Course Submission Cover Sheet

Module: CS6003ES Advance Software Engineering

Assignment no: 001

Weighting: 30%

Deadline: TBC

Module Leader: **Student ID:** KAN00257474

*Please note that there are specific regulations concerning **the use of AI and Academic Misconduct**. Below are extracts from these regulations. By signing, you acknowledge that you have read and understood these extracts.*

(signature:) Ravindu Date: 27/03/2025

This header sheet should be attached to the work you submit.

Academic Integrity means being honest in your academic work and your studies and making sure that you acknowledge the work of others and giving credit where you have used other people's ideas as part of presenting your arguments. Your assessment submissions must therefore always be entirely your own work, based on your own learning and appropriately referenced including how you have used Generative AI. The University regards the use of Generative AI applications by students to deceive to gain unfair advantage as **academic misconduct**. This usage includes:

- **Plagiarism**, where AI tools are used to generate output and ideas that are presented or submitted as if they were the student's own work, without proper citation or references.

- Where a complete assignment is created using Generative AI and represented as a student's own work, this will be regarded as contract cheating in the same way as commissioning an 'Essay Mill' or other third party to complete your work. Further information can be found on : Guidance on the use of Artificial Intelligence.

Academic misconduct: The University takes academic misconduct very seriously and seeks at all times to rigorously protect its academic standards. Plagiarism, collusion and other forms of cheating constitute academic misconduct, for which there is an explicit range of graduated penalties depending on the particular type of academic misconduct. The penalties that can be applied if academic misconduct is substantiated range from a reprimand to expulsion in very serious cases and for repeated instances of misconduct. You are also responsible for ensuring that all work submitted is your own and that it is appropriately referenced. The University does not tolerate cheating of any kind. You are strongly advised to familiarise yourself with the Academic Misconduct Policy and Procedure (Academic Misconduct), which list a range of categories of academic misconduct and associated penalties, covering instances of academic misconduct (plagiarism, collusion, exam cheating).

Declaration

Name hereby declare that the project titled "Online Gas Requesting and Delivering System for GasByGas" is my original work and has been completed as part of my academic requirements at ESOFT Metro Campus.

Name: Muthumuni Ravindu Gayal Lakshan De Silva

ID: KAN00257474

Date: 27/03/2025

Acknowledgement

I would like to express my sincere gratitude to **Mrs.Yamuna Nawarathne**, my lecturer for **BEng (Hons) Software Engineering**, for their guidance, support, and valuable insights throughout this assignment. Their encouragement and expertise have been instrumental in completing this work successfully.

Furthermore, I would like to thank my family and team members for their continuous support and motivation throughout this process

Name: Muthumuni Ravindu Gayal Lakshan De Silva

Contents

List of Figure

List of Tables

Introduction

The objective of this project is to develop a comprehensive Online Gas Distribution System for GasByGas (Pvt) Ltd., a leading company specializing in the distribution of liquefied petroleum gas (LPG). This system will allow customers to request and schedule gas deliveries online, streamlining the ordering process while improving operational efficiency. The development process will be guided by an agile methodology, ensuring that the system can evolve based on user feedback and changing business needs.

This report outlines the detailed project plan for the software development of the Online Gas Distribution System. It will cover all necessary aspects of the project, including the team structure, work breakdown, timeline, and cost estimates. The project will follow industry standards, such as ISO 12207-2008 for software lifecycle processes, and adhere to IEEE recommendations for project planning and management.

In this plan, I will take on the role of project manager, overseeing a cross-functional team consisting of both full-time and part-time members. The team will be tasked with managing key processes such as project planning, scope management, time management, cost management, and human resource management. The project will be organized into sprint cycles, with a focus on delivering incremental improvements in the system.

This report will also address the organizational structure of the team, the allocation of tasks to specific team members, and a detailed budget summary. Furthermore, a Work Breakdown Structure (WBS) and Gantt chart will be provided to map out the schedule and deliverables for the project. Finally, a monitoring and maintenance plan will be established to ensure the successful completion and sustainability of the system post-launch.

This structured approach ensures that the development of the Online Gas Distribution System is efficient, on time, and within budget while also meeting the expectations of GasByGas (Pvt) Ltd.

Chapter 01: Generic Plan Information

This chapter gives basic information regarding the project plan, such as terminology and documentation requirements.

1.1 Generic Plan Information

System details:

- System version: 1.0
- Issue Organization: GasByGas LP Gas Distribution
- Software Name: GasByGas Online Gas Requesting & Delivery System
- Date of System Issue:08.02.2024
- Current Situation: The system has been fully developed, the system has been fully developed, tested, and successfully deployed for use

1.1.1 Document Standard

This document adheres to the IEEE 12207.1 and ISO 12207-2008, which offer a methodical approach to software development lifecycle management. By guaranteeing a methodical approach to every stage of development, these standards foster uniformity, traceability, and quality in all project duties. These standards' recommendations ensure a dependable and effective project management process by assisting in aligning the project with industry best practices. Every stage of software development, from gathering requirements to deployment and maintenance, must be carefully planned and carried out, according to the IEEE 12207.1 standard, which stresses a methodical and disciplined approach. The acquisition, supply, development, operation, and maintenance lifecycle procedures are all clearly structured by it, ensuring that every action is accurately recorded and verifiable. This is enhanced by the ISO 12207-2008 standard, which provides thorough guidance for software lifecycle procedures, such as configuration management, project management, and quality assurance. It guarantees that the project complies with global best practices, encouraging cooperation, openness, and responsibility from all parties involved.

The report follows the Harvard Anglia referencing style throughout and is formatted in Times New Roman font. Consistent numbering is applied for clarity, while headings are bolded to enhance readability. Additionally, the line spacing is set to 1.5 to ensure a professional and well-presented document

Levels	Front Name	Front Size	Front Style	Front color
Heading	Times New Roman	12	Bold	Black
Sub Heading 1	Times New Roman	12	Bold	Black
Sub Heading 2	Times New Roman	12	Bold	Black
Caption	Times New Roman	12	Italic	Black
Paragraph	Times New Roman	12	Normal	Black

Table 1_Font Standards

IEEE 12207.1

A standardised framework for overseeing software development processes is offered by IEEE 12207.1. It outlines a number of stages, assignments, and activities that must be completed during the Software Development Life Cycle (SDLC). The GasByGas Online Gas Distribution System was developed using IEEE 12207.1, which guarantees that each step—from planning to deployment—is precise, standardised, and traceable. This standard place a strong emphasis on crucial elements that support the overall dependability and success of the software system, like quality management, risk assessment, and project monitoring.

ISO 12207-2008

IEEE 12207.1 is enhanced by ISO 12207-2008, which offers thorough guidelines for the software lifecycle that address design, development, testing, deployment, and maintenance. The GasByGas system complies with this criterion, guaranteeing responsibility, process enhancement, and conformity to international best practices while encouraging teamwork for effective implementation.

1.1.2 Acronyms

The following acronyms are used throughout this document

Acronym	Definition
LP Gas	Liquefied Petroleum Gas
WBS	Work Breakdown Structure
PM	Project Management
API	Application Programming Interface
NIC	National Identity Card

UI	User Interface, User experience
SRS	Software Requirement Specification
UML	Unified Modelling Language
DBMS	Database Management System
ERP	Enterprise Resource Planning

Table 2_Acronyms

Chapter 02: Project Overview

This chapter provides an overview of the GasByGas project, including its goals, assumptions, limitations, scope, and main deliverables. It provides a crucial framework for comprehending the project's structure and elucidating the anticipated results.

