team and supervisor at the start of the project.
- The updated version of the SPMP will be shared with the stakeholders regularly to keep them informed of the project's progress and any changes made to the plan.
- The SPMP will be available in a shared location accessible to all stakeholders, ensuring that everyone has access to the most recent version.

Change Control:

- Changes to the SPMP will be proposed through the project manager.
- The proposed changes will be reviewed and evaluated by the project team and the supervisor.
- The updated SPMP will be approved and implemented by the project manager once all stakeholders agree to the proposed changes.

Scheduled and Unscheduled Updates:

- Scheduled updates to the SPMP will occur regularly, typically during the weekly or bi-weekly project meetings.
- Unscheduled updates to the SPMP may be made when changes to the project's scope, budget, or schedule are necessary due to unforeseen circumstances.

- Both scheduled and unscheduled updates to the SPMP will be communicated to all stakeholders through project meetings and documentation updates.

Overall, the evolution of the SPMP will be closely monitored and controlled to ensure that the project's objectives are met and that all stakeholders are kept informed of the project's progress and changes.

2. Project Organization

2.1 Process Model

The process model is important in project management to provide a structured approach that increases efficiency, reduces errors, and encourages continuous improvement. Agile is one of the popular models that are widely used in software development. Below are the justification on why Agile is preferable based on IEEE standards for the software engineering team project.

Agile Model:

The Agile model is suited for projects where the requirements are not well-defined, or the requirements are likely to change frequently. In this project, the team has identified that the requirements are not always clear, and clients are unsure about their needs. Therefore, the team has decided to choose the Agile model for the project. The Agile model is ideal for software development projects where the team is experienced and self-organising. Our team has several experienced members who can make decisions independently and respond quickly to changes. Moreover, the Agile model is flexible, allowing us to adapt to changes in the project scope or requirements. This will ensure that the team can deliver high-quality software that meets the client's needs.

Waterfall Model:

The Waterfall model is a popular development model that is suited for projects where the requirements are well-defined, and the project scope is fixed. This model emphasises documentation, which is essential for regulatory compliance and maintaining a detailed record of project activities. Waterfall model is ideal for projects where the team is inexperienced or has limited experience with the project domain. This model provides a structured approach to software development, ensuring that the team follows a well-defined process. This will ensure that we can deliver high-quality software that meets the client's needs.

2.1.1 Agile Model

Steps	Description	
1. Project Initiation	**Roles**	Project Manager, Business Analyst.
	Activities	Define project goals, create a product backlog, conduct initial sprint planning, establish team roles and responsibilities.
	Entry Criteria	Approved project charter, initial requirements, budget approval.
	Exit Criteria	Approved product backlog, sprint plan, team agreement.
2. Requirements and Analysis	**Roles**	Project Manager, Business Analyst, Software Architect, Software Designer.
	Activities	Identify and prioritise user requirements, create user stories and acceptance criteria, define the architecture and design of the software, break down user stories into tasks.
	Entry Criteria	Approved product backlog, initial user requirements, system requirements.
	Exit Criteria	Approved user stories and acceptance criteria, software design and architecture, task breakdown.
3. Implementation and Testing	**Roles**	Project Manager, Software Developer, Quality Assurance Engineer, Software Tester.

	Activities	Develop and test product features, update the sprint backlog, conduct daily stand-up meetings, review progress towards sprint goals.
	Entry Criteria	Approved sprint backlog, acceptance criteria, sprint timeline.
	Exit Criteria	Completed product increment, sprint review, updated product backlog.
4. Integration and Test	**Roles**	Project Manager, Software Developer, Quality Assurance Engineer, Software Tester.
	Activities	Integrate product features, conduct integration testing, update the sprint backlog, review progress towards sprint goals.
	Entry Criteria	Completed implementation and testing phase, acceptance criteria, sprint timeline.
	Exit Criteria	Completed product increment, sprint review, updated product backlog.
5. Acceptance and Release	**Roles**	Project Manager, Software Developer, Quality Assurance Engineer, Software Tester, Business Analyst, Operations.
	Activities	Conduct user acceptance testing, finalise user documentation, deploy the software, monitor and support, provide training and documentation.

	Entry Criteria	Completed integration and test phase, user acceptance testing results, deployment plan.
	Exit Criteria	Deployed product, user documentation, support plan, training materials.
6. Operation and Maintenance	**Roles**	Project Manager, Software Developer, Quality Assurance Engineer, Software Tester, Business Analyst, Operations.
	Activities	Monitor and maintain the software, conduct maintenance and support, update user documentation and training materials, identify and resolve defects.
	Entry Criteria	Completed acceptance and release phase, deployed product, user feedback.
	Exit Criteria	Updated user documentation and training materials, resolved defects, improved software quality.
7. Project Initiation	**Roles**	Project Manager, Business Analyst.
	Activities	Conduct project review, document lessons learned, retire project team and resources, archive project documentation.
	Entry Criteria	Completed operation and maintenance phase, customer feedback.
	Exit Criteria	Completed project review, documented lessons learned, archived project documentation.

