STAFF HOUSE MANAGEMENT ONLINE SYSTEM
KD SULTAN ISMAIL
TANJUNG PENGELIH, PENGERANG, JOHOR

SITI MAHIRAH BINTI M TAHRIR

UNIVERSITI TEKNOLOGI MALAYSIA

STAFF HOUSE MANAGEMENT ONLINE SYSTEM
KD SULTAN ISMAIL
TANJUNG PENGELIH, PENGERANG, JOHOR

SITI MAHIRAH BINTI M TAHRIR

A thesis submitted in fulfilment of the
Requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

Faculty of Computing
Universiti Teknologi Malaysia

12 JULY 2024

DECLARATION

I declare that this thesis entitled "STAFF HOUSE MANAGEMENT ONLINE SYSTEM KD SULTAN ISMAIL TANJUNG PENGELIH, PENGERANG, JOHOR" is the result of my research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : SITI MAHIRAH BINTI M TAHRIR

Date : 12 JULY 2024

DEDICATION

This thesis is dedicated to my beloved husband, a devoted TLDM officer, whose unwavering support and encouragement have been my greatest strength throughout this journey.

Thank you for your patience and understanding, my dear children, who have provided me with inspiration and joy.

To my parents, for their unending love, guidance, and sacrifices that have shaped who I am today. Your belief in me has been my constant source of motivation.

ACKNOWLEDGEMENT

First and foremost, I'd like to thank my supervisor, Dr. Ismail Fauzi Bin Isnin, for his invaluable guidance, support, and patience throughout this project. His expertise and encouragement were essential in completing this work.

I sincerely thank my family for their unwavering support. Thank you to my husband for his understanding and encouragement, my children for their patience, and my parents for their love and belief in me.

I am also grateful to my colleagues and friends for their encouragement and assistance, which have served as a source of inspiration and strength.

Last but not least, I'd like to thank the Faculty of Computing at Universiti Teknologi Malaysia for providing the resources and opportunities that enabled this project.

ABSTRACT

This project focuses on creating the Staff House Management Online System KD Sultan Ismail Tanjung Pengelih, Pengerang, Johor, a web-based platform that will improve the management of staff houses at KD Sultan Ismail (KDSI), Tentera Laut Diraja Malaysia (TLDM), Johor. The system's goal is to provide an efficient solution for house allocation, tracking maintenance activities, and managing guest check-ins and check-outs. The primary goals are to design a user-friendly interface, implement a real-time dashboard for house status updates, enable maintenance complaint submissions with image uploads, and set up a secure system for managing guest visits. By automating and digitizing house management processes, the system is expected to improve operational efficiency, reduce manual errors, and increase user satisfaction. The project takes a step-by-step development approach, with proposal, planning, and design phases. User authentication, house management, maintenance management, and guest management are among the key functionalities. The front end of the system is built with web technologies such as HTML, CSS, JavaScript, and Bootstrap, while the back end is robust for data storage and processing. This report describes the project's early stages, including the problem background, project objectives, scope, and importance, as well as an analysis of the current system, a comparison with other house management systems, and an examination of the technologies used. The initial stages of development show that the KDSI Staff House Management System has the potential to significantly improve the efficiency and effectiveness of KDSI house management.

ABSTRAK

Projek ini memfokuskan kepada pembangunan Sistem Pengurusan Rumah Staf Secara Atas Talian KD Sultan Ismail Tanjung Pengelih, Pengerang, Johor, sebuah platform berasaskan web yang direka untuk meningkatkan pengurusan rumah kakitangan di KD Sultan Ismail (KDSI), Tentera Laut Diraja Malaysia (TLDM), Johor. Sistem ini bertujuan untuk menyediakan penyelesaian yang cekap bagi pengagihan rumah, penjejakan aktiviti penyelenggaraan, dan pengurusan daftar masuk dan keluar tetamu. Objektif utama adalah untuk mewujudkan antara muka mesra pengguna, melaksanakan papan pemuka masa nyata untuk kemas kini status rumah, membolehkan penghantaran aduan penyelenggaraan dengan muat naik imej, dan mewujudkan sistem yang selamat untuk menguruskan lawatan tetamu. Dengan mengotomatisasi dan mendigitalkan proses pengurusan rumah, sistem ini diharapkan dapat meningkatkan kecekapan operasi, mengurangkan kesilapan manual, dan meningkatkan kepuasan pengguna. Projek ini menggunakan pendekatan pembangunan berperingkat, termasuk fasa cadangan, perancangan, dan reka bentuk. Fungsi utama termasuk pengesahan pengguna, pengurusan rumah, pengurusan penyelenggaraan, dan pengurusan tetamu. Sistem ini menggunakan teknologi web seperti HTML, CSS, JavaScript, dan Bootstrap untuk bahagian depan, dengan backend yang kukuh untuk penyimpanan dan pemprosesan data. Laporan ini memperincikan peringkat awal projek, termasuk latar belakang masalah, objektif projek, skop, dan kepentingan, serta menyediakan analisis sistem semasa, perbandingan dengan sistem pengurusan rumah sedia ada, dan pemeriksaan teknologi yang digunakan. Peringkat pembangunan awal menunjukkan bahawa Sistem Pengurusan Rumah Kakitangan KDSI mempunyai potensi untuk meningkatkan kecekapan dan keberkesanan pengurusan rumah di KDSI.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
RDBMS	Relational Database Management System
MVC	Model-View-Controller
IDE	Integrated Development Environment
CI/CD	Continuous Integration/Continuous Deployment
JWT	JSON Web Tokens
SSL/TLS	Secure Sockets Layer/Transport Layer Security
AES	Advanced Encryption Standard

LIST OF SYMBOLS

δ	-	Minimal error
D, d	-	Diameter
F	-	Force
v	-	Velocity
p	-	Pressure
I	-	Moment of Inersia
r	-	Radius
Re	-	Reynold Number

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Staff House Management Online System for KD Sultan Ismail Tanjung Pengelih, Pengerang, Johor, represents a paradigm shift in how staff housing facilities are managed. Historically burdened by archaic manual processes, the current management system is riddled with inefficiencies that jeopardize both operational effectiveness and resident satisfaction. This innovative system aims to replace these antiquated practices through a sophisticated, automated approach. It promises to improve the efficiency, accuracy, and security of housing management processes while also introducing previously unattainable levels of operational fluidity. The primary goal is to streamline maintenance requests, allow for real-time monitoring of housing statuses, and manage guest visits with unparalleled security and efficiency.

The implementation of this system is expected to address several critical challenges in the management of staff housing. For years, manual record-keeping and ad hoc management practices have resulted in maintenance delays, inaccuracies in housing allocation, and security breaches during guest visits. These issues have not only strained resources but have also resulted in widespread dissatisfaction among the staff who work in these facilities. By automating these processes, the system will significantly reduce human error and administrative workload. Furthermore, it will use advanced data analytics to provide actionable insights, enabling proactive maintenance and management. This change is expected to significantly improve the responsiveness of services provided to residents, ultimately improving their living experience.

KD Sultan Ismail's transition to this automated system will set a new standard for housing management within military installations. The system's architecture is intended to be both robust and flexible, allowing for future technological

advancements and operational requirements. This ensures that the system remains adaptable and scalable, able to evolve in response to changing military housing needs. Furthermore, establishing a precedent for digital transformation in this sector demonstrates KD Sultan Ismail's commitment to implementing innovative solutions that improve operational efficiency and employee welfare. The facility's goal with this initiative is not only to improve its operational capabilities, but also to provide a more secure, efficient, and comfortable environment for its employees.

1.2 Problem Background

The management of staff housing at KD Sultan Ismail Tanjung Pengelih has long been hampered by antiquated manual processes that are not only time-consuming but also prone to errors. This traditional approach to housing management cannot provide real-time updates, resulting in significant operational inefficiencies and responsiveness. As a result, maintenance requests are frequently delayed beyond acceptable levels, with the root cause being the current inefficient methods of communication and tracking. This inefficiency in dealing with maintenance issues not only jeopardizes the structural integrity and safety of housing facilities but also significantly reduces employee satisfaction and trust in the management system.

Furthermore, the process of managing guest visits is riddled with inefficiencies, which manifest as long waiting times and disorganized lines at the front gate, posing serious security risks and operational bottlenecks. The current system's inability to effectively manage these aspects increases vulnerability, with unauthorized access potentially jeopardizing the safety of residents and the facility. Manual entry and verification processes are not only slow but also vulnerable to human error, exacerbating security concerns.

The culmination of these issues highlights the critical and urgent need for an overhaul of current management practices. Implementing a comprehensive, automated system is critical for streamlining all aspects of base housing management. Such a system would ensure that maintenance tasks are completed quickly and efficiently, improving the overall upkeep of the homes. It would also transform the management

of guest visits by implementing a secure, digital check-in process that speeds up verifications, reduces wait times, and significantly improves security protocols.

The proposed system would not only address the immediate operational challenges but would also lay the groundwork for future improvements. By automating routine tasks, resources would be freed up, allowing employees to focus on more strategic activities that improve the facility's living conditions and security. Furthermore, it would provide the administration with valuable data insights into day-to-day operations, allowing for better-informed decision-making and proactive management of the housing facilities.

In summary, the transition to an automated Staff House Management Online System is more than a functional upgrade; it is a necessary evolution to meet KD Sultan Ismail Tanjung Pengelih's current needs, ensuring that its operational practices are secure, efficient, and in line with modern standards.

1.3 Project Aim

The primary goal of this project is to develop and implement the Staff House Management Online System at KD Sultan Ismail, which will transform the way staff housing operations are managed. This sophisticated system is intended to increase housekeeping efficiency, streamline maintenance tracking, and significantly improve guest visit handling. The initiative aims to replace outdated manual processes with a fully automated, scalable, and user-friendly solution tailored to the needs of staff and administrators. The development process includes conducting a thorough analysis of the current operational requirements, designing an intuitive and efficient user interface that simplifies user interactions, and ensuring a strong backend capable of managing complex data and real-time operations. Furthermore, rigorous testing and evaluation are integrated into the project to ensure the systems

1.4 Project Objectives

The objectives of the project are:

(a) Examine the system's requirements to ensure they meet the needs of KDSI staff and administrators.

(b) Create a user-friendly interface and an efficient backend system to improve house management, maintenance, and guest visits.

(c) Implement and evaluate the system to ensure that it works properly and improves existing processes.

1.5 Project Scope

The scopes of the project are:

(d) The system will handle staff-house information, maintenance requests, and guest visits in a centralised and efficient manner.

(e) The system generates real-time reports on house statuses, maintenance activities, and guest visits.

(f) The system allows users to securely and efficiently manage guest check-ins and check-outs, as well as submit maintenance requests with image uploads.

1.6 Project Importance

The implementation of this system is critical for a variety of reasons. It increases operational efficiencies by automating traditional housekeeping tasks, which reduces manual labour, minimizes errors, and speeds up overall operations. The system's ability to provide real-time updates enables staff and administrators to make

faster decisions and respond to housing and maintenance issues. Enhanced security protocols for managing guest visits greatly improve facility safety. Furthermore, the system's user-friendly interface significantly improves usability and satisfaction among employees and administrators, resulting in a more efficient and pleasant working environment.

1.7 Report Organization

This report is organized into chapters that provide a comprehensive overview of the project. It begins with an overview of the project, including the background, goals, objectives, and scope. The following chapters consist of a review of existing systems and relevant technologies, a discussion of the system development methodology used for this project, a detailed analysis of the system requirements and architectural design, and a conclusion that summarizes the project's accomplishments and discusses potential future improvements. This structured approach ensures that readers understand the project's intent, development process, and expected impact on KD Sultan Ismail's housing management practices.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a thorough review of the literature relevant to the Staff House Management Online System for KD Sultan Ismail Tanjung Pengelih, Pengerang, Johor. It seeks to place the project within the larger context of organizational and operational management theories and practices relevant to military housing. Understanding the unique structural dynamics of KD Sultan Ismail's (KDSI) organization in terms of house management is of particular importance. This entails evaluating the current system's performance and efficacy in meeting military personnel's housing needs, as well as comparing it to existing systems implemented in similar military and civilian contexts.

The review begins with an examination of KDSI's organizational structure, which was specifically designed to optimize house management operations. This segment delves into the decision-making hierarchies and operational workflows that support current management practices. It also evaluates the challenges and limitations of the current manual processes, which have an impact on the overall efficiency and effectiveness of base housing management.

Following that, the chapter examines several existing house management systems used in similar military establishments and compares them to civilian residential management practices. This comparative analysis aims to identify best practices and innovative management strategies that could be used to improve the KDSI system. The effectiveness of various technologies currently used in advanced house management systems, such as automated maintenance tracking and digital guest management.

Finally, the literature review examines the technological landscape that enables modern housing management systems. It investigates the most recent advances in software and hardware that are influencing the future of residential administration, with a focus on scalability, security, and user interface design. This chapter lays the groundwork for the proposed enhancements to the Staff House Management Online System by combining findings from these various areas, ensuring that the system's design is both innovative and based on proven management principles.

2.2 Company Organisational Background

2.2.1 Company Organizational Structure: KDSI Relevant to House Management

KD Sultan Ismail's (KDSI) organizational structure is meticulously designed to ensure that staff housing facilities are managed efficiently and effectively. This hierarchical structure allows for clear lines of communication and authority, which are critical for maintaining high standards in housing management. The roles and responsibilities are clearly defined, ensuring that each level makes an effective contribution to the housing management operations.

2.2.2 Hierarchical Structure:

1. **Commanding Officer (CO):** As the highest authority at KDSI, the Commanding Officer oversees overall decisions and strategic direction. This role entails approving major policies and significant changes in household management, as well as ensuring that housing operations are consistent with the naval base's overall objectives. The CO's strategic vision sets the operational ethos and efficiency standards for the entire housing management team.

2. **Executive Officers (XO):** The Executive Officers report directly to the CO and play an important role in policy implementation. Their primary responsibility is to ensure that all house management activities run smoothly and adhere to established policies. They serve as a liaison between strategic

directives and operational execution, overseeing the implementation of policies and making necessary adjustments to meet changing housing needs.

3. **House Management Officers (HMO):** In charge of the day-to-day operations of staff housing. They are responsible for efficiently managing maintenance schedules, allocating housing, and monitoring occupancy rates. HMOs ensure that living conditions meet established standards and that any problems are resolved promptly. Their role is critical in ensuring the base personnel's daily satisfaction with their living arrangements.

4. **Maintenance Team:** Responsible for the upkeep of housing facilities and reports to HMOs. The Maintenance Team, which consists of skilled technicians and workers, is responsible for all repairs and maintenance tasks within the staff houses. They play an important role in ensuring that all housing facilities are functional and meet safety and comfort standards, which has a direct impact on residents' quality of life.

5. **Administrative Staff:** Assists the HMOs by overseeing all clerical and administrative aspects of house management. This includes managing documentation, processing booking approvals, and keeping detailed records of occupancy and maintenance activities. Their meticulous record-keeping and administrative efficiency ensure that housing operations run smoothly and that any administrative issues are quickly resolved. Organizational Structure for House Management

Figure 2.1 Organizational Structure for House Management

Figure 2.1 above was created using the stacking technique described by Peterson (2018), in which hierarchical levels are represented in a top-down structure, demonstrating the reporting relationships between different roles.