2.1 System Overview

The GasByGas system is an online platform designed to streamline the gas cylinder distribution process in Sri Lanka, offering a more efficient and user-friendly solution for both residential and industrial consumers. It allows consumers to easily request gas, receive a unique token for tracking, and get notifications about delivery statuses via email. The system automatically calculates delivery timelines and notifies users of any delays.

For outlet managers, the platform provides a comprehensive interface to handle requests, assign deliveries, and manage the reallocation of unfulfilled orders. It also includes a special interface for managing industrial and business gas requests, with registration and credential validation required for businesses. By automating notifications and improving communication, the GasByGas system reduces human error and enhances operational efficiency. It ensures timely deliveries, improves customer experience, and serves both residential and business sectors, making it a vital tool for the future of gas distribution in Sri Lanka.

2.2 Purpose

The GasByGas system's main goal is to modernise and streamline the gas ordering and distribution process by substituting an effective, automated, and user-friendly platform for manual, traditional techniques. A smooth and transparent gas supply chain, improved customer convenience, and streamlined supplier operations are just a few of the main goals that this system seeks to accomplish.

Through the digitisation of the gas ordering process, the system guarantees that clients obtain accurate delivery estimates, minimises delays, and offers real-time availability updates. Automated stock tracking and optimised route planning enhance supplier management, enabling effective gas distribution and reducing delivery interruptions. By including safe payment options, real-time alerts, and a customer service section to effectively address questions and grievances, the system also improves user experience.

The goal of GasByGas is to raise service standards, promote dependability in gas distribution, and provide businesses a competitive edge in the market by providing a cutting-edge, technologically advanced solution to satisfy the rising needs of both suppliers and customers.

2.3 Scope

A comprehensive online infrastructure for handling gas requests, scheduling deliveries and giving customers real-time notifications is what the GasByGas project seeks to provide. Developing an effective, user-friendly system that allows customers to request gas, receive a token for their request, and monitor the status of their delivery is the main goal of the project. The technology will also enable outlet managers to effectively manage gas distribution for business or industrial users, accommodate unmet deliveries, and handle gas demands.

Key functionalities within the scope of this project include

- **Gas Request System**: Customers will be able to use the web site to request LP gas. Every request will be associated with a distinct token that will enable the outlet management and the customer to monitor the request's progress.
- **Delivery Scheduling**: Users will be able to plan delivery times using the system according to the gas supply at their area. Additionally, the site will inform customers of the status of their delivery and provide an approximate arrival date.
- **Notifications**: Automated email notifications will be sent to customers regarding significant occurrences such request confirmation, delivery status updates, and delivery fulfilment. By improving communication between distributors and customers, these alerts will guarantee a smooth transaction.
- **Token Management**: In order to facilitate order tracking, delivery confirmation, and effective process management, each request will produce a distinct token.
- **Industrial and Business Gas Requests**: Businesses and industrial accounts will be able to register on the system and go through a validation process before they may obtain gas cylinders. This functionality guarantees that companies with unique gas requirements are properly handled.
- **Outlet Manager Interface**: Outlet managers will be able to arrange deliveries, handle incoming requests, reallocate unmet orders and supervise gas distribution through an

interface. To ensure on-time delivery and optimise the operational flow, this feature is essential.

2.4 Objective

The specific goals of the project include

Providing a reliable and efficient platform for consumers to request LP gas and track their deliveries

- Customers will be able to request LP gas online with ease thanks to the technology. Additionally, it will give them transparency and lessen delivery time uncertainty by enabling them to monitor the progress of their gas supplies.

Enabling outlet managers to handle request reallocations and ensure on-time deliveries.

- When necessary, outlet managers will be able to oversee and redistribute requests for petrol supply. Even if there are alterations or problems with the initial timetable, this will assist guarantee that customers receive their gas supplies on time.

Implementing a robust notification system (email) for effective communication.

- To notify customers about their gas demand and delivery schedules, the system will have an automated email notification capability. By doing this, communication will be enhanced and clients will be instantly informed of any changes or delivery status.

Offering secure management for industrial/business gas requests and validations

- Large-scale gas requests will be managed and validated securely by the system for business and industrial clients. This guarantees that the proper security and verification procedures are used to suit business-specific demands in order to stop fraud or misuse.

Delivering a system that improves operational efficiency and customer satisfaction.

- The system's automation of the gas request and delivery process will improve overall efficiency, streamline operations, and lower human error. Customers will thus receive quicker, more dependable service, which will increase their level of satisfaction.

2.5 Assumptions and Constraints

2.5.1 Assumptions of the Project

- **Availability of Required Technology:** The system will be dependent on generally accessible technology including email services, databases, and web development frameworks. It is anticipated that the infrastructure and technologies required for the project would be easily available and supported.
- **Stable Internet Access for Consumers and Managers:** For the purpose of interacting with the platform, the system expects that both outlet managers and customers will have dependable internet connectivity. This covers email alerts for communication as well as the online request and monitoring tools.
- **Data Privacy and Security Compliance:** In order to ensure secure management of user data, particularly for business and industrial requirements, the system will comply with privacy and data protection regulations. This presumption covers adherence to pertinent national and international security regulations.
- **User-Friendliness and Accessibility:** To guarantee user-friendliness for customers, including those with little technical expertise, it is expected that the system will be built with an intuitive and user-friendly interface. The platform will also be made to be responsive to mobile devices for ease of use.
- **Testing and Validation of Key Features:** The system will be extensively tested to make sure that all features such as processing gas requests, tracking, and notifications function as planned. The testing stages are anticipated to be finished without major problems, guaranteeing a seamless rollout.
- **Support and Training for Outlet Managers:** The system will be taught to outlet managers, with a focus on handling requests for reallocations and making sure that deliveries are made on time. It is anticipated that adequate support resources, such as manuals and help with troubleshooting, will be offered.

2.5.2 Constraints of the Project

- **Budget Limitations:** The project has limited financial resources allocated for the development, maintenance, and ongoing support of the system. This constraint may

affect the scope of certain features or the inclusion of additional functionalities beyond the core requirements.

- **Timelines:** Strict deadlines are in place for the completion of the system, which will require the development team to adhere to a clear and focused schedule. This constraint may limit the time available for testing, feedback incorporation, or additional refinements that could improve the system's overall quality.
- **Regulatory Requirements:** The system must comply with local data protection, privacy regulations, and industry standards to ensure secure handling of customer information, particularly for business and industrial gas requests. These legal constraints may require These legal constraints may require additional validation processes and documentation for compliance.
- **Stakeholder Involvement:** Active cooperation from key stakeholders, including outlet managers, consumers, and administrative staff, is essential for the project's success. Their timely feedback and input during the design, testing, and implementation phases are critical but may also be impacted by their availability or conflicting priorities.

2.6 Project Deliverables

2.6.1 Project Deliverables to the Client

- **Fully functional online gas requesting and delivery platform:** Customers can easily request gas deliveries, track their orders in real time, and receive estimated delivery times through this user-friendly system.
- **A backend management interface for outlet managers:** An easy-to-use and safe dashboard that gives outlet managers the ability to allocate deliveries, manage inventory, supervise orders, and create reports for improved operational management.
- **Integration with email notification systems:** An automated email system that guarantees smooth contact between customers and outlet managers by sending purchase confirmations, delivery updates, and critical notifications.
- **Documentation for users and administrators:** comprehensive help files and manuals that describe the administrative procedures and system features.
- **Training for outlet managers and other users on how to use the system:** comprehensive training courses that are offered both online and on-site to give employees and outlet managers the skills they need to effectively use the system and respond to client demands.