Table 2.1.1 Steps In Agile Method

2.1.2 Waterfall Model

Steps	Description	
1. Project Initiation	**Roles**	Project Manager, Business Analyst, Software Architect.
	Activities	Define project scope, create the requirements document, establish project timelines, identify project resources, and set project budget.
	Entry Criteria	Approval of the project proposal, defined project scope, budget approval.
	Exit Criteria	Approved requirements document, the project plan that defines project timelines, and team agreement.
2. System Design	**Roles**	Project Manager, Software Architect, Software Designer.
	Activities	Design the system architecture, create a high-level design document, break down the system into smaller components, design interfaces between the system components.
	Entry Criteria	Approved requirements document, project plan, and project budget.
	Exit Criteria	Approved high-level design document, approved interface design document, system architecture document.
3. Implementation	**Roles**	Project Manager, Software Developer, Software Designer.

	Activities	Write code, unit test the code, integrate the system components, and perform system testing.
	Entry Criteria	Approved high-level design document, interface design document, system architecture document.
	Exit Criteria	Completed code, tested software components, system test results, and test plans.
4. Testing	**Roles**	Project Manager, Quality Assurance Engineer, Software Tester.
	Activities	Conduct system and acceptance testing, prepare test cases, validate test results, and report defects.
	Entry Criteria	Completed implementation phase, code review, and system testing.
	Exit Criteria	Approved test results, defect reports, and updated system documentation.
5. Deployment	**Roles**	Project Manager, Business Analyst.
	Activities	Plan the release process, deploy the system, perform user acceptance testing, and train end-users.
	Entry Criteria	Completed testing phase, approved test results, and defect reports.

	Exit Criteria	Approved software release, user documentation, training material, and deployment plan.
6. Project Termination	**Roles**	Project Manager, Business Analyst.
	Activities	Conduct a post-implementation review, document lessons learned, and archive project documentation.
	Entry Criteria	Completed deployment phase, customer feedback, and project review.
	Exit Criteria	Documented lessons learned, archived project documentation.

Table 2.1.2 Steps In Waterfall Method

2.2 Organisational Structure

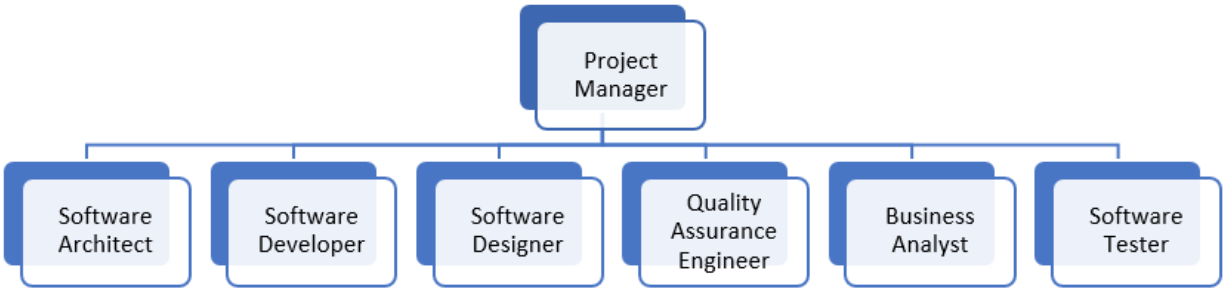


Figure 2.2: Organisation Chart

The internal management structure of a project comprises a team of professionals who hold distinct roles and responsibilities. At the top of the hierarchy is the Project Manager, who oversees the entire project and ensures that it is executed within the predetermined timeline and budget while meeting stakeholder expectations. The Software Architect is responsible for designing the software architecture and developing technical specifications, while the Software Developer writes the code and creates the software application. The Software Designer is responsible for creating the user interface (UI) and user experience (UX) design of the software application. The Quality Assurance Engineer tests the software to ensure that it meets the requirements and specifications, while the Business Analyst identifies business needs and requirements for the software application. The Software Tester executes tests and reports any defects or issues. It is crucial that team members communicate and collaborate effectively with one another to ensure the success of the project.

2.3 Organisational Interfaces

The administrative and managerial interfaces between the project and the primary entities with which it interacts are as follows,

Organisation	Liaison	Company	Contact Information
Mentor	Mr. Md.Shahzar Md. Idris	Head of Digital and Energy Division from PJBUMIHES Sdn.Bhd.	+60 18-779 0400
Lecturer	Dr. Salfarina Binti Abdullah	Universiti Putra Malaysia	+60 12-363 2118
Developer / Tester	1. Izzat Hakimi Bin Ikmal Hisham 2. Sofea Ezzaty Binti Mohd Khusri 3. Noor Syahirah Binti Mohd Sabri 4. Danish Al-Muhaimin Bin Mustaffa 5. Muhammad Rafiq Bin Abd Rashid 6. Julia Nurfadhilah Binti Mohamad Fauzi 7. Muhammad Harith Thaqif Bin Mohd Hairuddin	Universiti Putra Malaysia	1. +60 11-2312 4420 2. +60 11-2878 2791 3. +60 11-2076 7212 4. +60 13-589 2279 5. +60 11-5649 2770 6. +60 11-5165 5810 7. +60 17-241 5330

Table 2.3. Organisational Interfaces

2.4 Project Responsibilities

Each team member is allocated a role, as shown in Table 2.4.1 below.