2.2.3 Manual Operation

At KD Sultan Ismail, the house management system is heavily reliant on manual operations that necessitate significant administrative effort. This includes keeping physical files for each house, where allocations, maintenance requests, and tenant information are recorded. Houses are allocated manually based on availability and personnel rank, which frequently ignores individual or urgent needs, potentially resulting in inefficiencies and dissatisfaction. Similarly, maintenance management is handled manually, with submissions and documentation recorded in physical logs and processed by maintenance staff. This approach not only slows down maintenance issue resolution but also lacks transparency and traceability, making it difficult to keep track of the status of ongoing or completed tasks.

2.3 Current System Analysis: Applying Linear Regression

Linear regression is a statistical method used for predictive modelling. It correlates with an independent variable. Linear regression is a statistical technique used in predictive modelling. A correlation exists between an independent variable (x) and a dependent variable (y), such as maintenance needs or housing allocations.

$$y = mx + c$$

The formula represents the relationship where is m the slope indicates the change in y for a unit change in x, and c is the y-intercept, the value of y when x is zero.

Justification of the chosen method, in terms of KD Sultan Ismail's house management system predictive maintenance and allocation, using linear regression, the system could forecast future maintenance requirements based on previous trends, allowing for the scheduling of preventative maintenance before problems become critical. Similarly, housing allocation can be optimized based on staff demand and availability patterns.

Resource Optimization, and predictive analytics through linear regression enable better resource management by allocating resources where they are most likely to be needed, reducing waste and improving response time.

Data-Driven Decisions by quantifying relationships and trends in house management's historical data, linear regression provides a solid foundation for decision-making, facilitating the transition from a reactive to a proactive management model.

The use of linear regression to improve KD Sultan Ismail's house management system promises to streamline operations while also creating a more adaptive, efficient, and proactive management environment. By implementing these predictive models, management can ensure better resource utilization, increased employee satisfaction, and reduced operational overheads.

2.4 Comparison between existing systems

To identify best practices and potential improvements for KD Sultan Ismail's house management system, we compared it to other organizations' existing systems. This comparison is critical for understanding how various technologies and approaches can improve the efficiency and effectiveness of housekeeping operations.

2.4.1 System 1: RealPage Property Management.

RealPage Property Management is a cutting-edge property management system that automates various workflows and offers real-time data and maintenance tracking. According to Table 2.1, RealPage Property Management is efficient, scalable, and comprehensive, but it comes at a high initial cost. More information can be found on the RealPage website.

2.4.2 System 2: Yardi Voyager

Yardi Voyager is a comprehensive system for accounting, property management, and resident services. As shown in Table 2.1, it is durable, user-friendly, and has a wide range of features, but it requires training to use effectively. More information can be found on the Yardi site.

2.4.3 System 3: Buildium Property Management Software.

Buildium Property Management Software is a cloud-based platform that includes a tenant portal and financial tracking. As shown in Table 2.1, it is accessible, simple to use, and inexpensive, but it has few customisation options. More information can be found on the Buildium website.

Table 2.1 Comparison of Existing House Management Systems

System	Key Features	Strengths	Weaknesses	Source
RealPage Property Management	Automated workflows, Real-time data, Maintenance tracking	Efficient, Scalable, Comprehensive	High Initial Cost	RealPage
Yardi Voyager	Integrated accounting, Property management, Resident services	Robust, User-friendly, Extensive features	Requires Training	Yardi
Buildium Property Management Software	Cloud-based, Tenant portal, financial tracking	Accessible, Easy to use, Affordable	Limited customization	Buildium

This detailed comparison not only highlights each system's specific benefits and drawbacks, but also guides the decision-making process for selecting the best system for KD Sultan Ismail, considering operational needs, budget constraints, and ease of integration.

2.5 Technology Used

Examining the technologies used in existing house management systems reveals several key components that can be adapted for KDSI. These technologies have the potential to significantly increase the efficiency and effectiveness of KDSI's house management processes.

2.5.1 Technology 1: Digital Data Entry Systems

Digital data entry systems allow for efficient and accurate recording of owner and tenant information. These systems can be customized for specific needs or used as off-the-shelf solutions. For example, using digital data entry for house allocations and maintenance requests can boost efficiency and reduce errors. Technologies that can be used for this purpose are:

i. **PHP** is a popular server-side scripting language intended for web development. PHP is used to generate dynamic page content and manage data effectively.

ii. **MySQL** is an open-source relational database management system. MySQL is used to store and manage large amounts of structured data efficiently.

2.5.2 Technology 2: Management Systems for Guests and Visitors

A digital guest/visitor management system can be integrated into the main database, connecting guest or visitor information to house owner information for verification. This system ensures that security personnel can easily access and verify visitor information by maintaining a log for each guest or visitor. Technologies that can be used for this purpose are:

i. **Laravel** is a PHP framework that provides a solid foundation for building web applications, including guest management systems.

ii. **Bootstrap** is a front-end framework used to create responsive and mobile-first websites. Bootstrap can be used to design user-friendly interfaces for guest management.

2.5.3 Technology 3: Maintenance Request Issue and Tracking

Implementing a digital application for maintenance requests enables real-time notifications and precise tracking of the maintenance process. The app allows homeowners to submit requests, and the maintenance team can update the status and progress, increasing transparency and satisfaction. Technologies that can be used for this purpose are:

i. **React** is a JavaScript library for creating user interfaces. React allows you to create interactive and dynamic maintenance request forms.

ii. **Node.js** is a JavaScript runtime based on Chrome's V8 JavaScript engine. Node.js is used to create scalable network applications capable of receiving real-time updates for maintenance requests.

iii. **Firebase** is a Google's platform for developing mobile and web applications. Firebase supports real-time database management and authentication.

2.6 Chapter Summary

This chapter thoroughly examined the current house management system at KD Sultan Ismail (KDSI), identifying critical inefficiencies and areas for improvement. This review has discovered several best practices that could be beneficially integrated into KDSI's operations to improve efficiency, accuracy, and user satisfaction.

The investigation revealed that the primary limitations of KDSI's current manual system are labour-intensive, prone to human error, and lacking real-time operational capabilities. These issues not only impede efficient house management but also have an impact on overall satisfaction among both staff and housing administrators. In response to these challenges, the chapter emphasized the ability of advanced digital technologies to transform the current framework into a more efficient, error-free, and user-friendly system.

KDSI's house management system can be significantly improved in functionality and responsiveness by leveraging cutting-edge digital solutions such as PHP and MySQL for digital data entry, Laravel and Bootstrap for guest and visitor management, and a combination of React, Node.js, and Firebase for maintenance request handling. These technologies provide robust platforms for task automation, data streamlining, and real-time tracking of maintenance and housing allocations.

The integration of these technologies will help to shift from a reactive to a proactive management approach. With real-time updates and notifications, KDSI can

anticipate and resolve issues before they escalate, improving operational efficiency and reducing downtime. Furthermore, the use of comprehensive digital tools will allow for the creation of detailed reports and analytics, providing management with actionable insights that can drive strategic decisions and promote continuous improvement.

To summarize, the analysis in this chapter lays the groundwork for a transformative upgrade to KDSI's house management practices. KDSI can expect a more streamlined, accurate, and user-centric management system if it implements the outlined technologies and emulates the best practices identified in other systems. This advancement will not only improve daily operations at KDSI but will also significantly contribute to the overall satisfaction and well-being of its employees, establishing the institution as a model for effective and modern military housing management.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

This chapter meticulously describes the systematic approach and methodologies used in developing the KDSI House Management System. A detailed description of the methodology used reveals the strategic decisions made to ensure an effective and efficient development process. This methodology not only accommodates the project's dynamic needs but also allows for detailed documentation and justification of each phase, emphasizing its importance in meeting the project's objectives. The systematic approach is fundamental, guiding the project from concept to implementation while also addressing the comprehensive system requirements required for a complete understanding of the project scope.

3.2 Methodology Choice and Justification

The Agile methodology was chosen for the development of the KDSI House Management System due to its inherent flexibility, iterative nature, and emphasis on continuous improvement, all of which are well-suited to KD Sultan Ismail's dynamic environment. This methodology promotes adaptive planning, evolutionary development, early delivery, and continuous improvement, fostering a quick and flexible response to change, which is critical to the project's success. Agile methodology fosters a participatory environment for project stakeholders, ensuring that the development process is transparent and that the results closely match stakeholder expectations and requirements. Agile ensures that the system evolves responsively and robustly by incorporating regular stakeholder feedback into the development process. Effectively managing potential risks and ensuring that the project stays on track with its objectives.

To help the reader better understand the Agile Methodology, figure 3.1 below depicts the key phases involved:

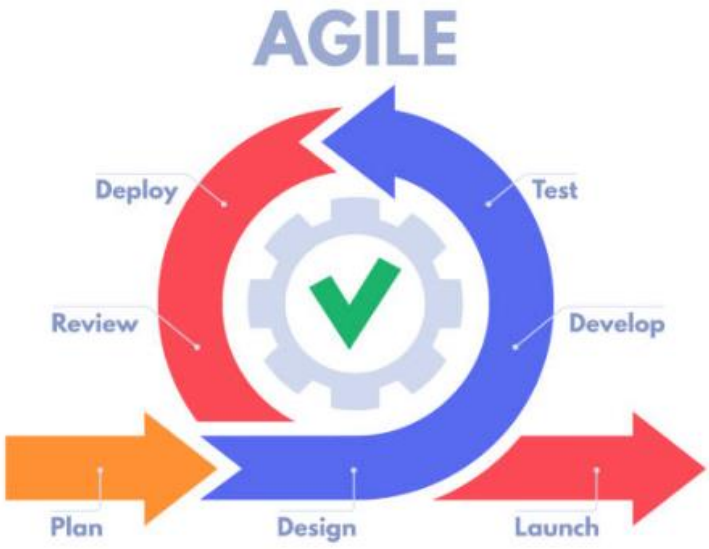


Figure 3.1 Agile Methodology Phases (Source: Peterson, 2018)

3.3 Phases of the Chosen Methodology

The KDSI House Management System was meticulously planned into distinct phases, each tailored to facilitate specific aspects of the development process. Initially, the Requirement Gathering phase was critical in developing a thorough understanding of the needs and challenges unique to KD Sultan Ismail's staff and administrators. This phase included in-depth interactions with potential users as well as a thorough analysis of existing manual processes to identify flaws and areas for improvement. Following that, the System Design phase was critical in developing detailed design documents that formed the basis for the system's architecture and user interface. This phase was critical in moving from conceptual designs to actionable blueprints that guided future development efforts. The Development phase saw these designs brought to life through iterative sprints that focused on creating functional, testable system components while ensuring they met the rigorous standards outlined in the initial designs. Testing was thorough, including unit, integration, and user acceptance tests to ensure a strong and dependable system. Following development, the Deployment phase placed the system in a controlled operational environment where it could be tested in real-world

scenarios. Finally, the Maintenance and Evaluation phase ensured that the system continued to function optimally after deployment, providing ongoing support and regular enhancements based on user feedback and system performance analysis.

3.4 Technology Used Description

The technologies used in the development of the KDSI House Management System were chosen based on their proven reliability and effectiveness in supporting complex web applications. The system's server-side components were built around PHP and MySQL, which allowed for robust data management and dynamic content generation. Laravel and Bootstrap were used to improve the system's usability and aesthetic appeal, making the interface intuitive and responsive. Furthermore, React and Node.js were used to handle real-time data processing and improve the user interface's interactivity, resulting in a system that is not only functional but also engaging to its users. Firebase's inclusion enabled real-time data synchronization and user authentication, which are critical for maintaining data integrity and security.

3.5 System Requirement Analysis

The system requirement analysis for the KDSI Staff House Management System involves identifying and specifying the software, hardware, network, and security requirements required to ensure the system's efficiency and effectiveness. This section provides a thorough examination of these requirements.

3.5.1 Software Requirements

The software requirements describe the various software components and tools required for the system's development and operation. The system requires a dependable operating system to host the web application, such as Windows Server 2019 or Ubuntu 20.04 LTS. A web server, such as Apache HTTP Server or Nginx, is needed to serve web pages to users. To store and manage data, the system requires a robust Database Management System (DBMS), which can be MySQL or PostgreSQL.

Backend and frontend development use programming languages such as PHP for server-side scripting and JavaScript for client-side scripting, respectively. Frameworks such as Laravel (a PHP framework) and React (a JavaScript library) are required to simplify development and provide structure. Integrated Development Environments (IDEs) and code editors, such as Visual Studio Code or PHPStorm, are also required for writing and debugging code.

3.5.2 Hardware Requirements

The hardware requirements specify the physical components needed to support the system. The web application and database must be hosted on servers that have at least an Intel i5 processor, 16GB of RAM, and 512GB of SSD storage. To access the system, end users will require devices with at least an Intel i3 processor, 4GB RAM, and 128GB HDD storage. A dependable network infrastructure, including a high-speed internet connection, routers, and switches, is also required to ensure connectivity and accessibility.

3.5.3 Network Requirements

The network requirements specify the network components and configurations needed to support the system's operations. Accessing the web application requires a high-speed internet connection of at least 100 Mbps. Firewalls and VPNs are examples of network security measures that are necessary to protect data and prevent unauthorized access.

3.5.4 Security Requirements

The security requirements outline the steps required to protect the system and its data from threats. Implementing secure login processes and role-based access control, such as JWT (JSON Web Tokens) and OAuth, ensures authentication and authorization. Encrypting sensitive data to protect it during transmission and storage, using tools like SSL/TLS for data in transit and AES for data at rest, is essential. Regular security audits to identify and address vulnerabilities, using security audit software and

penetration testing tools, are required. Implementing backup and recovery solutions to ensure data integrity and availability, such as backup software and cloud storage solutions, is also critical.

By addressing these system requirements, the KDSI Staff House Management System can be designed to run efficiently, securely, and effectively while meeting the needs and expectations of its users.

3.6 Chapter Summary

This chapter provided a thorough overview of the methodologies and technologies employed in the creation of the KDSI Staff House Management System. The Agile methodology was chosen due to its adaptability, iterative nature, and emphasis on continuous improvement. Agile was chosen for its ability to adapt to change, support iterative development, facilitate stakeholder collaboration, and effectively manage risks.

The Agile methodology was implemented in several phases, including requirement gathering, system design, development, testing, deployment, and maintenance and evaluation. Each phase was described in detail, including the expected outcomes, activities, and specific tools and technologies used.

The system's development technologies were also discussed, including PHP for server-side scripting, MySQL for database management, Laravel for framework support, Bootstrap for front-end design, React for building user interfaces, Node.js for server-side scripting, and Firebase for real-time database management and authentication.