2.6.2 Project Deliverables on Phases

The project has been broken down into several stages to guarantee the Online Gas Requesting and Delivery Management System is developed in an organized and methodical manner. Specific deliverables that support the system's overall performance and sustainability mark the end of each phase.

Phase 1: Requirement Gathering and Analysis

- **Requirement Gathering Techniques:** Techniques for gathering requirements should be tailored to the specific stakeholder type. While watching current procedures reveals inefficiencies, questionnaires and interviews aid in understanding functional demands. Examining current documentation, such as logs or reports, identifies opportunities for improvement and system constraints.
- **Requirement Analysis:** Finding problems with the current system, such as stock shortages or delivery delays, is the goal of requirement analysis. Clear system objectives, like automating the ordering process, boosting user-dispatch officer communication, and increasing order status transparency, are defined with the use of this research.
- **Feasibility Study:** The viability of the suggested solution is assessed by the feasibility study. It looks at the technical viability of using WordPress, MySQL, and Java Spring Boot to guarantee system scalability and resilience. A user-friendly interface for all parties involved is guaranteed by operational viability. Legal viability guarantees adherence to data protection laws, especially when managing private user information.
- **Requirement Specification Documentation:** The process of producing a comprehensive document that classifies functional and non-functional needs is known as requirement specification documentation. Functional requirements outline what the system must be able to do, such as letting users plan deliveries or request gas. Performance expectations for things like load handling, usability, and security are defined by non-functional criteria.

Phase 2: System Design and Architecture

- **Architecture Design:** The architecture of the system ought to be adaptable and scalable. This include creating the server and database structure, choosing the right technology stack (Java for the backend, WordPress for the frontend, and MySQL for the database), and making plans for future expansion. The best architecture is client-server since it keeps the user interface and backend services apart, which facilitates scalability and maintenance.
- **Database Design:** Managing massive volumes of user and transactional data requires a robust database architecture. Relationships between entities such as users, gas demands, tokens, and outlets will be modelled using entity relationship diagrams, or ERDs. In order to guarantee effective data retrieval and storage, this phase involves creating tables, indexes, and relationships.
- **User Interface (UI) Design:** User adoption depends on the user interface (UI), which needs to be designed simply and intuitively. The user interface will be made to be user-friendly on a variety of devices. Registration forms, gas request forms, and dashboards customized for different jobs will all have design mock-ups made.
- **Data and File Structures:** To effectively store user information, gas requests, tokens, and other relevant data, data structures will be created. For example, order information is stored in the "Gas Request" class, whereas personal information is stored in the "User" class. Documents such as certifications for business clients will be stored in files.

Phase 3: Implementation and Testing

- **Frontend Development:** WordPress will be used to build the frontend, and the WP REST API will be used to connect to the backend. With an emphasis on user experience, key pages such as registration, gas request forms, and dashboards for different users will be created.
- **Backend Development:** The logic for handling gas requests, creating tokens, and planning delivery will be provided by the backend services, which will be constructed with Java Spring Boot. Notification transmission and database interaction will also be managed by the backend.
- **Database Development:** Initial data will be added to the database after it has been developed. Using the finest database techniques, the team will make sure that data is efficiently retrieved and kept securely.

- **Integration:** One important step is to integrate the frontend and backend. In order to provide seamless data flow between the user interface and the backend services, this entails connecting the two. External services for notifications, such as SMS or email, will also be integrated.
- **Testing:** Multiple phases of testing will be carried out, including functional testing for task accuracy, integration testing for frontend-backend compatibility, and unit testing for component functioning. User Acceptance Testing (UAT) will guarantee stakeholder satisfaction, performance testing will evaluate load handling, and security testing will look for vulnerabilities.

2.7 Project Schedule

The GasByGas Online Gas Requesting and Delivery System project schedule describes the intended timeline, including significant checkpoints and the anticipated completion dates for each stage. The following table guarantees a systematic and effective development process by giving a summary of the planned tasks, their start and end dates, and the duration (in days) for each task.

Scheduled Task	Start Date	End Date	Duration (Days)
01. Planning	09-Dec-2024	20-Dec-2024	12
Project Initiation	09-Dec-2024	11-Dec-2024	3
Requirement Gathering	12-Dec-2024	17-Dec-2024	6
Feasibility Study	18-Dec-2024	20-Dec-2024	3
02. Design	23-Dec-2024	06-Jan-2025	6
System Design	23-Dec-2024	27-Dec-2024	3
UI/UX Design	28-Dec-2024	02-Jan-2025	15
Database Design	03-Jan-2025	06-Jan-2025	5
03. Development	07-Jan-2025	03-Feb-2025	6
Frontend Development (UI)	07-Jan-2025	14-Jan-2025	4
Backend Development	15-Jan-2025	28-Jan-2025	28

Integration	29-Jan-2025	03-Feb-2025	8
04. Testing	04-Feb-2025	10-Feb-2025	14
Unit Testing	04-Feb-2025	06-Feb-2025	6
System Testing	07-Feb-2025	10-Feb-2025	7
05. Deployment	11-Feb-2025	14-Feb-2025	3
Deployment & Preparation	11-Feb-2025	14-Feb-2025	4
06. System Launch and Review	17-Feb-2025	21-Feb-2025	4

Table 3_Project Schedule

2.8 Budget Summary

The budget for the GasByGas Online Gas Requesting and Delivery System project is primarily allocated to human resource expenses, as previously outlined. The total project budget amounts to Rs. 80,000,000, exclusively dedicated to team salaries throughout the project duration. Below is a detailed summary of the budget allocation.

Project Role	**Employee Type**	**Hourly Rate (Rs.)**	**Hours Worked**	**Total Amount (Rs.)**
Project Manager (Ravindu Silva)	Full-Time	3,500	500	1,750,000
Senior Developer (Sachini Lunuwila)	Full-Time	3,000	800	2,400,000
UI/UX Designer (Ahinsala Dissanayaka)	Part-Time	2,500	400	1,000,000
Quality assurance tester (Dewumini Dissanayaka)	Full-Time	2,800	600	1,680,000
Business Analyst (Dewumini Dissanayaka)	Full-Time	3,200	450	1,440,000

Junior Developer (Sachini Lunuwila)	Part-Time	2,000	300	600,000
Additional Costs				11,000,000
Total Hours				3,050
Total				Rs. 80,000,000
Duration: 12 Weeks Work Schedule: 8 Hours per day, 6 Days per week				

Table 4_Budget Summary

In order to effectively complete the GasByGas Online Gas Requesting and Delivery System within the budget allotted, the project team consists of a number of important personnel. The full-time project manager, Ravindu De Silva , oversees all project procedures in accordance with applicable standards for a total of Rs. 1,750,000. Her hourly wage is Rs. 3,500. The full-time Senior Developer, Sachini Lunuwila, earns Rs. 3,000 per hour for his work, totalling Rs. 2,400,000. at a total of Rs. 1,000,000, Ahinsala Dissanayaka, the part-time UI/UX Designer, designs user-friendly interfaces at Rs. 2,500 per hour. Software quality is guaranteed by Dewumini Dissanayaka, a full-time quality assurance tester, who charges Rs. 2,800 per hour, or Rs. 1,680,000. The full-time business analyst, Dewumini Dissanayaka, oversees requirements analysis and documentation at an hourly rate of Rs. 3,200, for a total of Rs. 1,440,000. At Rs. 2,000 per hour, or Rs. 600,000, part-time Junior Developer Sachini Lunuwila contributes to the development. While maintaining the project's budget, this well-thought-out distribution guarantees effective resource use and equitable remuneration depending on each member's position and contribution. To finish the project on time, the crew will work for 12 weeks.