Role	Description	Person
Project Manager	● Planning: defining project goals, creating a project plan, identifying risks, and determining resource requirements. ● Ensuring effective communication among team members and stakeholders. ● Risk Management: identifying, assessing, and managing project risks. ● Managing project resources (including personnel, equipment, and materials) to ensure the project is completed on time and within budget.	Muhammad Rafiq Bin Abd Rashid
Solution Architect	● Design: creating a detailed design of the software system that meets the requirements. ● System Architecture: creating an overall architecture for the software system. ● Selecting the appropriate technologies to be used in the software system.	Sofea Ezzaty binti Mohd Khusri
Back-end System Developer	● Development: writing the code for the software system. ● Creates APIs to enable communication between different components of a web application.	Danish Al-Muhaimin Bin Mustaffa
Front-end (CGUI) developer	● User Interface Design: designing the user interface of the software system. ● User Experience Design: designing the overall user experience of the software system.	Julia Nurfadhilah Binti Mohamad Fauzi
QA & QC System Engineer	● Testing: testing the software system to ensure that it meets the requirements and is of high quality. ● Defect Management: tracking and managing defects found during testing.	Muhammad Harith Thaqif Bin Mohd Hairuddin
Commercial Requirement Inspector	● Requirements Gathering: understanding the needs of the stakeholders and translating them into software requirements. ● Requirements Management: managing software requirements to ensure they are met throughout the project.	Noor Syahirah Binti Mohd Sabri
System Functional Examiner	● Test Planning: planning the testing activities and creating test cases. ● Test Execution: executing test cases and reporting defects found during testing.	Izzat Hakimi Bin Ikmal Hisham

Table 2.4.1 : Project Responsibility

3. Managerial Process

3.1 Management Objectives and Priorities

Our philosophy for managing this project is to ensure that we deliver a high-quality, reliable and scalable LoRa-based aquatank water acidity monitoring and alert system that meets the needs of our clients. We believe in a collaborative approach that involves regular communication with the stakeholders and the project team to ensure that everyone is on the same page.

Goals:

1. Develop a functional and user-friendly software solution for monitoring the acidity level of aquatank water and creating analytics based on monthly and yearly reading.
2. Ensure that the system is reliable and can provide accurate readings even in challenging environmental conditions.
3. Design the system with scalability in mind, to enable easy expansion or customization in the future.

Priorities:

1. Quality: To apply reliable and scalable solutions to meet stakeholders' needs.
2. Communication: To frequently respond and update the stakeholders with current development progress plus highlighting issues in a timely manner while synchronising with all members.
3. Time: To meet all tasks as per scheduled without sacrificing quality or functionality.

Flexibility Matrix:

To communicate the dimensions of the project that are fixed, constrained, and flexible, we are using a flexibility matrix, as shown below:

Dimension	Fixed	Constrained	Flexible
Scope (Functionality)			X
Schedule		X	
Quality			X

Resources (Data)			X
Features (Characteristics, Performance, Portability and Functionality)		X	

Table 3.1: Flexibility Matrix

Qualifiers:

- The changes of functionality can be made to it without causing significant delays or disruptions to other project activities.
- The activity of manoeuvring or adjusting the schedule without negatively impacting the project's progress or completion date.
- The quality requirements are not particularly strict because there is some room for negotiation or trade-offs in terms of quality versus other project constraints, which are time and cost.
- A particular resource, such as data or information, can be changed or adjusted without significant impact on other tasks or the overall project schedule. This is because the data or information requirements are not particularly strict. However, the project team shall not ignore data quality or accuracy.
- Features are classified as constrained due to its limitation by using XAMPP as the development platform. XAMPP is unable to run on other platforms such as Apple IOs, android and others.

3.2 Assumptions, Dependencies, and Constraints

As the team begins the project to develop a software solution for monitoring the acidity level of the water and generating data analytics, it is important to consider the assumptions, dependencies, and constraints that may impact the work to deliver a high-quality solution by using accurate data that meets the stakeholder's needs.

3.2.1 Assumptions

1. The data provided by the stakeholder is accurate and complete.
2. The data is in a structured format that can be easily analysed.

3. The team has access to the required software tools and resources necessary for data analysis and dashboard creation.
4. The selected mentor shall provide guidance and support to ensure the intended project is completed successfully.

3.2.2 External events dependencies

1. The project is dependent on the timely delivery of the data by the stakeholder.
2. The team needs to coordinate with the stakeholder to set a mutually agreed-upon date for the data handover meeting.

3.2.3 Constraints

1. The team has a limited budget for the development and implementation of the software solution.
2. The project has a specific timeline that must be met, and the team needs to deliver the dashboard and analytics within the specified schedule.
3. The project team needs to maintain effective communication with the mentor throughout the project.

3.3 Risk Management

The following measures will be taken to identify, analyse, and manage risk factors related with the project:

1. Identify potential risks: The project team will identify potential risks through brainstorming sessions, data analysis, expert opinions, and other sources.
2. Analyse the risks: Analyse the risks in terms of likelihood and potential impact using a 3x3 risk matrix or other tools.
3. Manage risks: Develop a risk management plan that outlines strategies for mitigating or reducing the risks.