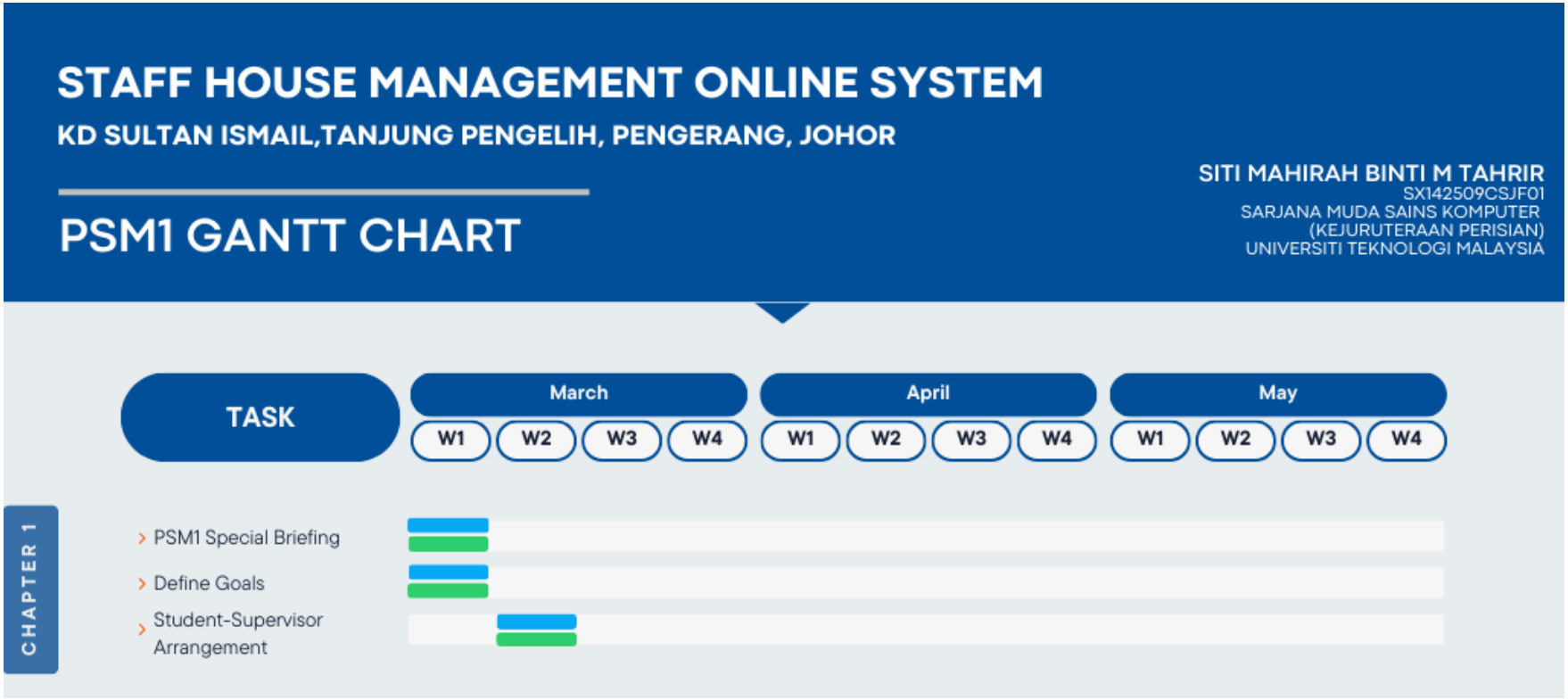
Furthermore, system requirements were examined, including software, hardware, network, and security requirements. Software requirements included operating systems, web servers, database management systems, programming languages, frameworks, and development tools. Hardware requirements specify the server and client hardware specifications. Network requirements emphasized the importance of high-speed internet connectivity and network security measures. Security

requirements emphasized the importance of authentication, data encryption, regular security audits, and backup and recovery solutions.

Overall, this chapter laid the groundwork for the KDSI Staff House Management System by outlining the methodologies, technologies, and requirements needed to create an efficient, secure, and user-friendly application.

Project Timeline and Scheduling

To ensure that the project is completed efficiently and on time, a detailed Gantt chart has been created. The Gantt chart shows the key activities, milestones, and timelines for the KDSI Staff House Management Online System.




CHAPTER 2
Start Literature Review
Submit Proposal to Supervisor
Supervisor Approval
Submit Approved Proposal
CHAPTER 3
Start System Design (High-Level)
Student-Supervisor Meetings
CHAPTER 4
Start Use Case Diagram
Start Sequence Diagrams
Start User Interface Design
CHAPTER 5
Documentation
Final Report Preparation
Supervisor Review & Feedback
Gantt Chart for PSM2
Final adjustments based on feedback
Submission of PSM1
Plan
Done
In Progress

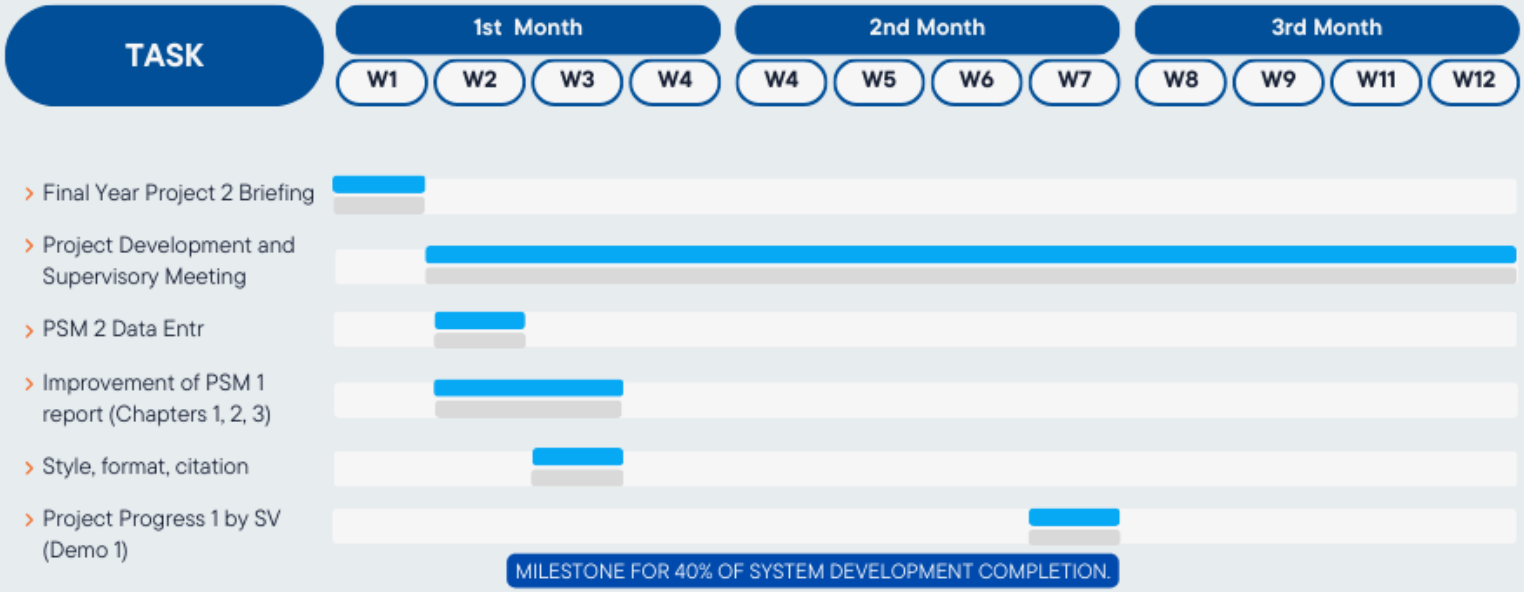
STAFF HOUSE MANAGEMENT ONLINE SYSTEM

KD SULTAN ISMAIL,TANJUNG PENGELIH, PENGERANG, JOHOR

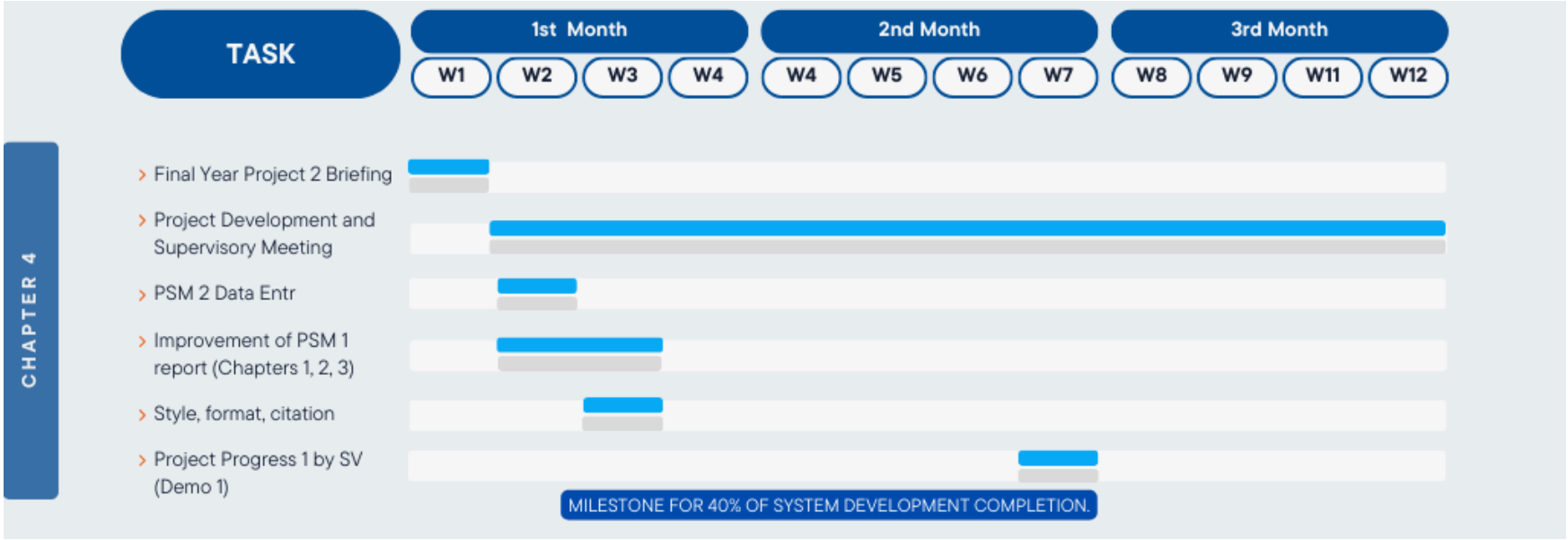
PSM2 GANTT CHART

SITI MAHIRAH BINTI M TAHRIR
SX142509CSJF01
SARJANA MUDA SAINS KOMPUTER
(KEJURUTERAAN PERISIAN)
UNIVERSITI TEKNOLOGI MALAYSIA

CHAPTER 4



CHAPTER 4


TASK
1st Month
2nd Month
3rd Month
W1
W2
W3
W4
W4
W5
W6
W7
W8
W9
W11
W12
Final Year Project 2 Briefing
Project Development and Supervisory Meeting
PSM 2 Data Entr
Improvement of PSM 1 report (Chapters 1, 2, 3)
Style, format, citation
Project Progress 1 by SV (Demo 1)
MILESTONE FOR 40% OF SYSTEM DEVELOPMENT COMPLETION.

CHAPTER 6


TASK
3rd Month
4th Month
5th Month
W8
W9
W10
W11
W12
W13
W14
W15
W16
W17
W18
W19
Completeness of Chapter 6 (Discussion and Conclusion)
Final formatting adjustments to the thesis.
Plagiarism issue (Turnitin)
Documents approval by supervisor
Perform Turnitin
Submit all related documents to SV
Presentation Eligibility
Submit all related documents to Coordinator
Project 2 Presentation Preparation
Prepare all document for Examiner
Project 2 Presentation
Final Thesis correction and submission
Plan
Done
In Progress

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

This chapter delves into the core analytical and architectural components of the KDSI Staff House Management System, highlighting the meticulous processes involved in tailoring the system's development to KD Sultan Ismail's specific requirements. The requirement analysis and design phase, which is the foundation of effective system development, lays the groundwork for a strong, efficient, and user-centric solution. This chapter thoroughly examines the system's specifications, combining user requirements into a cohesive architectural blueprint that guides all subsequent development activities.

This chapter begins with requirement analysis and describes the methods used at KDSI to collect, verify, and refine system requirements from a variety of stakeholders, including end-users, administrative staff, and IT personnel. This process ensures that the system is not only functional but also meets the real-world needs and expectations of its users. The chapter delves further into the transition from abstract requirements to concrete specifications that guide the development process, emphasizing how these requirements are translated into detailed functional and non-functional specifications.

Following the requirement specification, the chapter takes a thorough look at the system's design. This section describes the architectural planning and decision-making processes that define the system's structural framework. It includes in-depth discussions about database design, which covers how data is structured, stored, and accessed to ensure integrity, efficiency, and security. Interface design is also extensively covered, demonstrating how to create intuitive and tailored user interfaces that improve user interaction and satisfaction.

In addition to database and interface design, this chapter discusses overall project design. This section brings together all design elements into a cohesive strategy that ensures the system's architecture supports its intended functionalities while remaining adaptable and scalable. The incorporation of modern design principles and technologies is emphasized to show how they contribute to a more dynamic and responsive system.

The chapter concludes with a summary that highlights the importance of requirement analysis and system design in the creation of the KDSI Staff House Management System. It emphasizes how important the early stages of the project lifecycle are in setting the course for a successful implementation, ensuring that the system not only meets KD Sultan Ismail's current needs but also has the flexibility to evolve in response to future demands.

This chapter provides a clear and detailed roadmap for developing a system capable of handling the complexities of house management at KD Sultan Ismail, while also improving operational efficiency and user experience.

4.2 Requirement Analysis

Requirement analysis is a critical phase in the development of the KDSI Staff House Management System, as it is the first step in understanding and documenting the needs and expectations of all stakeholders. This process entails systematically gathering, organizing, and analyzing requirements to ensure that the final system is in line with KD Sultan Ismail's operational objectives and challenges. This meticulous analysis allows the project to effectively translate user demands into a set of well-defined functional and non-functional requirements.

Functional Requirements:

1. The capability to add, amend, and remove user data is known as user management. Distinct access levels should be assigned to distinct roles (worker, visitor, and admin).

2. Manage the details of your home, including any updates on its status (empty, occupied, or undergoing maintenance).

3. Maintenance Management: Track and submit maintenance requests, with the option to attach photos for more accurate reporting.

4. Guest management includes verifying visits by guests and managing check-ins and check-outs.

5. Dashboard and Reporting: Up-to-date dashboard that shows summaries of guest visits, maintenance requests, and house statuses. Create reports every month for analysis.

Non-Functional Requirements:

1. Usability: The interface of the system should be simple to use and intuitive.

2. Security: Put strong security measures in place, such as role-based access control and user authentication.

3. Performance: The system should respond to user activities quickly and efficiently.

4. Scalability: Without compromising performance, the system must be scalable to handle more users and data.

5. Maintaining data accuracy and consistency throughout the system is known as data integrity.

4.3 Project Design

The project design section outlines the overarching architecture and detailed structural design of the KDSI Staff House Management System. This critical phase translates the refined requirements into a blueprint that dictates how the system will be built, focusing on ensuring robust functionality, optimal performance, and high usability.

4.3.1 System Architecture

The KDSI Staff House Management System's architecture is modular and scalable, allowing it to meet both current and future requirements. The system is designed with a three-tier architecture that separates the presentation, business logic, and data access layers, resulting in a clear separation of concerns and easier maintenance.

1. The Presentation Layer manages all user interface and browser communication logic, ensuring an intuitive and responsive user experience. It is built with React.js to create dynamic and interactive UI components that improve user interaction and satisfaction.

2. Business Logic Layer: This layer is responsible for processing the system's core functionalities, including user management, house allocation, maintenance requests, and guest management. Laravel, a robust PHP framework, is used here because of its elegant syntax and powerful features that make tasks like routing, sessions, and authentication easier.

3. Data Access Layer: This layer communicates with the MySQL database and manages all data persistence mechanisms. It ensures data integrity and consistency throughout the system, allowing for efficient data retrieval and storage operations.