Chapter 03: Project Team and Organizational Structure

The project team's composition, responsibilities, and accountability structures are described in this chapter. Throughout the project lifecycle, it makes clear how the project will be managed and what each team member's roles are to guarantee efficient execution and communication.

3.1 Organizational Structure

Team members will report to both the project manager and their respective functional heads under the GasByGas project's matrix organisational structure. This methodology guarantees effective cooperation amongst several fields, including operations, testing, and development. The matrix format was used to allow for resource allocation flexibility and to promote efficient communication among project team members.

Role	Employee Name	Responsibilities
Project Manager	Ravindu De Silva	Oversee the reporting, resource distribution, and project execution.
Senior Developer	Sachini Lunuwila	Oversee development activities, evaluate code, and guide less experienced developers.
UI/UX Designer	Ahinsala Dissanayaka	Create interfaces that are easy to use and enhance user satisfaction.
Quality assurance tester	Dewumini Dissanayaka	Check software for errors, functionality, and performance.
Business Analyst	Dewumini Dissanayaka	Compile requirements, examine business requirements, and record procedures.
Junior Developer	Sachini Lunnuwila	Help with feature implementation, issue fixing, and development activities.

Table 5_Organizational Structure

3.1.1 Selected Organization Structure

Depending on the stage of the project or their area of expertise, team members may report to several managers under the matrix organisational structure, which combines functional and project-based responsibility. Project managers are ultimately in charge of the project's progress in this situation, but functional leaders such as the development or testing leads will also have an impact on their teams to make sure project deliverables adhere to operational and technical requirements.

Because it facilitates cross-functional cooperation and variable resource allocation, the matrix form is advantageous for intricate projects like GasByGas. It promotes effective collaborative decision-making procedures and transparent accountability.

3.2 Project Team

The GasByGas project team consists of professionals with diverse skills and expertise required to ensure the system's successful development, implementation, and maintenance. Each team member will be assigned clear roles and responsibilities to ensure smooth workflow and timely project delivery.

3.2.1 Team Structure Description

To ensure successful project execution, the GasByGas Project is developed with an effective and cooperative team structure. Team members report to the project manager in addition to functional heads under the project's matrix organizational model. Clear communication, adaptability in allocating resources, and productive cooperation amongst various roles are encouraged by this arrangement. Through concerted efforts, each team member contributes specialized skills to their position, helping the project succeed.

Team Member	Background and Expertise	Role in the Project	Percentage of Involvement
Ravindu Silva	Agile approaches and project management.	Oversees risk management, resource allocation, and project execution.	100%
Sachini Lunuwila	Full-stack development.	Oversees development, guarantees system	80%

		functionality, and guides more junior developers.	
Ahinsala Dissanayaka	Prototyping, user research, and UI/UX design.	Creates interfaces that are easy to use and improves the user experience.	100%
Dewumini Dissanayaka	Quality assurance and software testing.	Checks software for errors, functionality, and performance.	50%
Dewumini Dissanayaka	Requirements collection and business analysis.	Evaluates business requirements and guarantees project coherence.	50%
Sachini Lunuwila	Debugging and bug fixing.	Helps with debugging and bug fixing.	20%

Table 6_Team Structure Description

3.2.2 Team Responsibility

The GasByGas Project depends on a well-defined allocation of duties to guarantee seamless cooperation and effective project implementation. Every team member has distinct responsibilities that correspond with their areas of expertise, guaranteeing that the right person is handling every facet of the project. Every team member is essential to accomplishing project objectives, from design and quality assurance to project management and software development.

Team Member	Responsibility
Ravindu Silva	Oversees the scheduling, planning, and implementation of projects. controls risk, distributes resources, and makes ensuring the team works together to accomplish goals and deadlines.
Sachini Lunuwila	Oversees software development, guarantees code quality, handles technical issues, guides less experienced developers, and assists in putting the system's essential features into practice.
Ahinsala Dissanayaka	Carries out user research, builds prototypes, designs and develops user-friendly interfaces, and makes sure that the project's user experience runs well.

Dewumini Dissanayaka	Works closely with the team, finds defects, and tests software to guarantee quality and functioning. In order to match project deliverables with client expectations, it also examines business requirements.

Table 7_Team Responsibility

Chapter 04: Work Breakdown Structure (WBS) and Project Schedule

4.1 Work Breakdown Structure (WBS)

One of the most important tools for project management is the Work Breakdown Structure (WBS), which divides a project into manageable, smaller parts. The WBS is set up according with ISO 12207-2008 for the GasByGas Online Gas Distribution System, guaranteeing adherence to industry best standards for software development. System design, development, testing, deployment, and continuing support are all covered by this systematic technique, which breaks the project down into important stages, subphases, and jobs. The project team can effectively distribute resources, monitor progress, and make sure all-important deliverables are fulfilled without forgetting important tasks.

There are three levels in the GasByGas WBS:

- **Level 1:** Major Project Phases (This top-level framework outlines the project's main phases)
- **Level 2:** Sub-Phases within Each Major Phase (There are subgroups within each phase that concentrate on particular duties.)
- **Level 3:** Tasks Under Each Sub-Phase (Each sub-phase at this level has distinct tasks.)

The three-level WBS guarantees that each project phase is organized and controllable. In accordance with worldwide software development standards, this hierarchical breakdown guarantees high-quality software delivery, improves coordination, and lowers complexity.

4.2 WBS for System

Here’s the Work Breakdown Structure (WBS) for the GasByGas project, organized by phases with a clear breakdown of tasks.