3.3.1 Risk Tracking and Mitigation Plan

1. Risk Register: Keep a risk register in which all identified hazards will be documented, including their likelihood of occurrence, potential impact, and mitigation techniques.

2. Risk Analysis: Undertake periodical risk assessments to identify whether the identified hazards' probability or impact has changed.
3. Contingency Planning: Develop contingency plans for each identified risk, outlining the steps that need to be taken in case the risk materialises.
4. Risk Reporting: Report the status of the identified risks and any updates to the risk register to the project stakeholders regularly.
5. Risk Monitoring: Throughout the project's life cycle, monitor regularly the risks to ensure that the mitigation techniques are effective and to identify any new risks that may surface.

3.3.2 Risk Factors and Method

The following risk factors have been identified for the project:

1. Contractual risks: Risks related to meeting the terms of the contract with the client, such as delays or failure to meet requirements.
2. Complexity of the product: Risks related to the size and complexity of the system, such as difficulties in integration or quality control.
3. User acceptance risks: Risks related to whether the customer will accept the product, such as failure to meet expectations or lack of interest.

To manage these risk factors, the project team will implement the following methods:

1. Developing a clear and detailed contract with the client that includes specific milestones, deadlines, and deliverables to manage contractual risks.
2. Breaking the project down into manageable tasks and establishing clear roles and responsibilities for each team member to manage risks due to the size and complexity of the product.
3. Conducting regular user testing and soliciting feedback from stakeholders throughout the project lifecycle to manage risks in achieving customer acceptance of the product.

The project team will be better positioned to produce a successful LoRa Based AquaTank Water Acidity Monitoring and Alert System project that fulfils the expectations of the client and achieves the project's objectives if these risk factors are adequately managed.

3.4 Monitoring and Controlling Mechanisms

The reporting mechanisms, report formats, review and audit mechanisms, and other tools and techniques to be used in monitoring and controlling adherence to the Software Project Management Plan (SPMP) for Lora software will vary depending on the specific needs and requirements of the project. These are the mechanisms and tools include:

1. Reporting mechanisms: Progress reports should be generated periodically to provide updates on the status of the project. These reports should include information on the current status of work packages, issues, risks, and changes. Reporting should occur at regular intervals and should be sent to stakeholders, including the project team, management, and clients.
2. Report formats: Reports should be clear, concise, and consistent to enable stakeholders to understand the status of the project easily. Some common formats for reports include tables, graphs, charts, and dashboards. The format should be tailored to the specific needs of the stakeholders.
3. Review and audit mechanisms: Periodic reviews and audits should be conducted to ensure that the project is adhering to the SPMP. Reviews and audits should be conducted by qualified personnel, and findings should be documented and communicated to the relevant stakeholders. Corrective actions should be taken to address any deficiencies identified during the review and audit process.
4. Work package monitoring mechanisms: Monitoring and controlling mechanisms for work packages include measuring progress against planned schedules, tracking resource utilisation, identifying and resolving issues, monitoring risks, and ensuring that quality standards are met. This monitoring should be conducted at the level of individual work packages.
5. Quality assurance: Quality assurance mechanisms should be put in place to ensure that the project's deliverables meet the required standards. Quality reviews, inspections, and testing should be conducted to verify that the deliverables meet the quality requirements.
6. Configuration management: Configuration management mechanisms should be implemented to track changes to the project's configuration items. This includes identifying, documenting, controlling, and reporting changes to the software, hardware, and documentation.

7. Documentation: Documentation and training mechanisms should be established to ensure that the project's stakeholders are aware of the project's requirements, procedures, and standards. This includes developing and maintaining project documentation, providing training to the project team and stakeholders, and conducting knowledge transfer sessions.

Overall, a comprehensive monitoring and controlling plan should be developed that includes all the above mechanisms and tools, tailored to the specific needs of the project. The plan should be regularly reviewed and updated to ensure that it remains relevant and effective throughout the project lifecycle.

3.5 Staffing Approach

A team with the necessary abilities and knowledge is crucial for the effective completion of a project. Depending on the size and complexity of the project, several types of skills are needed and it generally include:

- Problem-Solving Skills: As projects inevitably encounter unexpected issues and problems, it's important to have team members with strong problem-solving skills to develop solutions and make informed decisions.

- Technical Skills: These are specific skills required for the project, such as software development, data analysis, or engineering.

- Communication Skills: Communication skills are vital for conveying information, instructions, and updates to team members, stakeholders, and customers.

- Teamwork Skills: Effective teamwork skills such as collaboration, conflict resolution, and leadership are necessary for successful project completion.

- Management Skills: Project managers should have skills in planning, scheduling, budgeting, risk management, and communication.

Necessary education, training, and certifications should also be taken into account when recruiting competent team members for a project.

Training is essential for the project team to enhance their skills and knowledge to effectively plan and manage projects. Project management courses, technical training, and soft skill training such as leadership and communication may be a part of this training. These trainings can be provided in-house, by external consultants, or through online resources.