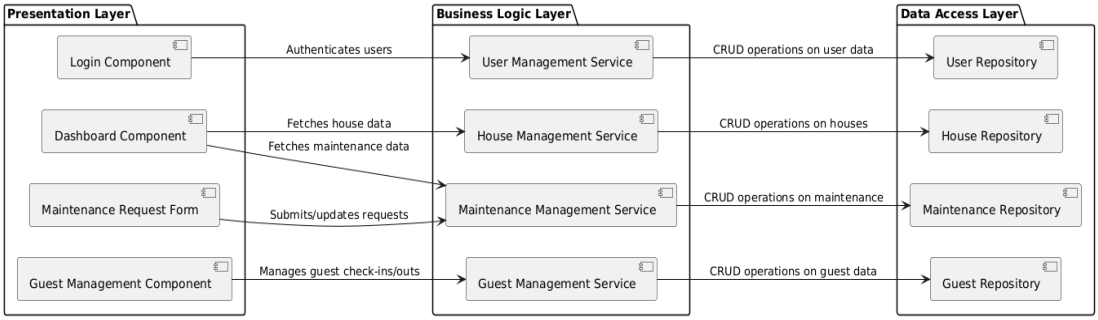


Figure 4.1 Component Model of the KDSI Staff House Management Online System

The component model of the KDSI Staff House Management System is structured into a three-tier architecture, encompassing the Presentation Layer, Business Logic Layer, and Data Access Layer, each pivotal to the system's operations. The Presentation Layer is crucial for user interaction, consisting of various components designed to facilitate the interface between the user and the system. This includes the Login Component, which manages user authentication; the Dashboard Component, which displays real-time data on house and maintenance status; the Maintenance Request Form, which enables users to submit and update maintenance requests; and the Guest Management Component, responsible for handling guest check-ins and check-outs.

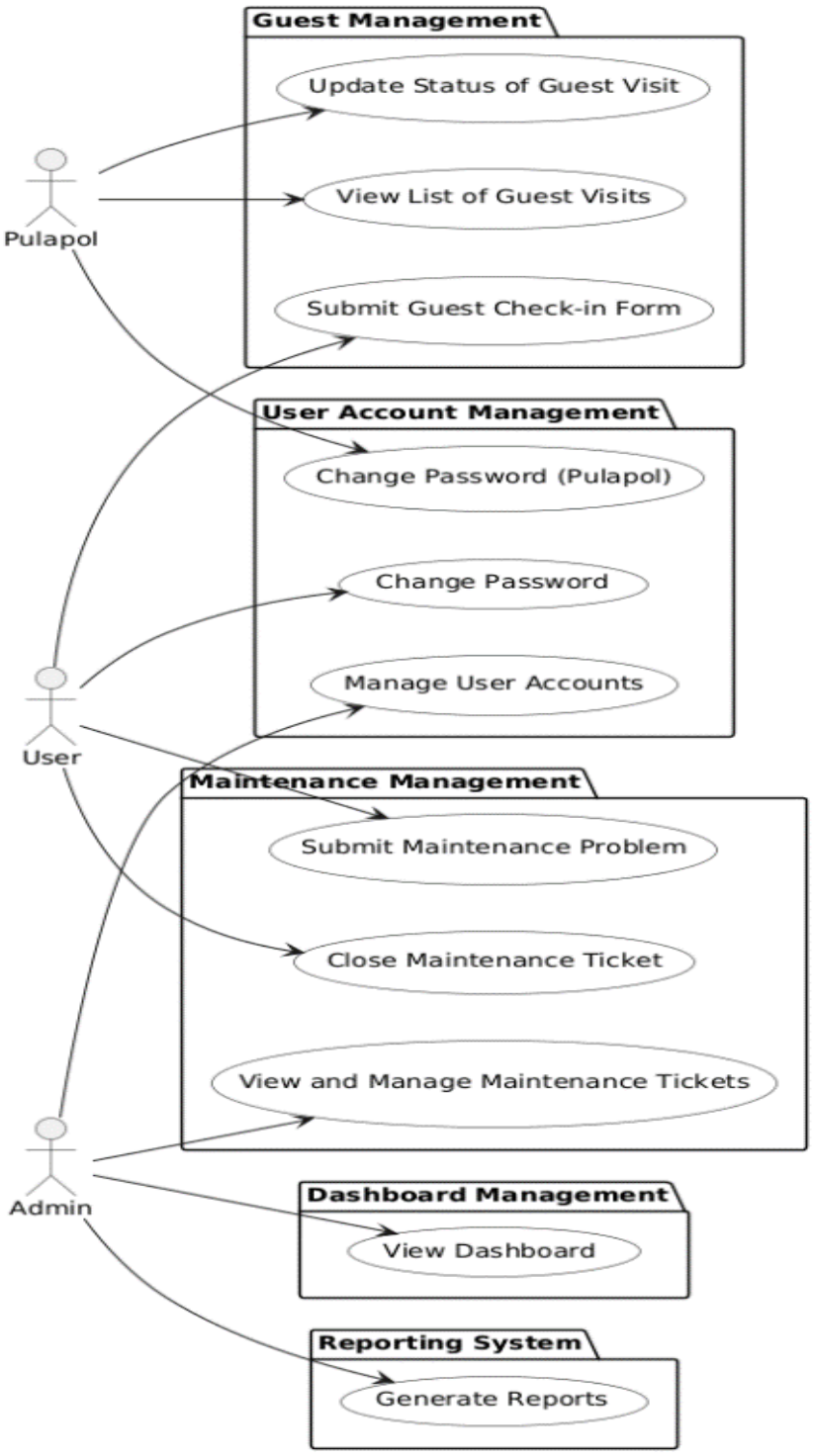
Beneath the user-facing layer, the Business Logic Layer processes data and implements business rules. This layer includes the User Management Service which handles all user-related functionalities, the House Management Service which manages information related to staff housing, the Maintenance Management Service which oversees maintenance operations, and the Guest Management Service which manages all aspects of guest visits. These services ensure that the system's business rules are adhered to and that data flows efficiently between the user interface and the data storage systems.

Supporting these operations, the Data Access Layer ensures robust data management and retrieval. It includes the User Repository for storing and retrieving user data, the House Repository for managing house-related data, the Maintenance Repository for handling data associated with maintenance requests, and the Guest

Repository for managing data concerning guests. This layer is essential for the persistence, security, and integrity of data across the system.

Interactions between these layers are fundamental to the system's functionality. Components in the Presentation Layer interact with the Business Logic Layer to send and receive the data necessary for user operations. The Business Logic Layer, in turn, communicates with the Data Access Layer to perform CRUD (Create, Read, Update, and Delete) operations on the database, reflecting changes made through the user interface. This interconnected structure not only provides a clear overview of the system’s high-level organization but also highlights the distribution of responsibilities among different subsystems. The collaborative nature of the system components collectively supports the robust functionality of the KDSI Staff House Management System, ensuring that it meets the operational needs efficiently and effectively.

4.3.2 Use Case Diagram

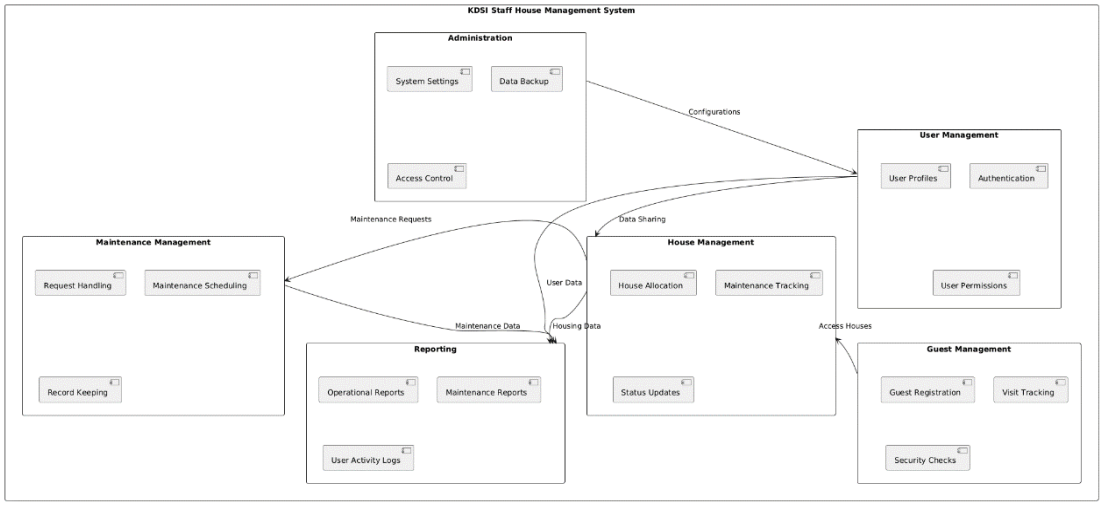


0

1

Figure 4.2 Use Case Diagram of KDSI Staff House Management System

4.3.3 Use Case Diagram



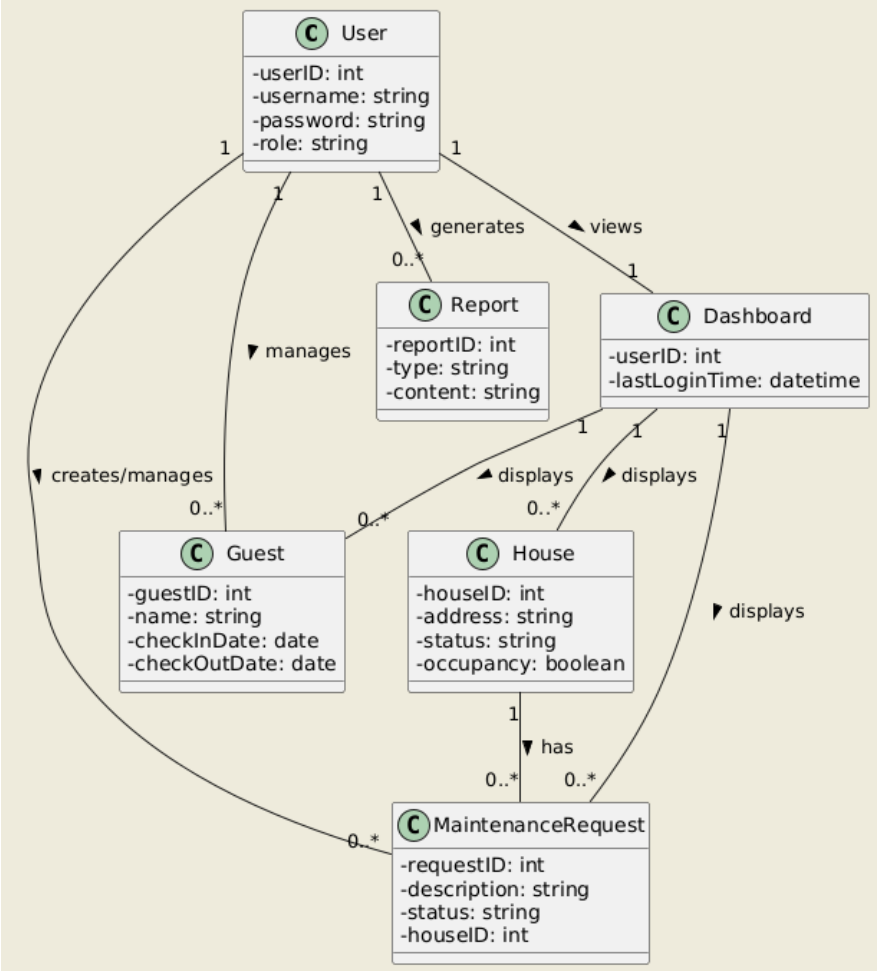
2

Figure 4.3 Subsystem Diagram of KDSI Staff House Management System

The KDSI Staff House Management System's subsystem diagram depicts a structured overview of its main functionalities and interactions. Each subsystem is assigned a unique package that represents the system's division of responsibilities and workflow. The User Management Subsystem is in charge of user authentication, profile management, and access controls, which ensure secure and personalized user experiences. The House Management Subsystem is in charge of house allocations, status updates, and occupancy monitoring, all of which are critical for effective house management.

Furthermore, the Maintenance Management Subsystem is responsible for managing all aspects of maintenance requests, from scheduling to actual repairs, ensuring that maintenance issues are resolved quickly and efficiently. The Guest Management Subsystem manages visitor registrations and tracking, which improves facility security by controlling guest access. The Reporting Subsystem generates operational reports, audit logs, and performance reports, which provide critical insights and data for management decision-making.

4.3.4 Class Diagram



3

Figure 4.4 Class Diagram of KDSI Staff House Management System

The KDSI Staff House Management System class diagram provides a detailed structural view of the system's various classes and how they are interconnected. The diagram depicts the system's architecture by highlighting each class's roles and responsibilities, as well as the relationships between them, to ensure a thorough understanding of the system's functionality.

The User class is responsible for managing personal and authentication data, storing critical attributes such as the user ID, username, password, and role, and facilitating access control throughout the system. The House class catalogues properties,

encapsulating critical details such as house ID, location address, and occupancy status, all of which are essential for operational transparency and efficiency.

Maintenance activities are encapsulated in the Maintenance Request class, which tracks each request by ID, status, and detailed description, linking directly to the associated property to simplify repair and upkeep tasks. Similarly, the Guest class keeps track of visitors, storing identities as well as check-in and check-out times, to ensure that guest visits are properly managed.

The Report class is responsible for creating a variety of reports, including operational, audit, and performance analyses, using data gathered throughout the system to support administrative decision-making and compliance monitoring. In parallel, the Dashboard class is critical in that it aggregates and displays real-time data from multiple sources such as house statuses, maintenance updates, and guest logs, providing users with a dynamic and interactive interface to effectively monitor and control system activities.

Interactions between these classes are represented by arrows indicating data flow, such as how user actions trigger maintenance requests or guest registrations, and how these interactions are reflected across the system's modules. This interconnected setup not only improves the functionality of the KDSI Staff House Management System but also ensures that it meets the operational needs and management objectives of KD Sultan Ismail's housing administration.

4.3.5 Sequence Diagram

UC001: Use Case Admin Views Dashboard

Table 4.1 Use Case Admin Views Dashboard

Use case: Admin Views Dashboard
ID: UC001
Actors: Admin
Pre-conditions: Admin must be logged into the system.
The flow of events: 1. Admin logs into the system. 2. Admin navigates to the dashboard. 3. The system retrieves house status data from the database. 4. The system displays the dashboard with house status data to the admin.
Post-conditions: Admin views the status of all houses and maintenance tickets on the dashboard.