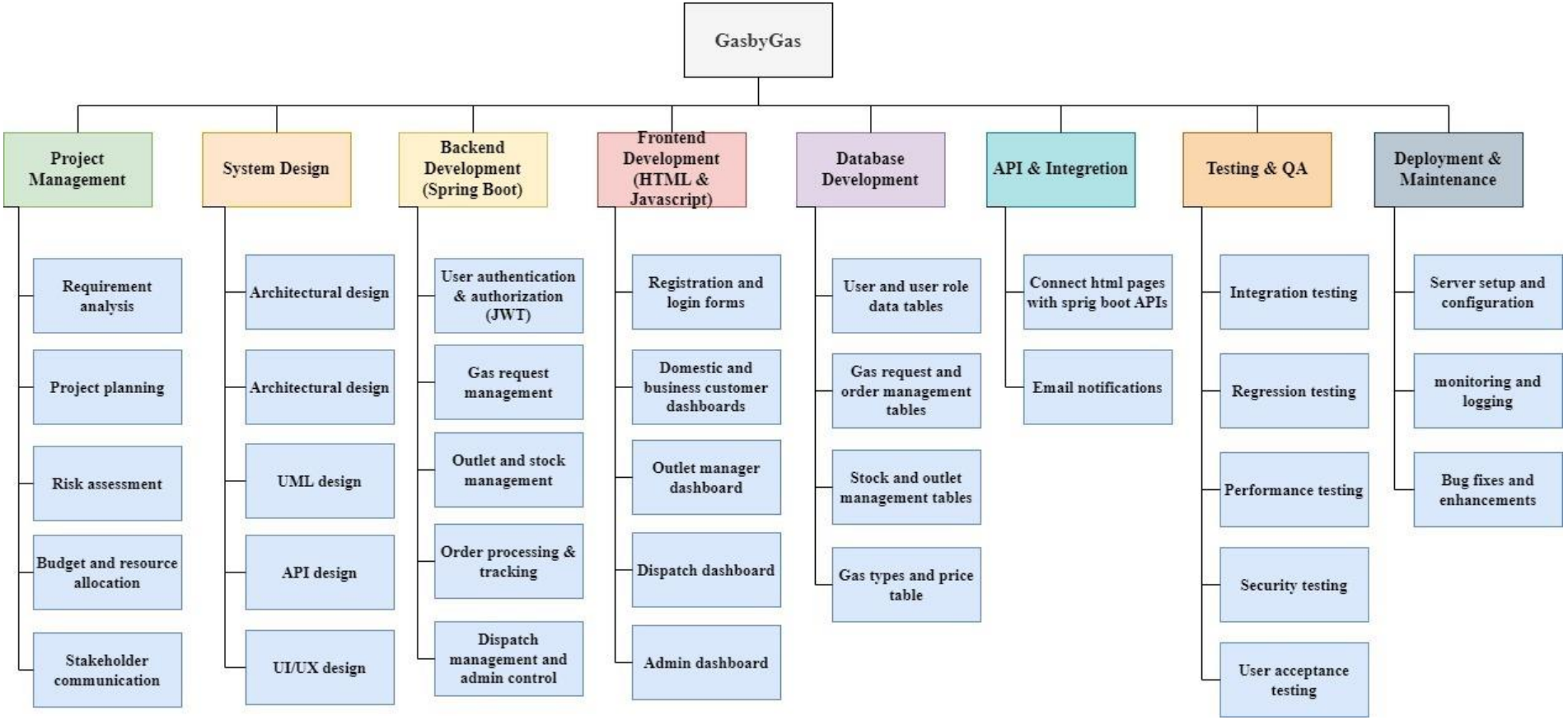

Table 8_WBS for System

4.3 Rationale for the WBS

The GasbyGas system's Work Breakdown Structure (WBS) was created to guarantee a methodical and effective approach to project execution. It breaks the project down into core areas, each of which focuses on a crucial component of system development, implementation, and upkeep. In addition to guaranteeing the project's successful completion, this organized breakdown aids in improved resource allocation, deadline management, and accountability.

1. Project Management

From the start of the project until its conclusion, project management is essential. Stakeholder communication, risk assessment, project planning, requirement analysis, and budgeting are all included. These tasks guarantee that the project is carried out in accordance with company objectives while staying within the specified scope, timing, and cost restrictions. Issues like scope creep, financial overruns, and delays could arise if the project is not properly managed.

2. System Design

Determining the system's technological and architectural basis is the main goal of system design. It covers UI/UX considerations, API design, UML modeling, and architectural design. Software that satisfies functional requirements and is scalable and maintained is guaranteed by proper system design. Incorporating UI/UX design at this point guarantees that user requirements are taken into account right away, producing an interface that is simple to use and intuitive.

3. Backend Development

The implementation of the system's essential features is the responsibility of Backend Development (Spring Boot). This covers order processing, stock tracking, dispatch management, gas request management, and user authentication. The backend guarantees effective, safe, and scalable operations by utilizing Spring Boot. Backend logic that is properly organized ensures data security and integrity while facilitating seamless data flow between different system components.

4. Frontend Development

A smooth and interesting user experience requires frontend development (HTML & JavaScript). The creation of several dashboards for administrators, outlet managers, and

domestic and commercial clients is included in this area. The dispatch dashboard and registration/login forms guarantee that users may effectively engage with the system. Better usability and accessibility are guaranteed by a well-designed frontend, which increases user happiness and adoption.

5. Database Development

The main goal of database development is to organize and manage the data needed for system functions. For users, roles, gas requests, orders, stock, and price, the system requires a number of tables. Secure storage, data consistency, and quick query execution are all guaranteed by a well-optimized database. Maintaining system speed and facilitating future scaling depend heavily on database design.

6. API and Integration

Frontend and backend components can communicate with each other without any problems thanks to API & Integration. While email notifications improve user engagement and offer crucial updates, APIs allow data flow across various system modules. This category makes sure that every part of the system functions properly, giving users a seamless experience.

7. Testing and QA

In order to guarantee the system's dependability and security, testing and quality assurance are essential. It covers a range of testing methodologies, including security, performance, regression, integration, and user acceptability testing. Prior to deployment, these tests aid in finding and repairing flaws, guaranteeing a reliable and effective system. The system may experience bugs, crashes, and performance snags if it is not properly tested.

8. Deployment & Maintenance

Making the system operational and guaranteeing its long-term stability are the main goals of deployment and maintenance. It covers server configuration, setup, logging, monitoring, and ongoing issue fixes. The system must be deployed effectively and maintained continuously to be safe, effective, and current with changing business requirements. Frequent monitoring helps to prevent system breakdowns by proactively identifying and fixing issues.

The GasbyGas system will be developed and implemented in a methodical manner thanks to this WBS. In order to provide a top-notch, dependable, and user-friendly solution, each component is essential. The project can accomplish its objectives effectively while lowering risks and optimizing performance by adhering to this planned breakdown.

4.4 Gantt Chart

The GasByGas system's project schedule can be seen visually by a Gantt chart, which displays the order and timing of all activities and tasks specified in the Work Breakdown Structure (WBS). It enables efficient scheduling, progress tracking, and task completion on schedule for the project team. The GasByGas project's Gantt chart, which was created using the previously provided WBS, is shown below.

4.5 Gantt Chart for System

Year				2024/2025							
Month				December				January			
Week				2	3	4	5	1	2	3	4
Task	Start date	End date	Completion								
Project management											
Identify requirements	10/12/24	12/12/24	100%								
Feasibility study	12/12/24	14/12/24	100%								
Create SRS	12/12/24	14/12/24	100%								
System design											
Architecture diagram	15/12/24	15/12/24	100%								
UML diagrams	15/12/24	17/12/24	100%								
ER diagram	17/12/24	17/12/24	100%								
UI/UX design	18/12/24	21/12/24	100%								
Database development											
Create flyway scripts	22/12/24	28/12/24	100%								
Backend development											
Repository layer	29/12/24	31/12/24	100%								
Service layer	01/01/25	05/01/25	100%								
Controller layer	06/01/25	08/01/25	100%								
Frontend development											
Style scripts	22/12/24	24/12/24	100%								
Web page structures	25/12/24	30/12/24	100%								
Data retrieve and logics	31/12/24	07/01/25	100%								
API & Integration											
Fetch data	08/01/25	11/01/25	100%								
Implement email system	08/01/25	10/01/25	100%								
Testing & QA											
System testing	12/01/25	25/01/25	100%								
Performance testing	19/01/25	26/01/25	100%								
Usability testing	19/01/25	26/01/25	100%								
Exploratory testing	23/01/25	23/01/25	100%								
Regression testing	19/01/25	25/01/25	100%								
Sanity testing	19/01/25	26/01/25	100%								
Smoke testing	25/01/25	01/02/25	100%								
Acceptance testing	27/01/25	27/01/25	100%								
Deployment & Maintenance											
Bug fixing	19/01/25	25/01/25	100%								
Performance tuning	22/01/25	26/01/25	100%								
Deploying to production	28/01/25	31/01/25	100%								

Table 9_Gantt Chart for System

4.6 Rationale for the Gantt chart

The GasbyGas system's Gantt chart is set up to guarantee effective workflow, distinct job dependencies, and on-time project completion. The justification for each stage is provided below, along with relevant dates:

1. Project Management (December 10 – December 14, 2024)

- **Identify Requirements (10/12/24 – 12/12/24):** Makes sure that the needs of the system are understood before moving forward with design and development.
- **Feasibility Study (12/12/24 – 12/12/24):** Establishes the technical, financial, and operational feasibility of the project.
- **Create SRS (12/12/24 – 14/12/24):** Records needs, including functional and non-functional, for development use.