4. Technical Process

4.1 Methods, Tools, and Techniques

4.1.1 Methods

Project planning methods are important as it helps teams to organise and manage projects effectively. Project management methodology also provides a structured approach to manage projects, which can help to ensure that the project is completed successfully. On time, within budget, and satisfies all the quality standards. There are several methods that will be used in this project.

1. Agile methodology

Agile is a project management methodology that emphasises the importance of collaboration, flexibility, and adaptability. Teams can react to changes in requirements and deliverables throughout the project lifecycle due to the iterative and incremental process.

As this methodology emphasises collaboration and communication, it can lead to better outcomes and more satisfied stakeholders.

2. Waterfall methodology

The waterfall methodology is a traditional project management approach that involves a linear and sequential process for planning and executing a project. This approach is characterized by a series of distinct phases, each with its own specific set of deliverables and objectives, that must be completed before progressing to the next phase.

Waterfall is a useful approach for managing projects where the scope and requirements are known upfront, and the end goal is clear. However, it can be less flexible than agile approaches, making it less suitable for projects where there is uncertainty or change.

4.1.2 Tools

The use of project management software is crucial in modern project management. There are several tools that can be used to assist in task management, project planning and scheduling, and communication.

1. Jira Software

   Jira is a project management tool that is primarily used for issue tracking and agile project management. It has a range of features that make it easy for teams to collaborate, including threads, real-time updates, and integration with other tools such as Slack and Confluence. Jira can also be customised to fit the needs of different teams and projects with users able to create custom fields, workflows, and labels to organise their work and track progress. It also has a range of built-in reports that allow users to track progress and identify trends.

2. Asana

   Asana is a project management software which allows teams to organise, collaborate, plan and execute tasks. It is a web-based task management and collaboration tool which eliminates the need for any redundant emails and brings all tasks together. Teams can use Asana to break down large work into manageable tasks. It is a comprehensive work management tool that allows teams to track project and task progress, share files, comments, and notes, and keep track of deadlines.

4.1.3 Techniques

The usage of Project Management techniques can be utilised to keep track of deadline compliance, work appraisal, and task completion. These techniques are also employed to effectively portray projects visually.There are several techniques that we used to effectively plan, execute, and control projects.

1. Gantt chart

   Gantt charts are one of the most crucial project management techniques. Gantt charts can be used to assist with project planning, project scheduling, and project tracking across the many phases of your projects. Users can create milestones, modify timetables, link task dependencies, and schedule the work of the entire team.

Team members can see the start and end dates for the project, individual project tasks, when they should begin and end, which team members are working on which tasks, how long each task takes, milestones, and task dependencies with a Gantt chart.

2. Work Breakdown Structure (WBS)

   Work Breakdown Structure (WBS) is a project management strategy that entails splitting a huge project into smaller, more manageable jobs so that they may be better planned, scheduled, and monitored.

   Every task in the project is represented graphically by the WBS. The completed project is displayed at the top of the page, with a line leading to a box (or boxes) that symbolise the larger tasks that led up to it. Then, lines that pass beneath each of these boxes tie it to lesser jobs.

3. Critical Path Method (CPM)

   Critical Path Method (CPM) is used to find the critical path—the sequence of actions that must be completed on time in order to achieve the project deadline—and the longest sequence of tasks in a project. The tasks that make up the critical path are referred to as critical tasks and are given top priority in project schedules. To ensure that the project is finished on time, the CPM assists project managers in identifying potential delays, prioritising activities, and effectively allocating resources.

4.2 Software Documentation

To ensure that the implementation of the software satisfies the requirements, the following documentation is required as a minimum:

1. Software Requirements Specification (SRS)
2. Software Design Description (SDD)
3. Software Test Plan

4.2.1 Software Requirements Specification (SRS)

The SRS describes each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement is

defined such that its achievement is capable of being objectively verified and validated by a prescribed method, for example, inspection, analysis, demonstration, or test.

4.2.2 Software Design Description (SDD)

The Software Design Document (SDD) is a document that outlines the major components of the software solution's design for the LoRa based AquaTank Water Acidity Monitoring software. The SDD includes details about the different components, modules, and layers that make up the system's architecture.

One of the key components of the SDD is the data model. This section of the document describes the different entities and their relationships that the software uses to store and manage data. It also describes the database design including the different tables, columns, and indexes that make up the database.

Another important component of the SDD is the user interface design. This section describes the different screens, buttons, and menus that make up the user interface of the software. It also includes details about how users interact with the software and how the software responds to user input.

The internal interfaces section of the SDD describes the different modules that make up the software solution and how they interact with each other. This section helps ensure that the different components of the software work together seamlessly and efficiently.

Overall, the SDD provides a comprehensive guide for the development team to build a robust, scalable, and reliable software solution for monitoring the acidity level of aquatank water using LoRa technology.

4.2.3 Software Test Plan

To ensure that the LoRa Based AquaTank Water Acidity Monitoring and Alert System is thoroughly tested and functioning correctly, the system requires an effective test plan. The following test plans are what the team agreed on.

1. Unit Testing

Unit testing is performed to the system with the objective of testing the functionality of individual components such as the LoRa modules and pH sensor. To verify the accuracy of the pH sensor readings, the team will compare the data taken from the pH sensor with the existing data that has been collected. Other than that, unit testing also tests the alert system by simulating a drop in pH value and verifying that the alert is triggered. The testing includes test scripts and test procedures.