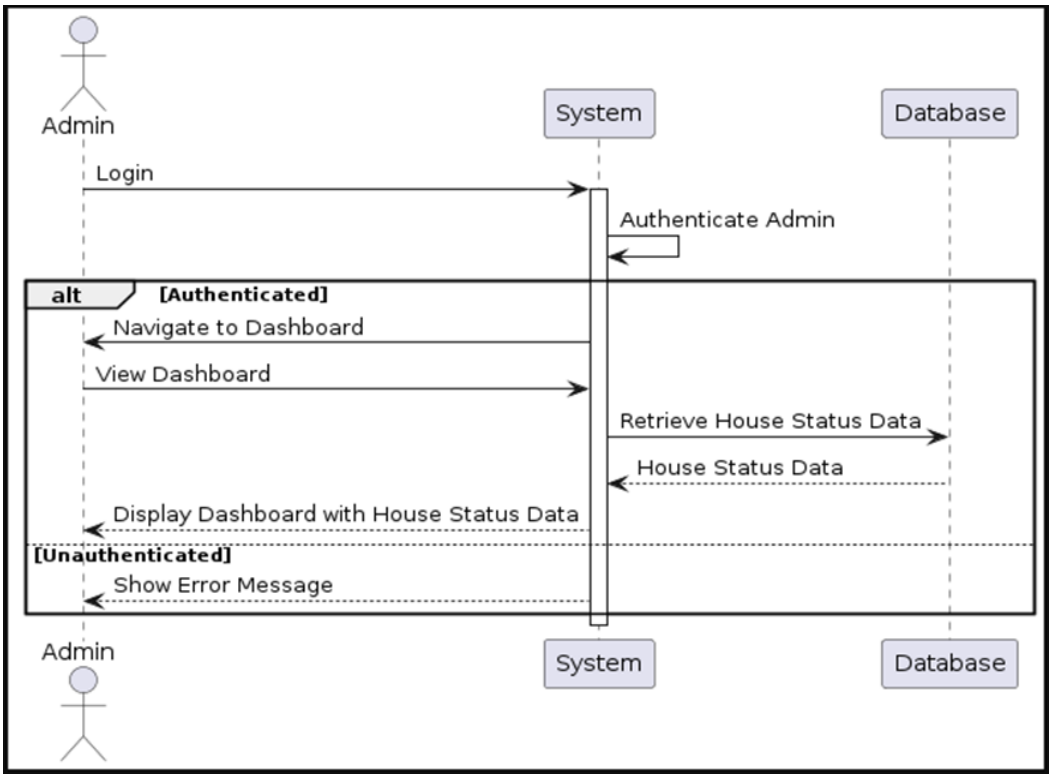


Figure 4.5 Sequence Diagram for Admin Views Dashboard

UC002: Use Case User Submits Maintenance Problem

Table 4.2 Use Case User Submits Maintenance Problem

Use case: User Submits Maintenance Problem
ID: UC002
Actors: User
Pre-conditions: The user must be logged into the system
The flow of events: 1. User logs into the system. 2. The user navigates to the maintenance form. 3. The user submits the maintenance problem.
Post-conditions:

The maintenance problem is submitted and recorded in the system.

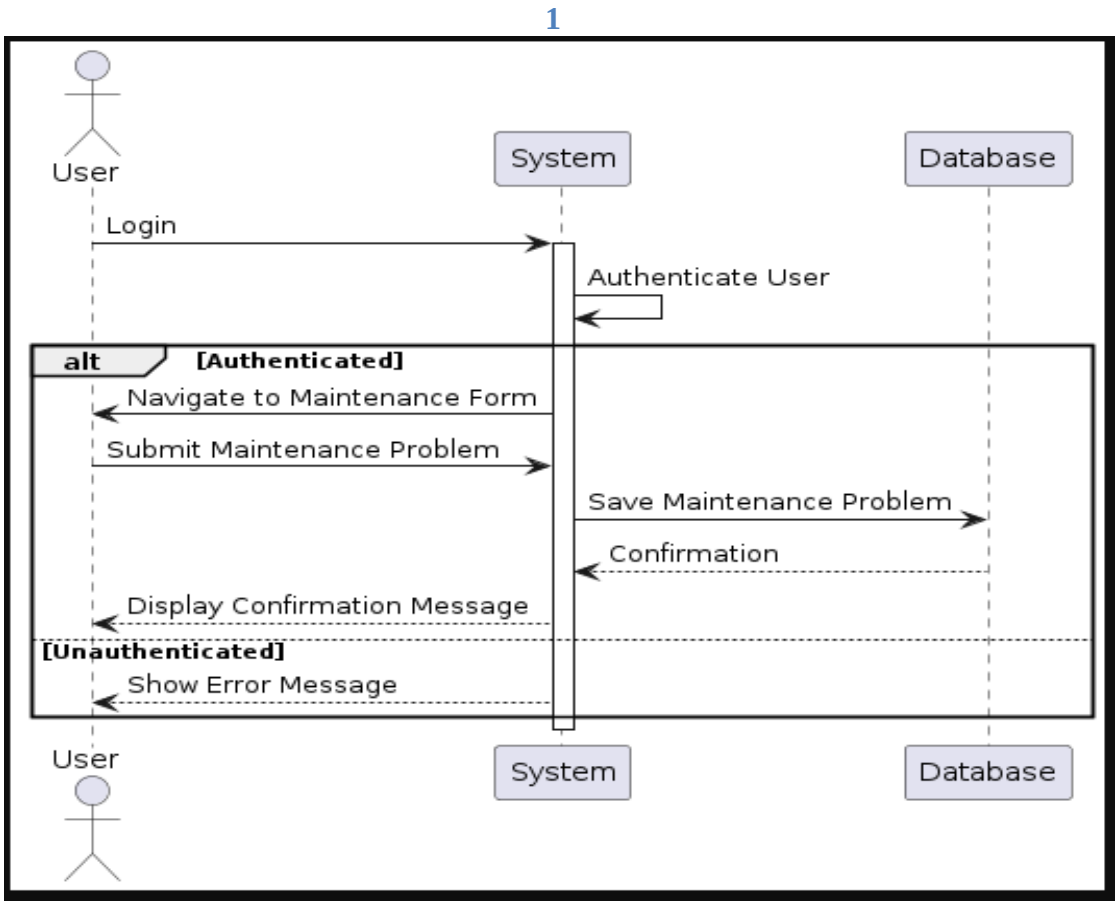


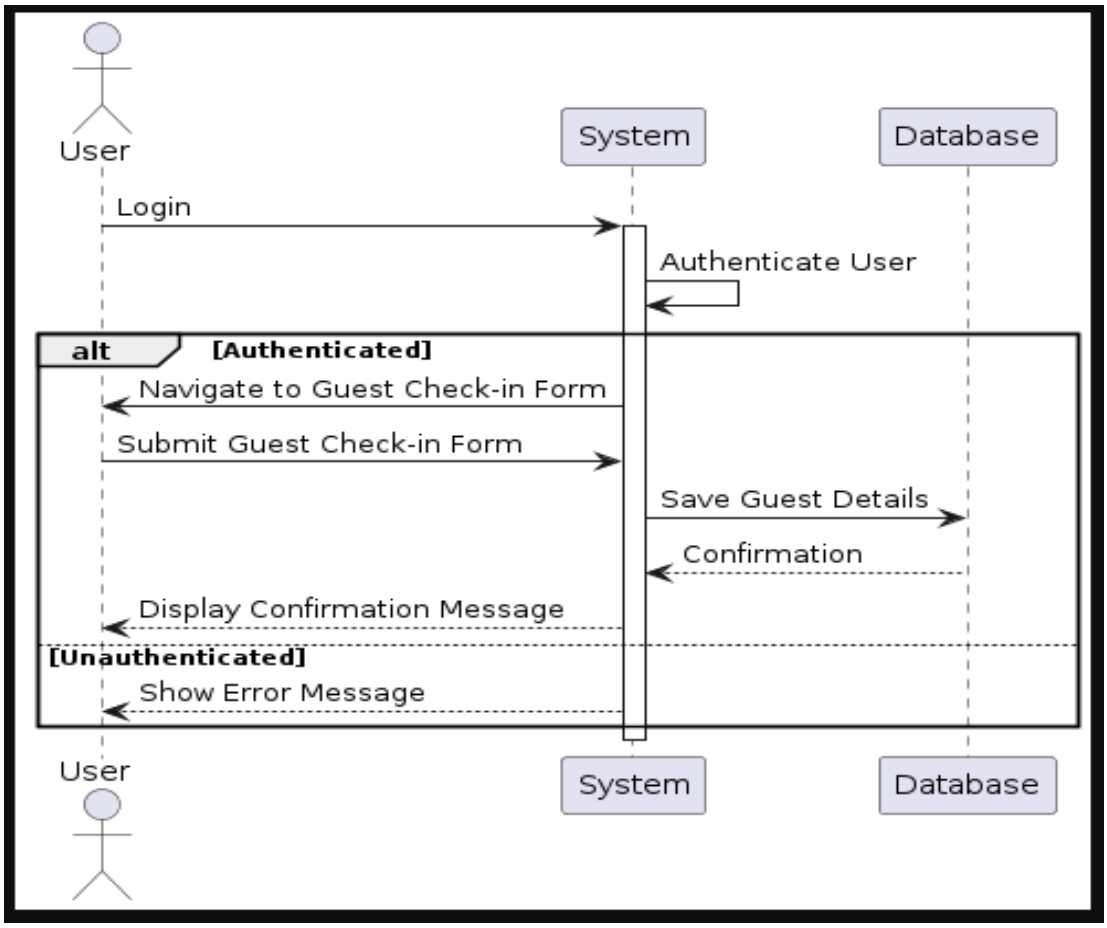
Figure 4.6 Sequence Diagram for User Submits Maintenance Problem

UC003: Use Case User Submits Guest Check-in Form

Table 4.3 Use Case User Submits Guest Check-in Form

2

Use case: User Submits Guest Check-in Form User
ID: UC003
Actors: User
Pre-conditions: User must be logged into the system.
The flow of events: 1. User logs into the system. 2. User navigates to the guest check-in form. 3. User submits the guest check-in form.
Post-conditions: The guest check-in details are submitted and recorded in the system.



4

Figure 4.7 Sequence Diagram for User Submits Guest Check-in Form

UC004: Use Case User Closes Ticket

Table 4.4 Use Case User Closes Ticket

Use case: User Closes Ticket
ID: UC004
Actors: User
Pre-conditions: User must be logged into the system.
The flow of events: 1. User logs into the system.

2. User navigates to the list of open maintenance tickets. 3. User selects a ticket to close. 4. User closes the ticket.
Post-conditions: The maintenance ticket is closed in the system.

5

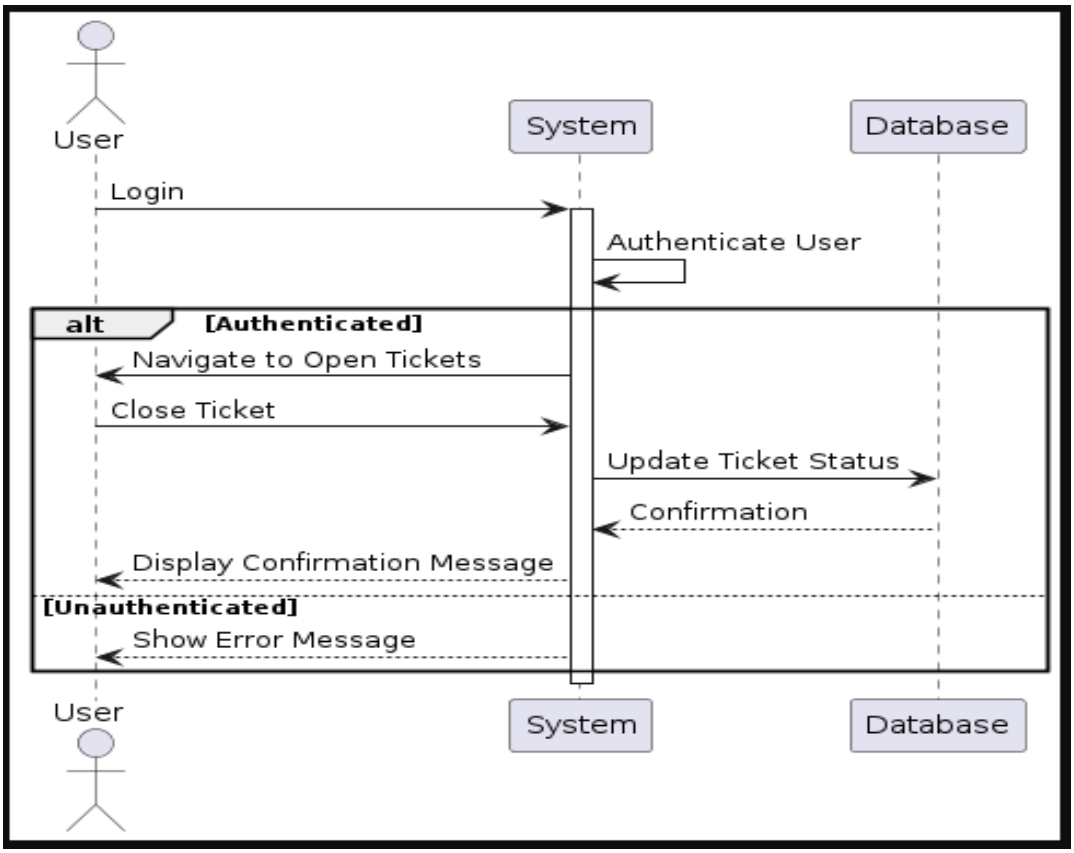


Figure 4.8 Sequence Diagram for User Closes Ticket

UC005: Use Case User Changes Password

Table 4.5 Use Case User Changes Password

Use case: User Changes Password
ID: UC005
Actors: User
Pre-conditions: User must be logged into the system.
The flow of events: 1. User logs into the system. 2. User navigates to the change password page. 3. User enters the current password and a new password. 4. User submits the password change request.
Post-conditions: The user's password is updated in the system.

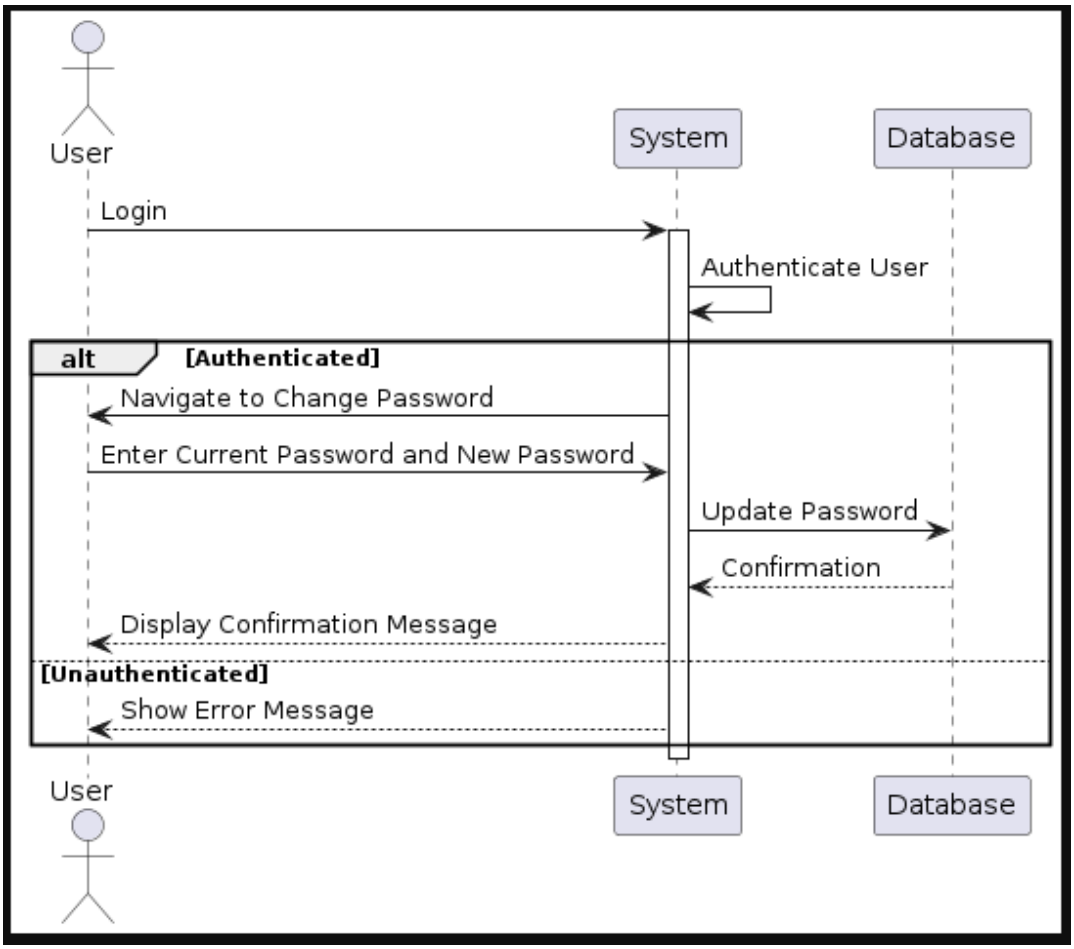


Figure 4.9 Sequence Diagram for User Changes Password

UC006: Use Case Admin Views and Manages Maintenance Tickets

Table 4.6 Use Case Admin Views and Manages Maintenance Tickets

Use case: Admin Views and Manages Maintenance Tickets
ID: UC006
Actors: Admin
Pre-conditions: Admin must be logged into the system.
The flow of events: 1. Admin logs into the system. 2. Admin navigates to the maintenance tickets page.