2. System Design (December 15 – December 21, 2024)

- **Architecture Diagram (15/12/24 – 15/12/24):** Outlines the high-level components and structure of the system.
- **UML Diagrams (15/12/24 – 17/12/24):** Helps developers comprehend system flow by visualizing system interactions.
- **ER Diagram (17/12/24 – 17/12/24):** Outlines the layout of databases to guarantee effective data linkages.
- **UI/UX Design (18/12/24 – 21/12/24):** Makes an interface that is both aesthetically pleasing and easy to use.

3. Database Development (December 22 – December 28, 2024)

- **Create Flyway Scripts (22/12/24 – 28/12/24):** Ensures consistency across environments by automating database version control.

4. Backend Development (December 29, 2024 – January 8, 2025)

- **Repository Layer (29/12/24 – 31/12/24):** Manages data retrieval and database interactions.
- **Service Layer (01/01/25 – 05/01/25):** Carries out essential processing functions and business logic.
- **Controller Layer (06/01/25 – 08/01/25):** Routes data between the frontend and backend and maintains API endpoints.

5. Frontend Development (December 22, 2024 – January 7, 2025)

- **Style Scripts (22/12/24 – 24/12/24):** Uses JavaScript and CSS style to create a unified user interface.
- **Web Page Structures (25/12/24 – 27/12/24):** Creates the main application pages' design and navigation.
- **Data Retrieval and Logics (31/12/24 – 07/01/25):** Enables real-time data exchanges by connecting frontend elements to APIs.

6. API and Integration (January 8 – January 10, 2025)

- **Fetch Data (08/01/25 – 11/01/25):** Enables smooth data flow by integrating frontend and backend APIs.
- **Implement Email System (08/01/25 – 10/01/25):** Sends out automated email notifications for confirmations and system alerts.

7. Testing and QA (January 12 – January 27, 2025)

- **System Testing (12/01/25 – 15/01/25):** Guarantees that every system component works as a whole.
- **Performance Testing (19/01/25 – 21/01/25):** Evaluates the responsiveness, scalability, and speed of the system.
- **Usability Testing (19/01/25 – 21/01/25):** Assesses navigational ease and user experience.
- **Regression Testing (23/01/25 – 25/01/25):** Confirms that upgrades don't interfere with already-existing functionality.

- **Sanity Testing (19/01/25 – 25/01/25):** Verifies the accuracy of minor bug fixes and code modifications.
- **Smoke Testing (25/01/25 – 25/01/25):** Verifies that essential features are operational prior to deployment.
- **Security Testing (25/01/25 – 27/01/25):** Finds weaknesses and makes data protection stronger.

8. Deployment and Maintenance (January 25 – January 31, 2025)

- **Bug Fixing (25/01/25 – 25/01/25):** Fixes important problems discovered during testing.
- **Performance Tuning (29/01/25 – 26/01/25):** Enhances system performance prior to widespread implementation.
- **Deployment to Production (28/01/25 – 31/01/25):** Releases the system's final version to end users.

A logical workflow is ensured by the Gantt chart, which arranges jobs according to dependencies and resource availability. The project reduces risks, delays, and inefficiencies by organizing phases in a sequential manner, guaranteeing seamless execution and superior delivery.

4.7 Resource Allocation

Task	Team's members assigned	Effort House
01. Planning		
Project Initiation	Project Manager (Ravindu De Silva)	25
Requirement Gathering	Business Analyst (Dewumini Dissanayaka), Senior Developer (Sachini Lunuwila)	45
Feasibility Study	Project Manager (Ravindu De Silva, System Architect (John Doe)	50
02. Design		
System Design	System Architect (Ahinsala Dissanayaka)	200
UI/UX Design	UI/UX Designer (Ahinsala Dissanayaka)	300

Database Design	(Ravindu De Silva)	215
03. Development		
Frontend Development	Senior Developer (Sachini Lunuwila)	400
Backend Development	Senior Developer (Sachini Lunuwila)	400
Integration	Senior Developer (Sachini Lunuwila), System Architect (Dewumini Dissanayaka)	200
04. Testing		
Unit Testing	Quality Assurance Tester (Dewumini Dissanayaka)	100
System Testing	Quality Assurance Tester (Dewumini Dissanayaka), Senior Developer (Sachini Lunuwila)	300
05. Deployment		
Deployment Preparation	Project Manager (Ravindu De Silva)	
Go-live	System Architect (Ahinsala Dissanayaka)	
Post-deployment Monitoring	(Sachini Lunuwila)	
Deployment & Preparation	Project Manage (Dewumini Dissanayaka)	120
Total Estimated Effort: 3,050 Hours		

Table 10_Resource Allocation

Effective allocation of resources is made certain by the GasByGas project's 3,050 hours of anticipated total effort, which precisely matches the allotted hours. The roles of the team members, their hourly rates, and the needs of the project determine how much time is allotted to each activity. Within the limitations of the project, this strategic distribution maximises efficiency by carefully balancing task durations, technological requirements, and labour expenses. Expertise-based assignments make sure that the correct people are working on the right tasks by matching the team's talents to particular task needs throughout all project phases. By continuously recording hours spent, following the project schedule, and keeping a careful eye on the budget, this method ensures that financial and schedule goals are reached without reducing quality.

4.8 Project Milestones

Project milestones represent significant points in the project's lifecycle, marking the completion of major phases. For the GasByGas project, the following milestones are identified

Milestone	Success Criteria
Planning Phase Completion	• Project scope finalized • Requirements documented • Feasibility study approved
Design Phase Completion	• System architecture signed off • UI/UX designs approved by stakeholders
Development Phase Completion	• Front-end, back-end, and integration tasks completed • All components tested
Testing Phase Completion	• Unit tests passed • System tests completed with zero critical defects
Deployment Preparation	• Environment configured • Deployment checklist finalized
Go-Live	• System operational for end-users • Core functions verified
Post-deployment Monitoring	• Critical issues resolved

Table 11_Project Milestones

These milestones help in tracking project progress, provide checkpoints for evaluations, and ensure that the project is proceeding according to plan.

4.9 Rationale for the Milestones

The project's milestones are intended to guarantee organised and methodical development cycle progress, allowing for on-time delivery while upholding high standards of quality. This is the justification for every milestone.

- **Planning Phase Completion**

  The project's foundation is set up during the planning phase. All stakeholders will have a clear grasp of the project's goals, scope, and viability if the project scope is finalised, requirements are documented, and the feasibility study is finished. This phase lays the

foundation for a successful project by defining expectations and coordinating team actions with the project's goals.