2. Integration Testing

Integration testing ensures that all the components in the system are communicating with each other as expected, especially between the database and the user interface. The pH value recorded in the database must be displayed correctly on the report in order to ensure the integrity of the data. The test also must verify that the alert system is triggered when the pH value drops below the threshold value.

3. Usability Testing

A usability test is conducted to evaluate the system's ease of use and functionality. It is also to gather feedback from users and use it to improve the system's usability and performance. The system should also fulfil the requirements set by stakeholders.

4.3 User Documentation

The team has agreed to plan and develop comprehensive user documentation for the LoRa based aquatank water acidity monitoring. The user documentation will include online help, network accessible files, paper documentation, and support facilities to assist users in understanding and using the system. The following steps will be taken to plan and develop the user documentation:

1. Determine the user documentation requirements: Identify the target audience for the user documentation and their needs. Determine the level of detail required, the types of information needed, and the format of the documentation.

2. Develop a plan: Create a plan for developing the user documentation, including the scope, schedule, and budget. Assign responsibilities to team members for creating and reviewing the documentation.
3. Create content: Develop the content for the user documentation, which could include online help, network accessible files, paper documentation, and support facilities. Ensure that the content is accurate, clear, and easy to understand.
4. Design and format the documentation: Create a consistent and professional look and feel for the user documentation, including formatting, design, and layout.
5. Review and revise: Review and revise the user documentation to ensure that it is accurate, complete, and up-to-date. Incorporate feedback from users and stakeholders to improve the documentation.
6. Publish: Publish the user documentation and make it accessible to users. Ensure that users are aware of the availability of the documentation and provide support facilities for any questions or issues.
7. Maintain and update: Maintain the user documentation, updating it as necessary to reflect changes to the software or the user's needs. Ensure that the documentation remains accurate, complete, and up-to-date.

4.4 Project Support Functions

The successful completion of the LoRa Based AquaTank Water Acidity Monitoring and Alert System project requires the effective planning and execution of supporting functions. This section outlines the plans for the project support functions, including configuration management, software quality assurance, and verification and validation.

1) Configuration Management Plan:

The configuration management plan includes a system for tracking and managing changes to the software and hardware components of the system. This system also includes guidelines and procedures for managing the configuration of the system and ensuring that all changes are properly documented and approved. Additionally, a schedule for configuration audits is included to ensure that the system is operating according to specifications.

- Responsibilities: The project manager is responsible for overseeing the configuration management process, and each team member is responsible for adhering to the guidelines and procedures established in the plan.
- Resource requirements: The configuration management system will be implemented using software tools such as Git and GitHub, and will require adequate storage capacity for version control of all software and hardware components.
- Schedule: Configuration audits will be conducted weekly, with a more thorough audit at the end of each development phase.

2) Software Quality Assurance Plan:

The software quality assurance plan outlines procedures for testing and validating the system at various stages of development. This plan includes guidelines for reviewing code and ensuring that it meets the project's quality standards. Also, a schedule for conducting regular quality audits is included to ensure that the system is meeting its requirements.

- Responsibilities: The software quality assurance (SQA) team, led by the SQA manager, is responsible for planning, executing, and reporting on all testing activities. The development team is responsible for ensuring that their code meets the project's quality standards.
- Resource requirements: The SQA team will require access to test environments, hardware, and software tools for testing and validating the system. The development team will require access to software tools for code review and quality assurance.
- Schedule: Testing activities will be conducted in each development phase, with a final quality audit conducted prior to release.

3) Verification and Validation Plan:

The verification and validation plan outlines procedures for testing the system to ensure that it meets its specified requirements. This plan also includes guidelines for conducting user testing and soliciting feedback from stakeholders to ensure that the system meets their needs.

Additionally, a schedule for conducting regular validation checks is included to ensure that the system is functioning properly.