3. Admin views the list of maintenance tickets. 4. Admin can view, edit, or close individual maintenance tickets.
Post-conditions: Admin views and manages the maintenance tickets in the system.

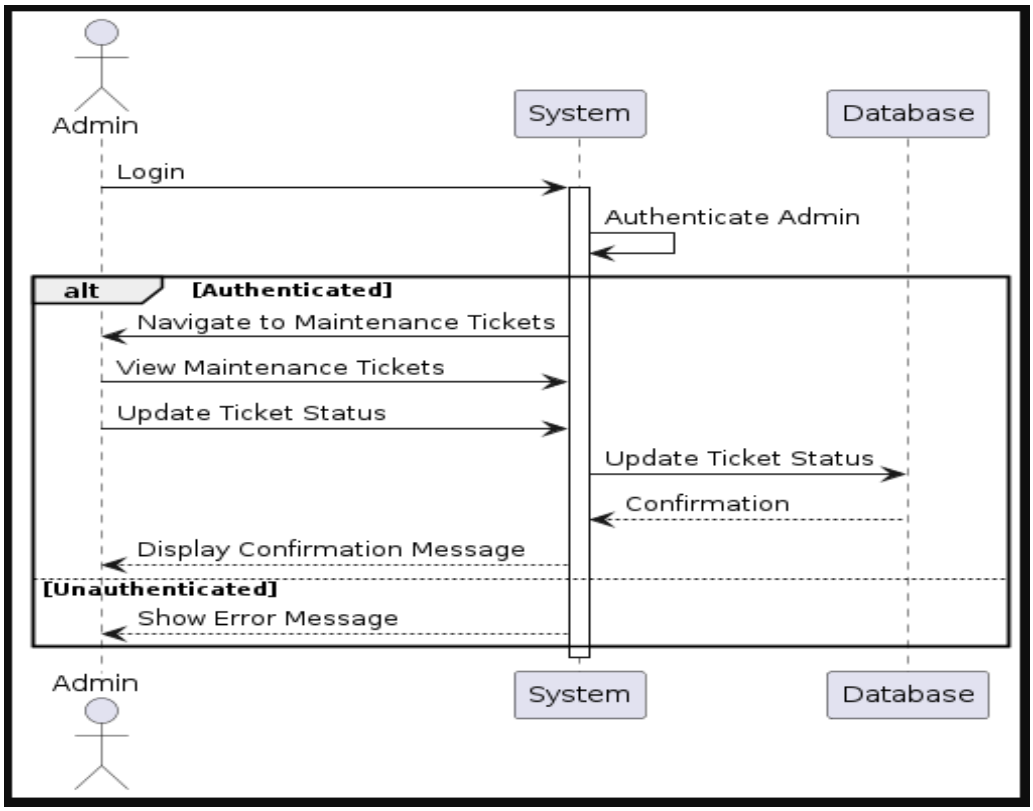


Figure 4.10 Sequence Diagram for Admin Views and Manages Maintenance Tickets

UC007: Use Case Pulapol Views List of Guest Visits

Table 4.7 Use Case Pulapol Views List of Guest Visits

Use case: Pulapol Views List of Guest Visits
ID: UC007
Actors: Pulapol
Pre-conditions: Pulapol must be logged into the system.
The flow of events: 1. Pulapol logs into the system. 2. Pulapol navigates to the list of guest visits. 3. Pulapol views the list of guest visits.
Post-conditions: Pulapol views the list of guest visits.

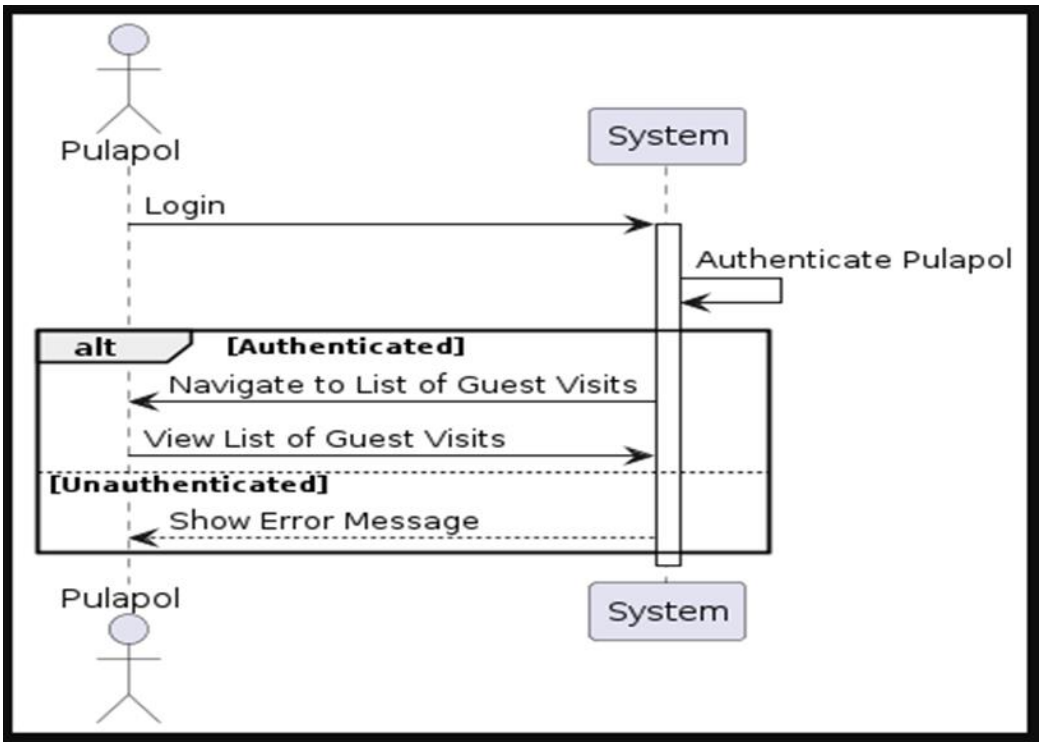


Figure 4.11 Sequence Diagram for Pulapol Views List of Guest Visits

UC008: Use Case Pulapol Updates Status of Guest Visit

Table 4.8 Use Case Pulapol Updates Status of Guest Visit

Use case: Pulapol Updates Status of Guest Visit
ID: UC008
Actors: Pulapol
Pre-conditions: Pulapol must be logged into the system.
The flow of events: 1. Pulapol logs into the system. 2. Pulapol navigates to the list of guest visits. 3. Pulapol selects a guest visit. 4. Pulapol updates the status of the guest visit.
Post-conditions:

The status of the guest visit is updated in the system.

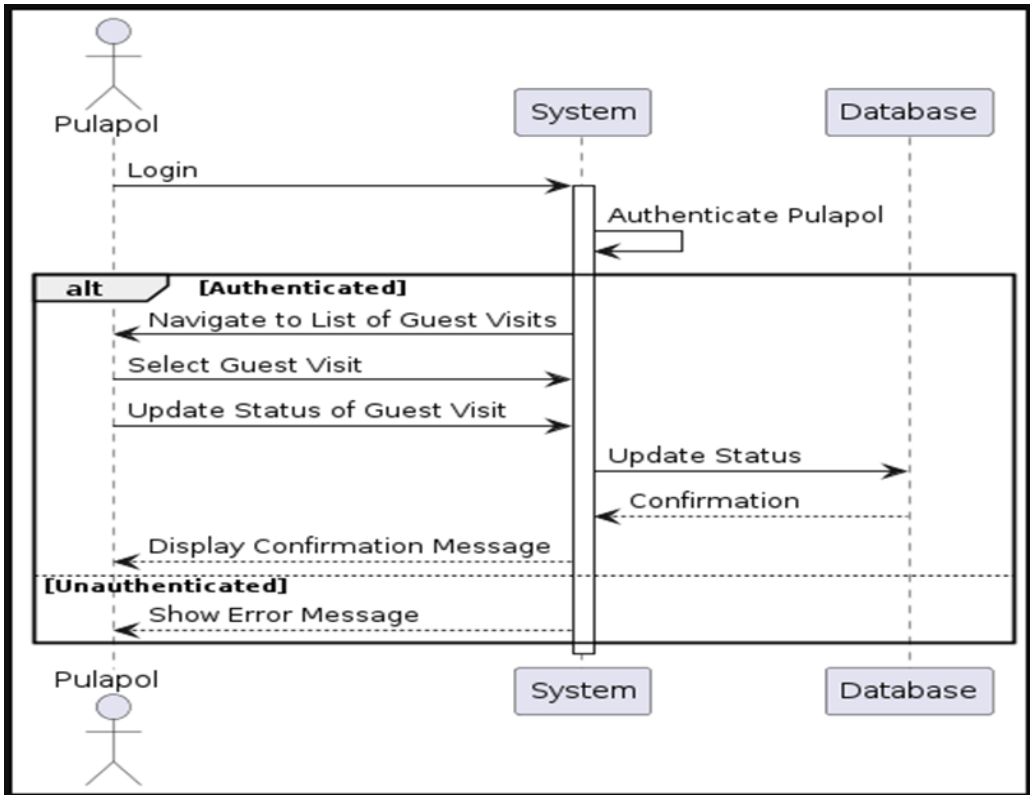


Figure 4.12 Sequence Diagram for Pulapol Updates Status of Guest Visit

UC009: Use Case Pulapol Changes Password

Table 4.9 Use Case Pulapol Changes Password

Use case: Pulapol Changes Password
ID: UC009
Actors: Pulapol
Pre-conditions: Pulapol must be logged into the system.

The flow of events: 1. Pulapol logs into the system. 2. Pulapol navigates to the change password page. 3. Pulapol enters the current password and a new password. 4. Pulapol submits the password change request.
Post-conditions: Pulapol's password is updated in the system.

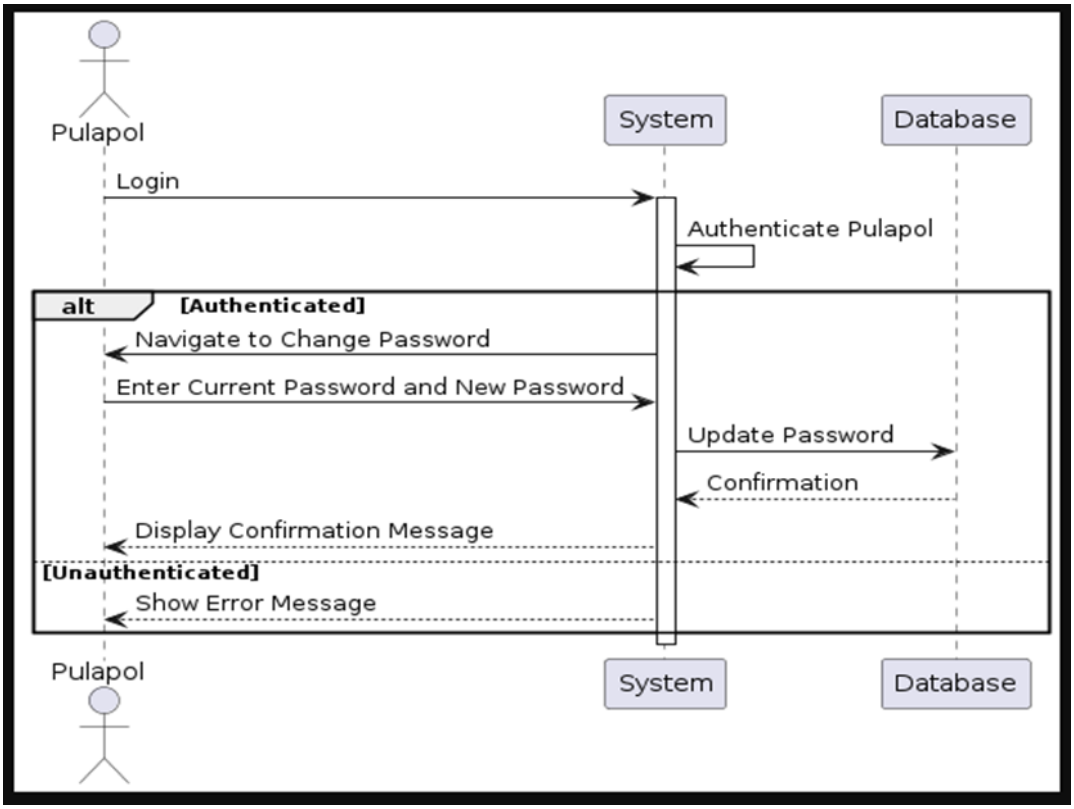


Figure 4.13 Sequence Diagram for Pulapol Changes Password

UC0010: Use Case Admin Manages User Accounts

Table 4.10 Use Case Admin Manages User Accounts

Use case: Admin Manages User Accounts
ID: UC0010
Actors: Admin
Pre-conditions: Admin must be logged into the system.
The flow of events: 1. Admin logs into the system. 2. Admin navigates to the user management section. 3. Admin selects to create, deactivate, or update a user account. 4. Admin submits the changes.
Post-conditions: User accounts are managed in the system.