- **Design Phase Completion**

  The design phase is essential since it establishes the functionality and appearance of the system. System architecture and UI/UX designs must be approved in order for the technological framework to meet usability requirements, user expectations, and business needs. By assisting stakeholders in visualising the finished product, this phase gives them assurance that the system's architecture and user interface are suitably built for scalability, functionality, and usability.

- **Development Phase Completion**

  The development phase is when the main elements of the system are constructed. The completion of integration, back-end, and front-end jobs guarantees that every part functions as a whole. By finishing and testing every component, the chance of unanticipated problems during the last stages is reduced and the system is guaranteed to be reliable, feature-complete, and prepared for additional testing.

- **Testing Phase Completion**

  Testing makes sure there are no serious flaws that could affect system performance or user experience and that the system is operating as intended. The software's stability, dependability, and compliance with all functional requirements are guaranteed by effective unit and system tests. Delivering a high-quality product during this phase is crucial to guaranteeing that no significant faults will negatively impact the system's user experience once it goes live.

- **Deployment Preparation**

  The system is guaranteed to be production-ready at the Deployment Preparation milestone. Completing the deployment checklist and setting up the environment ensures that the system will be deployed successfully and without any technical issues. Technical preparedness can be confirmed during this phase, which guarantees a seamless deployment and lowers the possibility of system failure or outage.

4.10 Network Diagram

One of the most important tools for effectively managing the GasByGas project is the network diagram. The project manager can monitor job sequences and spot possible bottlenecks early on because to its clear, visual depiction of all project tasks and their dependencies. The graphic aids in scheduling optimisation to guarantee timely completion by illuminating the connections

between tasks and identifying the important path. By helping with resource allocation and job prioritisation, this planning tool minimises delays and facilitates more efficient task progression across all project phases. It also acts as a vital point of reference for preserving project schedules, which keeps the project on course to fulfil budgetary targets and deadlines.

Task Dependencies Table

Activity	Predecessor	Duration
Project management, A1		1
System design, A2	A1	1
Database development, A3	A2	1
Backend development, A4	A2, A3	2
Frontend development, A5	A2	3
API & Integretion, A6	A4, A5	1
Testing & QA, A7	A6	3
Deployment & Maintenance, A8	A7	2

Table 12_Task Dependencies Table

The table uses shorter identifiers for convenience and clearly describes the duration and interdependence of each action. Based on the Work Breakdown Structure (WBS), this thorough breakdown serves as the foundation for the Gantt chart and network diagram. The integrity of the project workflow is maintained by making sure that activities are carried out in the right order by precisely defining the tasks and their interdependencies. Accurate scheduling and the visualisation of the critical route are made possible by this scientific approach, which also helps to ensure timely completion and effective project management.

The table organizes project activities with the following details:

1. **Activity**

- Represents individual project tasks
- Uses short alphanumeric codes (e.g., T1, T1.1) for easy identification

2. **Duration (Days)**

- Specifies the estimated time required to complete each task

3. **Dependencies**

- Indicates prerequisite tasks
- Ensures tasks are executed in the correct sequence

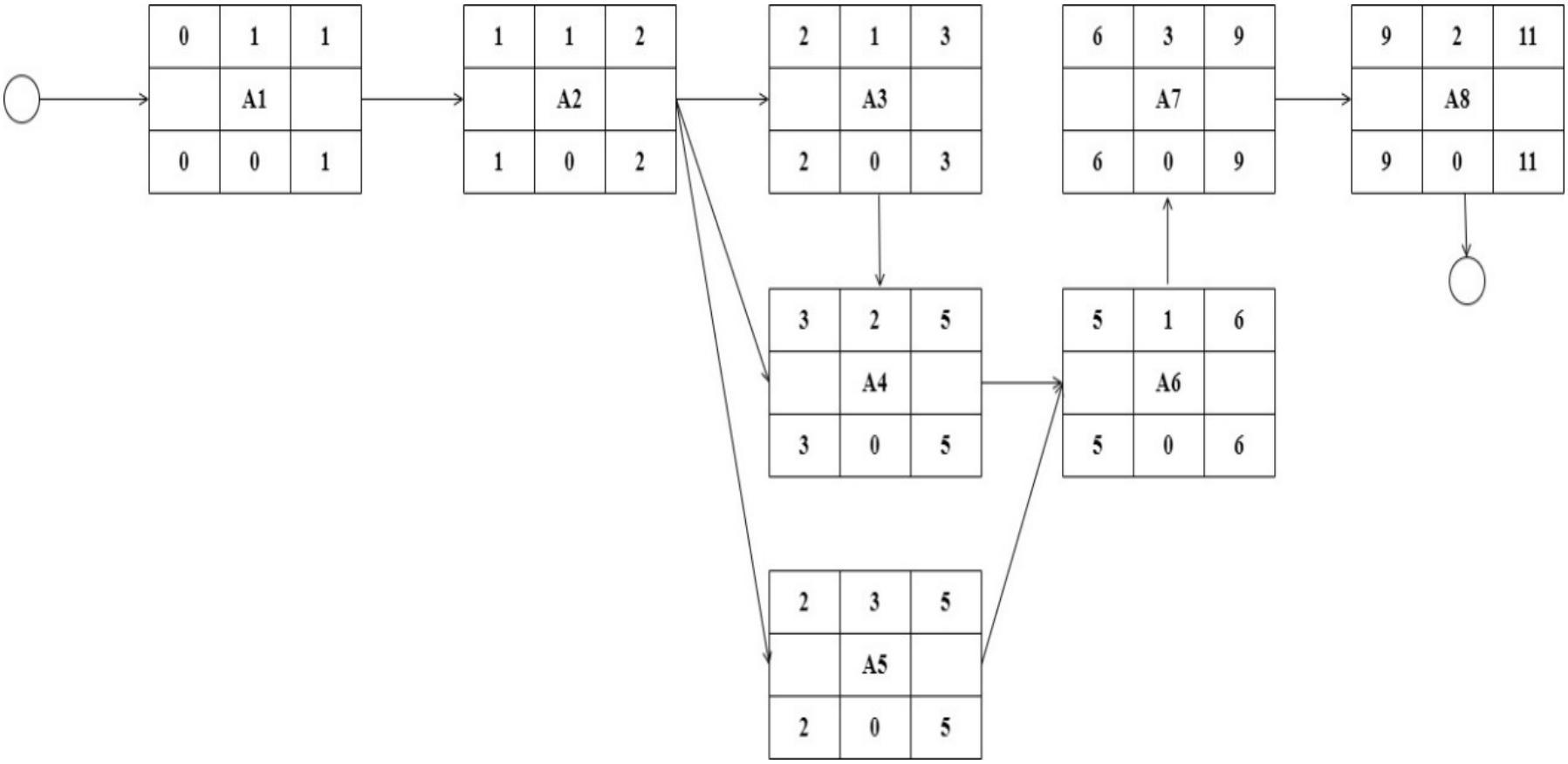

Table 13_Activity Network Diagram

4.11 Critical Path of the Project

The Critical Path in project management is the sequence of tasks that determines the minimum project duration. It is the longest path through the project and consists of tasks that have no slack or buffer time. Any delay in these tasks will directly impact the project's overall completion time. Identifying the critical path is crucial for project managers, as it helps in prioritizing tasks and allocating resources efficiently to avoid delays. By focusing on the critical path, project managers can ensure that key tasks are completed on time, helping to meet the project's deadline and objectives.

Critical Path identification

i. A1 > A2 > A3 > A4 > A6 > A7 > A8 = 11

ii. A1 > A2 > A4 > A6 > A7 > A8 = 10

iii. A1 > A2 > A5 > A6 > A7 > A8 = 11

Critical path is 11.