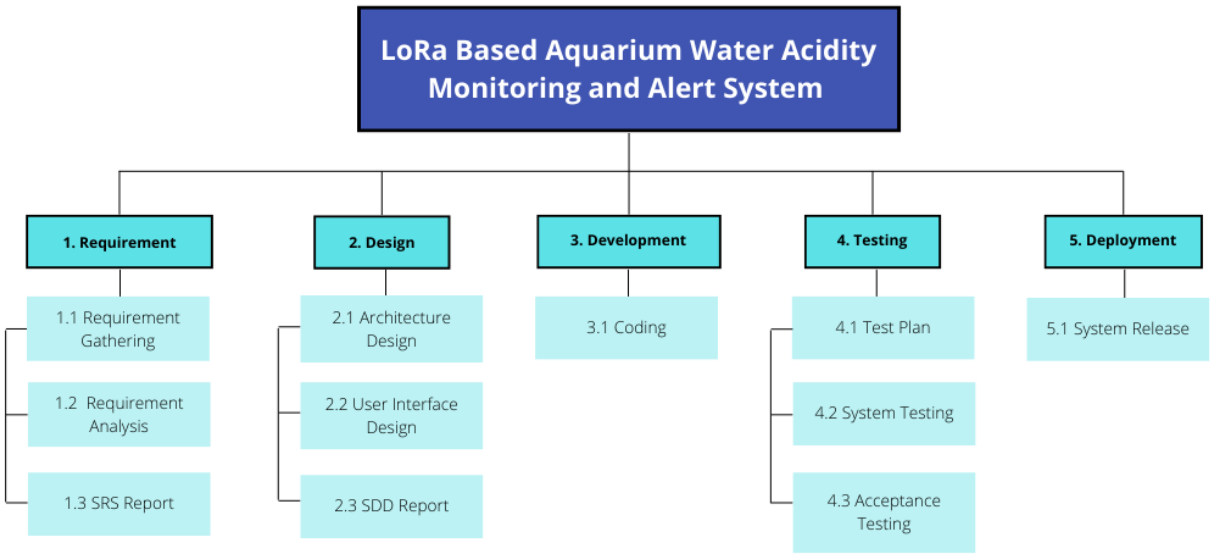
- Responsibilities: The testing team, led by the testing manager, is responsible for executing the verification and validation activities. The development team is responsible for ensuring that the system meets its specified requirements.
- Resource requirements: The testing team will require access to test environments, hardware, and software tools for testing and validating the system. The development team will require access to software tools for requirement tracking and change management.
- Schedule: Verification and validation activities will be conducted in each development phase, with a final acceptance test conducted prior to release.

5. Work Packages, Schedule, and Budget

5.1 Work Packages

The work packages that include a set of tasks that are necessary to complete and contribute to the overall success of the project are as below.

Work Breakdown Structure (WBS)



5.2 Dependencies

5.2.1 Task dependencies

Task ID	Task Detail	Duration (week)	Dependencies
T1	Requirement Gathering	2	-
T2	Requirement Analysis	1	T1
T3	SRS Report	2	T2
T4	SDD Report	3	T2,T3

T5	Architecture Design	1	T4
T6	User Interface Design	2	T4
T7	Coding	3	T5,T6
T8	Test Plan	1	T7
T9	System Testing	1	T7
T10	Acceptance Testing	1	T7
T11	System Release	1	T8,T9,T10

Table 5.2.1 Dependencies between tasks

5.2.2 External dependencies

External Event ID	Detail	Dependencies
EE1	The project is dependent on the timely delivery of the data by the stakeholder.	T1, T2
EE2	The team needs to coordinate with the stakeholder to set a mutually agreed-upon date for the data handover meeting.	T1, T10, T11