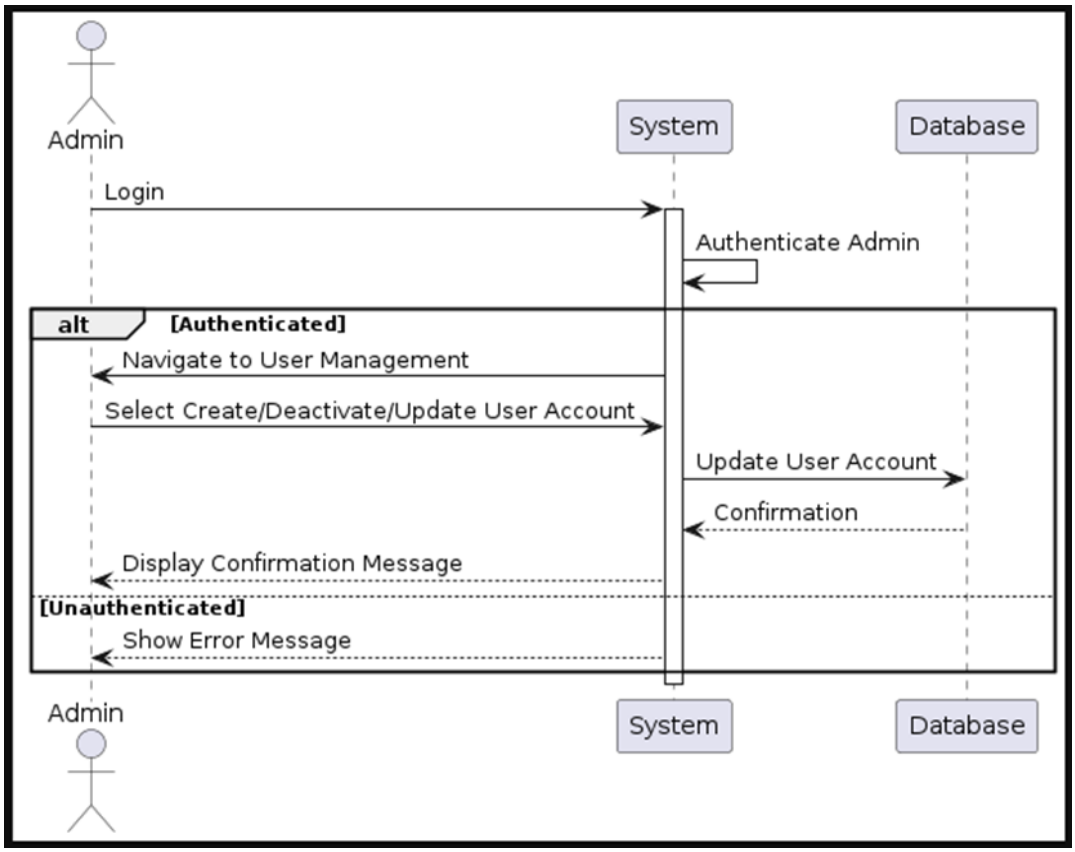


Figure 4.14 Sequence Diagram for Admin Manages User Accounts

UC0011: Use Case Admin Generates Reports

Table 4.11 Use Case Admin Generates Reports

Use case: Admin Generates Reports
ID: UC0011
Actors: Admin
Pre-conditions: Admin must be logged into the system.
The flow of events: 1. Admin logs into the system. 2. Admin navigates to the reports section.

3. Admin selects the type of report to generate. 4. Admin generates the report.
Post-conditions: Reports are generated and available for review. Pulapol's password is updated in the system.

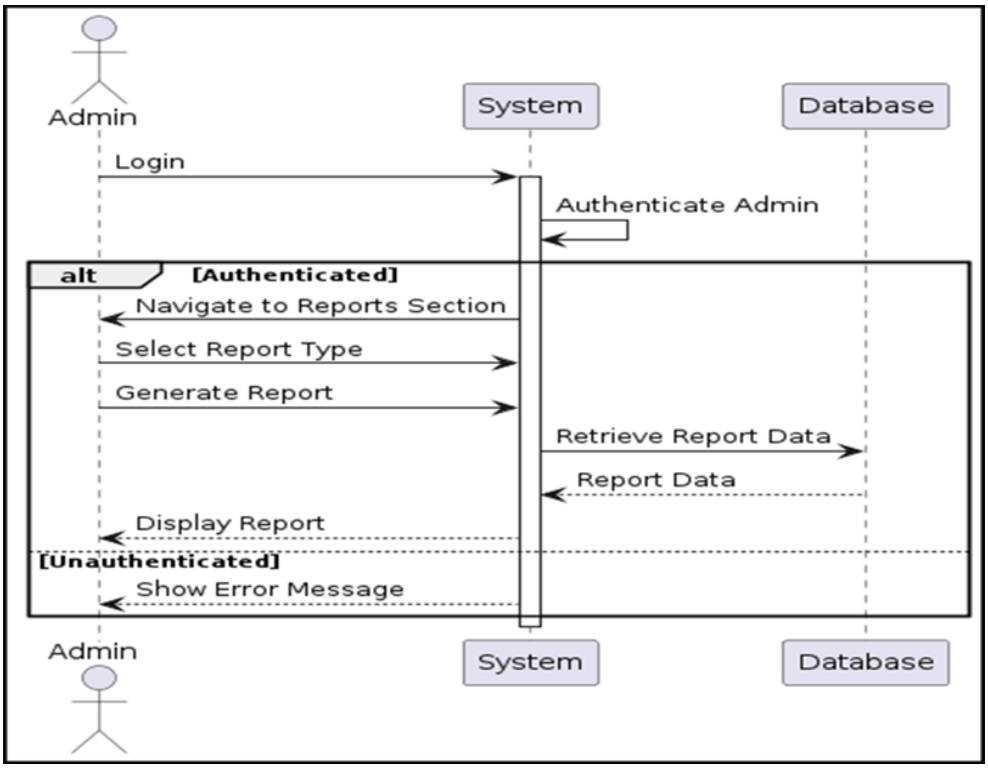


Figure 4.15 Sequence Diagram for Admin Generates Reports

4.4 Database Design

4.4.1 User Table

Stores information about all users in the system.

Table 4.12 User Table

Field Name	Data Type	Description
user_id	INT	Primary Key
username	VARCHAR (50)	User's username
password	VARCHAR (255)	User's password (hashed)
full_name	VARCHAR (100)	User's full name
email	VARCHAR (100)	User's email
phone_number	VARCHAR (15)	User's phone number
role	VARCHAR (20)	User's role (admin, staff, guest)

4.4.2 House Table

Stores information about houses managed by the system.

Table 4.13 House Table

Field Name	Data Type	Description
house_id	INT	Primary Key
house_number	VARCHAR (20)	House number
address	VARCHAR (255)	House address

status	VARCHAR (20)	House status (vacant, occupied, under maintenance, etc.)
tenant_id	INT	Foreign Key to User Table

4.4.3 Maintenance Request Table

Stores information about maintenance requests.

Table 4.14 Maintenance Request Table

Field Name	Data Type	Description
request_id	INT	Primary Key
house_id	INT	Foreign Key to House Table
user_id	INT	Foreign Key to User Table
description	TEXT	Maintenance request description
image_url	VARCHAR (255)	URL to the uploaded image
status	VARCHAR (20)	Status of the request (pending, in progress, completed)
created_at	DATETIME	Timestamp of the request creation
updated_at	DATETIME	Timestamp of the last update

4.4.4 Guest Check-In/Check-Out Table

Stores information about guest check-ins and check-outs.

Table 4.15 Guest Check-In/Check-Out Table

Field Name	Data Type	Description
visit_id	INT	Primary Key
guest_name	VARCHAR (100)	Guest's name
guest_id	INT	Foreign Key to User Table
house_id	INT	Foreign Key to House Table
check_in_time	DATETIME	Check-in timestamp
check_out_time	DATETIME	Check-out timestamp
verification_status	VARCHAR (20)	Verification status (verified, not verified)

4.4.5 Dashboard Table

Stores summarized information for the dashboard.

Table 4.16 Dashboard Table

Field Name	Data Type	Description
dashboard_id	INT	Primary Key
total_houses	INT	Total number of houses
vacant_houses	INT	Number of vacant houses
occupied_houses	INT	Number of occupied houses
under_maintenance_houses	INT	Number of houses under maintenance
maintenance_requests	INT	Number of maintenance requests
guest_visits	INT	Number of guest visits
report_date	DATE	Date of the report

4.4.6 Activity Log Table

Stores logs of activities performed in the system.

Table 4.17 Activity Log Table

Field Name	Data Type	Description
log_id	INT	Primary Key
user_id	INT	Foreign Key to User Table
action	VARCHAR (255)	Description of the action
timestamp	DATETIME	Timestamp of the action

4.5 Interface Design

The goal of interface design is to create a system that is easy to use and intuitive. Important interfaces consist of:

1. **Login Page:** Allows users to authenticate and access the system based on their roles.

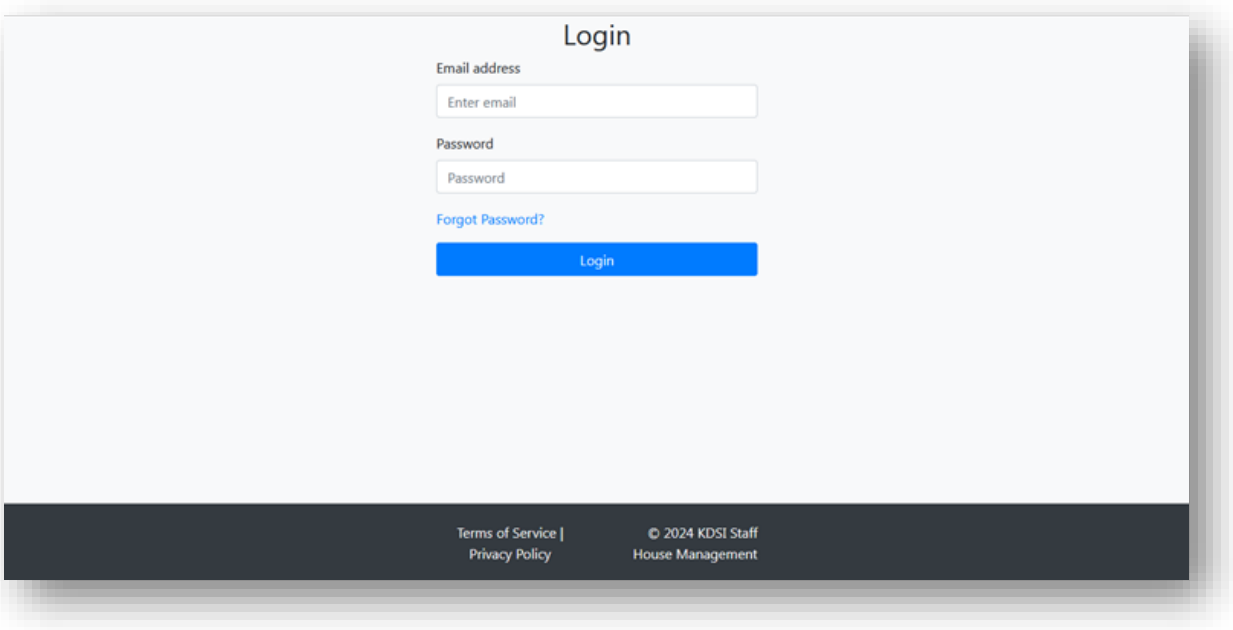


Figure 4.16 Login Page Interface

2. **Admin Dashboard:** Provides an overview of the system's status, including house details, maintenance requests, and guest visits.

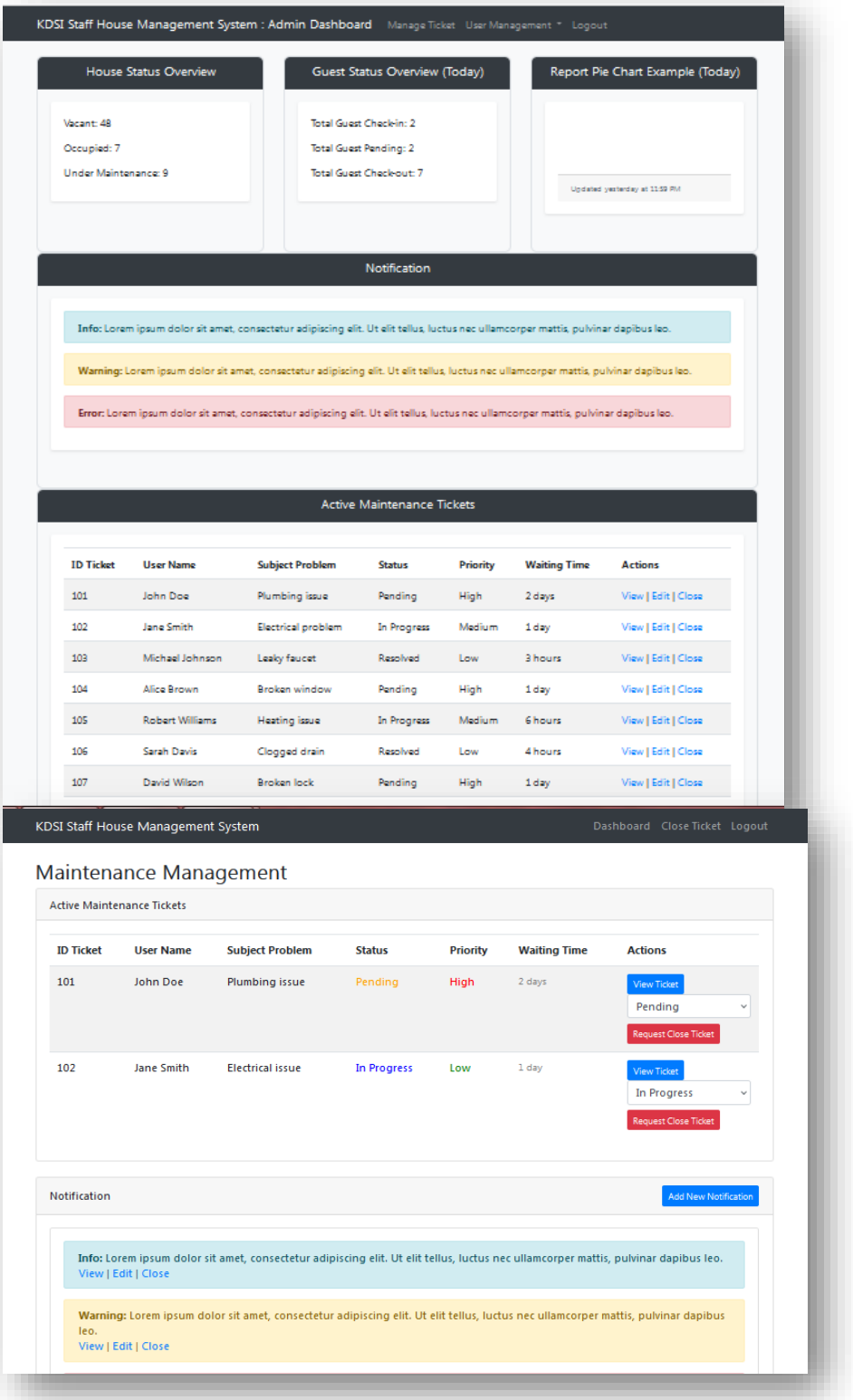
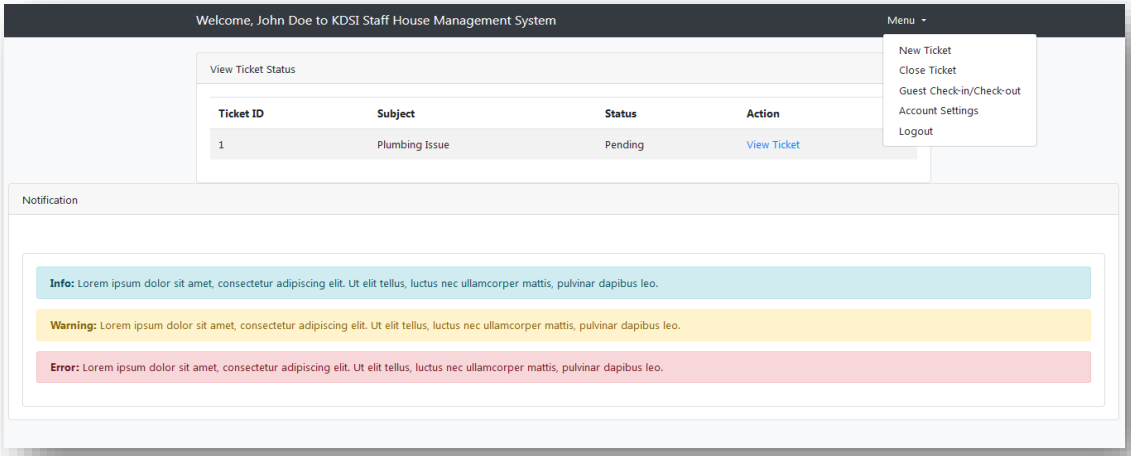


Figure 4.17 Admin Dashboard Interface

3. **User Dashboard:** Displays personalized information for logged-in users,



such as their assigned houses and submitted maintenance requests.