Analyse the critical Path

The critical path for this project outlines the sequence of tasks that directly determine the project's total duration. It begins with the planning phase, where Project Initiation (T1.1) sets the foundation by defining scope and objectives. This is immediately followed by Requirements Gathering (T1.2) and Feasibility Study (T1.3), which must be completed before any design work can begin. The path then moves to the design phase, where both System Design (T2.1) and UI/UX Design (T2.2) must be finalized to enable development work to start. During development, Front-end Development (T3.1) and Back-end Development (T3.2) run in parallel but must both be completed before Integration (T3.3) can occur. The critical path then proceeds through rigorous testing phases - Unit Testing (T4.1) and System Testing (T4.2) - which are essential for quality assurance. Finally, the path concludes with deployment activities: Deployment Preparation (T5.1), Go-live (T5.2), and Post-deployment Monitoring (T5.3). Each of these tasks is interdependent, with zero float time, meaning any delay in any of these tasks would directly extend the project's overall timeline. The critical path methodology helps project managers identify these crucial tasks that require close monitoring to ensure the project stays on schedule.

Chapter 05: Project Maintenance

5.1 Project Maintenance

The Online Gas Requesting and Delivery Management System has a thorough and proactive maintenance plan in place to ensure its long-term sustainability and efficiency. Regular maintenance guarantees a seamless experience for all users, including administrators and customers, in addition to keeping the system safe and operational.

Regular Software Updates

The system is updated on a regular basis to incorporate the newest technology and adapt to changing user and business needs. Security patch deployment, feature enhancement, known problem fixes, and preserving compatibility with more recent devices, operating systems, and browsers are all examples of updates. Regular updates maintain the system up to date with industry standards and competitive.

Issue Tracking and Debugging

A strong issue tracking system is used to record, keep an eye on, and fix errors or weaknesses. By giving developers and stakeholders the ability to monitor the status of issues in real time, this platform guarantees accountability and transparency. Debugging is prioritized according to severity, guaranteeing that urgent problems are resolved right away to reduce downtime and service disruptions.

Performance Optimization

The system is regularly checked for inefficiencies and performance snags. Regular analysis is done on key performance indicators (KPIs) like website load speed, database query performance, and server response time. Load balancing, caching tactics, and database indexing are examples of optimization approaches that provide quick and responsive operations even during periods of high usage.

Security Monitoring

Ensuring user trust and data privacy requires a high level of security. Security audits are conducted on a regular basis to identify and address possible risks. Techniques for encrypting data are used while it is in transit and while it is at rest. Additional safeguards are included to stop unwanted access and safeguard private data, including firewall setups, intrusion detection systems (IDS), and two-factor authentication.

Database Maintenance

Regular backups of the system's databases guard against data loss in the event of cyberattacks or system breakdowns. To cut down on storage overhead, outdated and unnecessary data is either archived or deleted in accordance with data retention standards. SQL queries are constantly improved to reduce resource usage and speed up data retrieval.

User Support and Feedback Handling

A specialized user support team is set up to provide prompt assistance to delivery personnel and customers. Chatbots, a hotline, and email are examples of support channels. Feedback from users is actively gathered via surveys, reviews, and suggestion forms. Incorporating feasible recommendations into upcoming system upgrades guarantees that the system develops in line with user expectations.

Documentation Updates

All technical documentation, API references, and user guides are regularly updated to take into account new features or modifications to the system's operation. Effective system administration, development, and end-user comprehension are guaranteed by well-maintained documentation.

Third-party Service Monitoring

The system depends on a number of outside services, including payment gateways, email providers, and SMS gateways. These services are kept under constant observation to make sure

they are operating as planned. In order to quickly install alternate solutions, automated notifications are set up to notify administrators of any outage or deterioration in service quality.

Scalability Enhancements

Scalability becomes crucial as the user base grows. Periodically, the system infrastructure is assessed and improved to meet growing demand. This includes optimizing cloud resources, increasing bandwidth, and scaling server capacity. The architecture of the system is scalable and flexible, making it simple to expand into other geographical areas and add more functions.

Conclusion

In conclusion, the GasByGas Online Gas Requesting and Delivery System project presents a significant opportunity to streamline the gas distribution process through an agile, user-friendly web application. The project plan, including the Work Breakdown Structure (WBS), Gantt Chart, and resource allocation, has been meticulously crafted to ensure the systematic completion of each project phase. Key milestones and a well-defined critical path will help maintain focus on project deliverables, ensuring the platform is delivered on time, within budget, and with all required features.

The involvement of both full-time and part-time team members ensures an efficient use of resources, while the allocation of tasks and timelines adheres to the principles of good project management. By incorporating clear documentation, training, and support phases, the project will not only meet the technical requirements but also equip the client with the necessary tools to effectively use and manage the system post-deployment.

Ultimately, the successful implementation of this system will enhance the efficiency of the gas distribution process, improve customer satisfaction, and provide GasByGas (Pvt) Ltd. with a scalable solution that can grow with the demands of the future.

References

ISO - International Organization for Standardization (2017). *ISO/IEC 12207:2008*. [online] ISO. Available at: https://www.iso.org/standard/43447.html.

IEEE Standard for Software Project Management Plans Software Engineering Standards Committee of the IEEE Computer Society IEEE-SA Standards Board. (1998). Available at: https://wildart.github.io/MISG5020/standards/IEEE-1058-1998.pdf.

Microsoft.com. (2024). *Microsoft Project for the Web | Manage Projects Online*. [online] Available at: https://www.microsoft.com/en-us/microsoft-365/planner/microsoft-project.

Project Management Institute (2021). *PMBOK® Guide*. [online] Pmi.org. Available at: https://www.pmi.org/standards/pmbok.

Appendix

Individual Contribution

Name	Contribution
M.R.G.L. De Silva	• Instruct, manage and lead the team to the end of the course work. • Identify requirements and define scope with the team. • Created mind maps and SRS. • Reseach about similar systems and got ideas. • Planned system designs and draw system designs with the team. • Created test plan and test cases. • Decided tools and technologies to use. • Planned, design and build base codes for the product. • Developed, tested and debugged outlet manger related modules like outlet registration, outlet dashboard and outlet gas requests.

Table 14_Individual Contribution

Own reflection of my experience

This is my first time building a whole product this big and working with a team for a coursework so I had many difficulties and did so many mistakes and learned valuable lessons and I can avoid those mistakes in the future.

- Mistook hidden requirements and added unwanted features, later removed and fixed them. This the first time identify requirements for a project and building a SRS, after done this now I am confident on it.
- I had a limited knowledge and forgot UML diagram designs and had no idea about architecture diagram designs. I researched and found how to them and now I am good at those designs.
- I ran into an issue when selecting programming languages because other team members know languages I don't know and they only have limited knowledge about those languages they know, So I had to choose language I know since I am better at it and I have to lead and there are no other options.
- Even I am good at backend development I have no experience in frontend development and team members only had limited knowledge, so had a big issue when developing webpages.

- Creating test plan and test cases, database and developing backend was very easy with prior experience.
- Even I am good at backend development there had a lot of error/blockers for unknown reasons and terminal was unable to provide proper information. Wasted a lot of time because of this and somehow got fixed the issues after a lot of research and putting a lot of time, fixed issues after using AI chatbots, even with help from AI it wasn’t an easy task I had to give up many times.