Table 5.2.2 Dependencies between tasks and external events

5.2.2 Gantt Chart

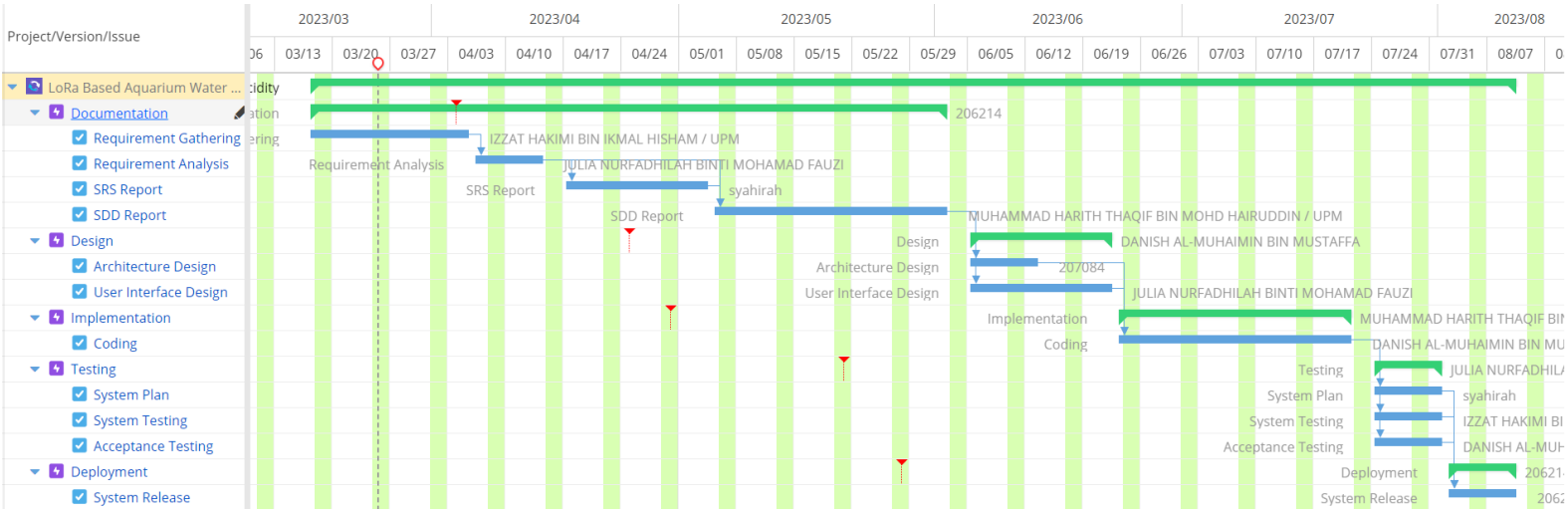


Figure 5.2.2 Gantt Chart

5.2.3 Critical Path Method

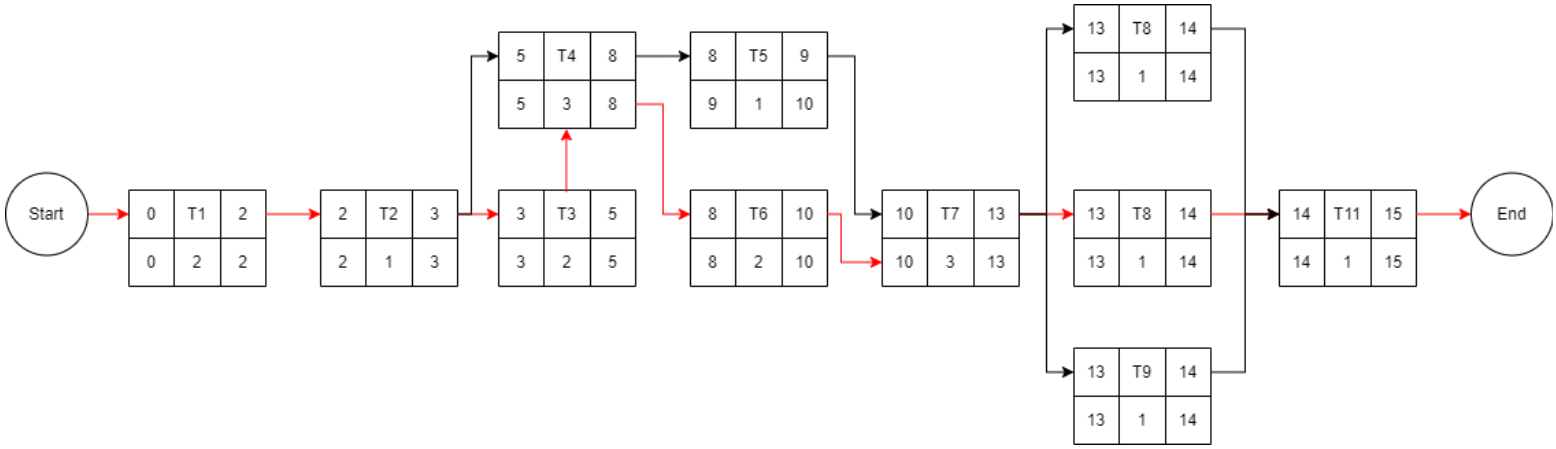


Figure 5.2.3 Critical Path Method

5.3 Resource Requirements

1) Personnel: Project Manager (1 person), Software Architect (1 person), Software Developer (1 person), Software Designer (1 person), Software Quality Engineer (1 person), Business Analyst (1 person), and Software Tester (1 person).
2) Computer Time: The software will be deployed on a server or cloud platform, so the required computer time will depend on the hosting platform and the level of traffic the software generates.

3) Support Software:
    a) XAMPP: Open source web server solution, which can be used for local testing and development
    b) MySQL: Open source relational database management system for storing data
    c) PHP: Server-side scripting language for dynamic web applications
    d) Flutter: Open source mobile application development framework
4) Computer Hardware:
    a) Computers/laptops for each team member
5) Office and Laboratory Facilities: Office space for the team members to work together and collaborate on the project

5.4 Budget and Resource Allocation

The allocation of the budget will be RM0.00. Most tools will be using free-tier versions.In cases of needed budget, the development team will be asking compensation from the customer.

5.5 Schedule

Task ID	Task Detail	Start Date	Duration	Finish Date
T1	Requirement Gathering	17/3/2023	2	5/4/2023
T2	Requirement Analysis	6/4/2023	1	14/4/2023
T3	SRS Report	17/4/2023	2	4/5/2023
T4	SDD Report	17/4/2023	3	15/4/2023
T5	Architecture Design	16/5/2023	1	24/5/2023
T6	User Interface Design	16/5/2023	2	2/6/2023
T7	Coding	6/5/2023	3	3/7/2023
T8	Test Plan	4/7/2023	1	12/7/2023
T9	System Testing	4/7/2023	1	12/7/2023
T10	Acceptance Testing	4/7/2023	1	12/7/2023
T11	System Release	12/7/2023	1	21/7/2023

Table 5.5: Schedule

6. Additional Components

6.1 Index

key terms: management, schedule, acidity, resources, requirements, quality assurance, procedures, plan, stakeholder

acronyms:

1. WBS - Work Breakdown Structure
2. SRS - Software Requirements Specification
3. SDD - Software Design Description

6.2 Appendices

The project will be done based on reading from sensors in a Neptune Aquaculture Farm. The description of an aquafarm are as below:

Neptune NRCP (or “Neptune”) is a sector-focused private equity firm dedicated to investing in small to medium-sized businesses operating within the global aquaculture value chain, with a specific focus on Europe, the Americas and Australasia. This includes businesses operating in and across the global fish farming, wild-catch, supply chain, services & inputs, technology & equipment, processing & value adding sectors.

Figure 6.2: Aquafarm