Figure 4.18 User Dashboard Interface

4. **New Maintenance Request Form**: Allows users to submit new maintenance requests with image uploads.

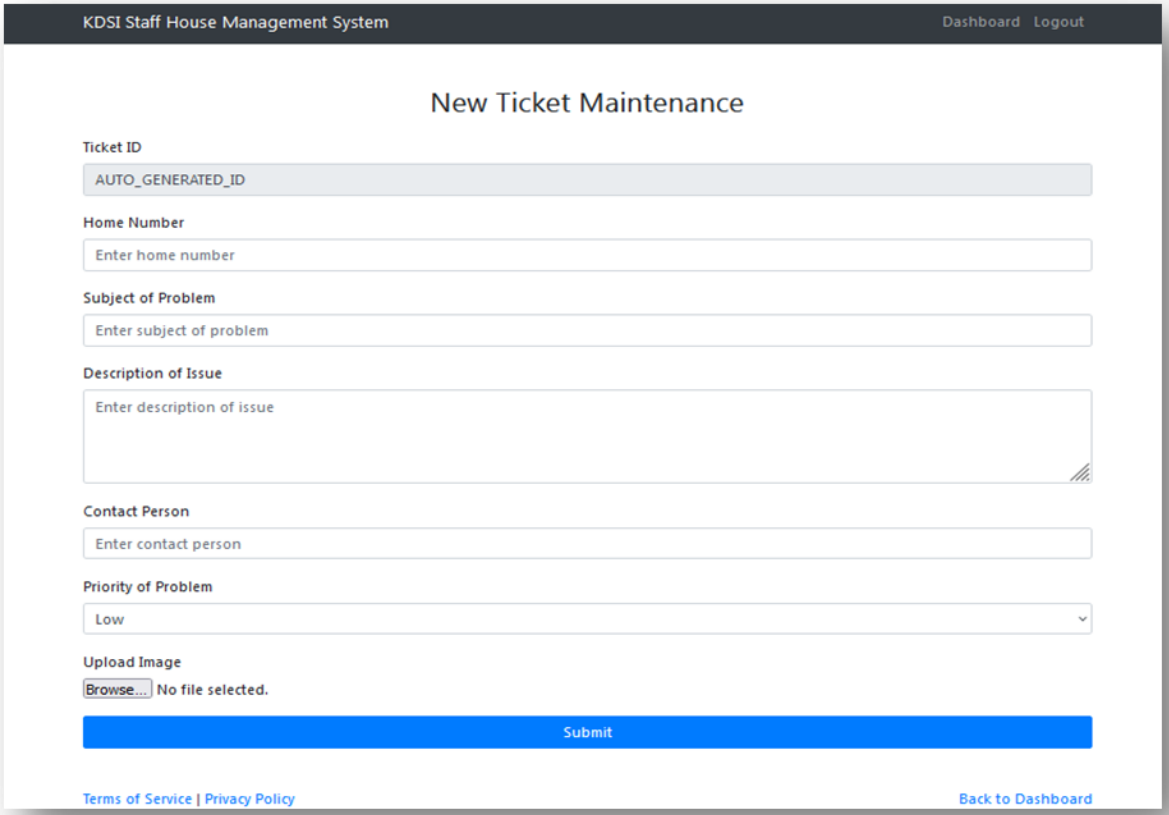
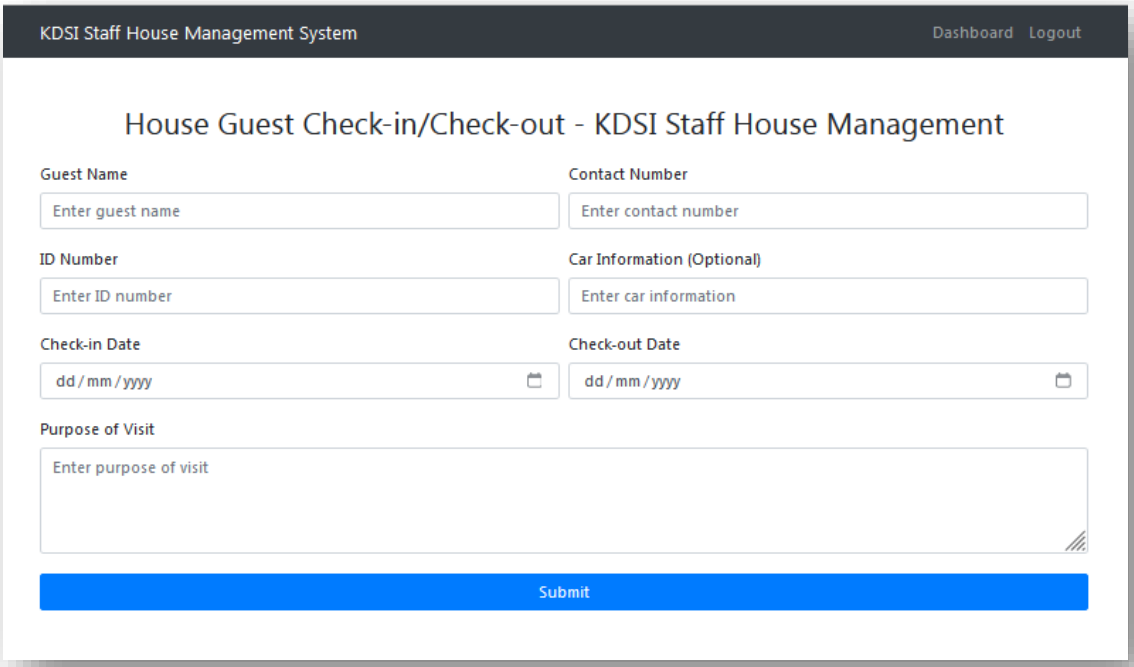


Figure 4.19 New Maintenance Request Form Interface

5. **Guest Check-In/Check-Out Form:** Manages guest entries and exits,



including verification status

Figure 4.20 Guest Check-In/Check-Out Form Interface

4.6 Chapter Summary

This chapter provides a detailed analysis and design of the KDSI Staff House Management System, focusing on several critical aspects that contribute to the system's robustness and functionality. We began by conducting a thorough requirement analysis, identifying and documenting both functional and non-functional requirements that shaped system development. This analysis was critical in understanding KD Sultan Ismail's specific needs for house management, ensuring that the system was designed to meet these specific requirements effectively.

The project design section described the system's architectural framework, emphasizing a modular and scalable three-tier architecture that divides concerns into

distinct layers: presentation, business logic, and data access. This separation improves the system's maintainability and allows for easier updates and modifications. The architecture enables efficient data flow and operation management, which is critical for the real-time functionality required in house management.

In database design, we created a relational database that stores all of the necessary data entities, including users, houses, maintenance requests, and guests. This design ensures data integrity and allows for complex queries, which are critical to the system's operational efficiency. The interface design was created to provide a user-friendly experience, with intuitive navigation and responsive layouts that adapt to different devices, thereby increasing user interaction and satisfaction.

Furthermore, the class diagram shown in this chapter depicts the relationships and dependencies between various classes in the system. It provided a blueprint for the system's structure, demonstrating how data is encapsulated within various classes and how these classes interact to carry out the system's core functions.

This chapter not only lays the groundwork for the subsequent development and implementation phases, but it also ensures that the system design is in line with the operational objectives of the KDSI Staff House Management System. The methodologies and design principles discussed here are critical for guiding the development team through the construction phase, ensuring that the final product is robust, efficient, and meets user expectations.

CHAPTER 5

CONCLUSION

This chapter summarizes the progress and insights gained during the first phase of the Staff House Management Online System project for KD Sultan Ismail. Throughout PSM1, significant progress was made in understanding and developing the system's foundational components. The project has successfully defined the key requirements and started the design and implementation processes with a structured Agile methodology. This approach has proven effective in managing the project's dynamic needs while maintaining consistent progress.

Several challenges arose during this phase, particularly in integrating complex system requirements and ensuring an intuitive and responsive user interface. These challenges were met with detailed analytical and adaptive strategies, which helped the project maintain its momentum.

The project is currently at a critical stage, with the basic framework established and significant work completed on the system's architecture and initial functionality. However, because this is the culmination of PSM1, the project is not without limitations. These constraints, which primarily concern the scope of tested functionalities and the depth of user feedback, will be addressed in the following phase.

PSM2's future plans include expanding functionality, refining the system based on user feedback, and rigorously testing the system to ensure reliability and efficiency. Future work will also go deeper into system security and performance optimization in order to effectively handle real-world operational demands.

Once fully implemented, this system is expected to significantly improve the efficiency of house management operations, shorten maintenance response times, and

secure guest management processes, all of which will benefit KD Sultan Ismail's daily operations.

This conclusion chapter highlights the project's current achievements and lays out a clear plan for the project's continuation and completion in PSM2, with the goal of creating a robust, efficient, and user-friendly system.

REFERENCES

Chen, M.-C., & Huang, S.-H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Systems with Applications, 24(4), 433-441.

Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58-73.

Gosnell, M., Woodley, R., Hicks, J., & Cudney, E. (2014). Exploring the Mahalanobis-Taguchi Approach to Extract Vehicle Prognostics and Diagnostics. Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 84-91.

Gupta, A. (2015). Classification of Complex UCI Datasets Using Machine Learning Algorithms Using Hadoop. International Journal of Scientific & Technology Research, 4(5), 85-94.

Hu, J., Zhang, L., Liang, W., & Wang, Z. (2009). Incipient mechanical fault detection based on multifractal and MTS methods. Petroleum Science, 6(2), 208-216.

Huang, C.-L., Chen, Y. H., & Wan, T.-L. J. (2012). The Mahalanobis Taguchi system—adaptive resonance theory neural network algorithm for dynamic product designs. Journal of Information and Optimization Sciences, 33(6), 623-635.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4-37.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. 2014 Science and Information Conference, 372-378.

Li, C., Yuan, J., & Qi, Z. (2015). Risky group decision-making method for distribution grid planning. International Journal of Emerging Electric Power Systems, 16(6), 591-602.

Lv, Y., & Gao, J. (2011). Condition prediction of chemical complex systems based on Multifractal and Mahalanobis-Taguchi system. ICQR2MSE 2011 -

Proceedings of 2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, 536-539.

der Maaten, L. J. P., Postma, E. O., & den Herik, H. J. (2009). Dimensionality Reduction: A Comparative Review. Technical Report TiCC TR 2009-005, 10(January), 1-41.

Motwani, R., & Raghavan, P. (1996). Randomized algorithms. ACM Computing Surveys, 28(1), 33-37.

Qinbao Song, Jingjie Ni, & Guangtao Wang. (2013). A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. IEEE Transactions on Knowledge and Data Engineering, 25(1), 1-14.

Rao, V. M., & Singh, Y. P. (2013). Decision Tree Induction for Financial Fraud Detection. Proceeding of the International Conference on Artificial Intelligence in Computer Science and ICT (AICS 2013), 321-328.

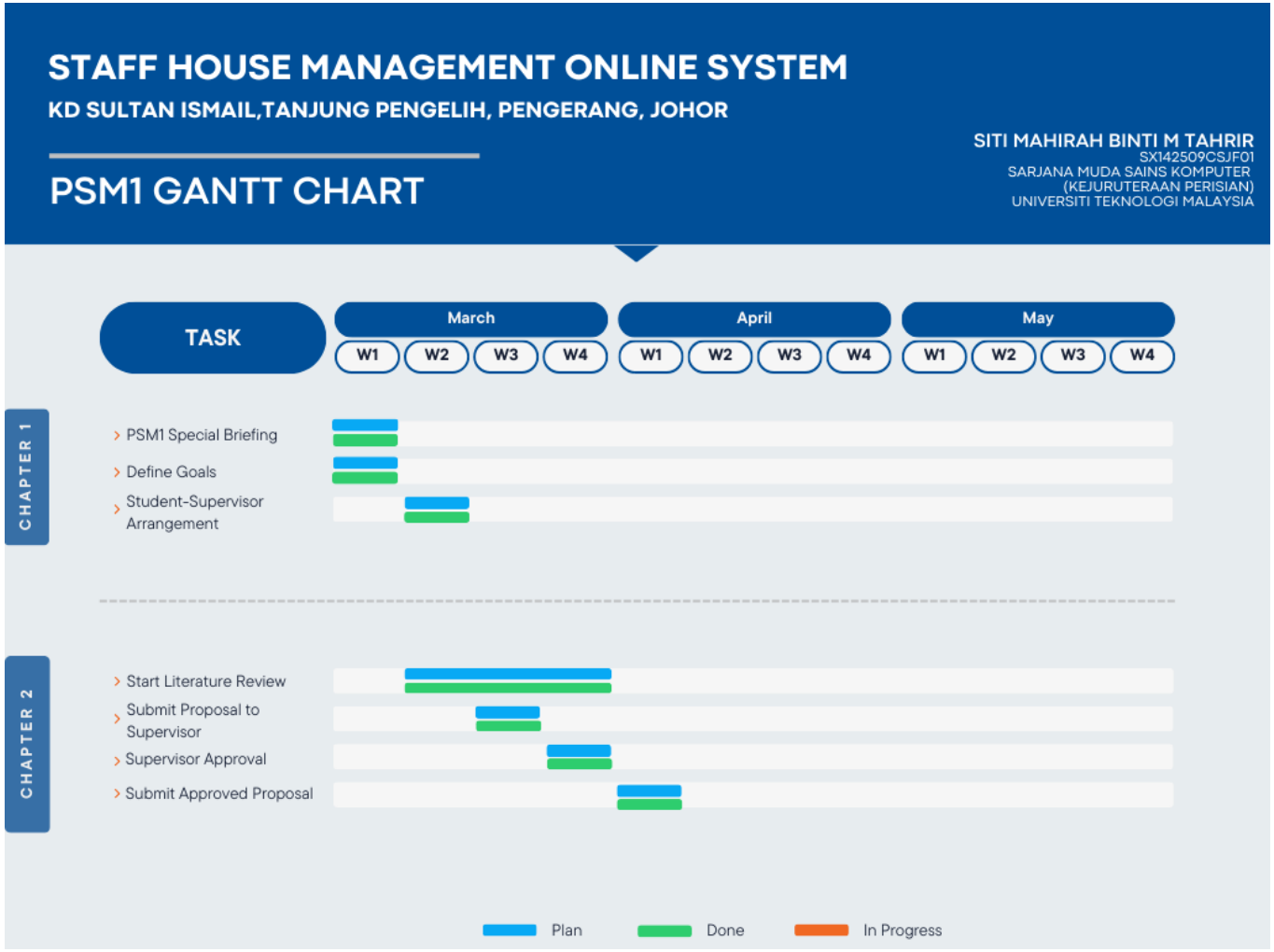
Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 69-73.

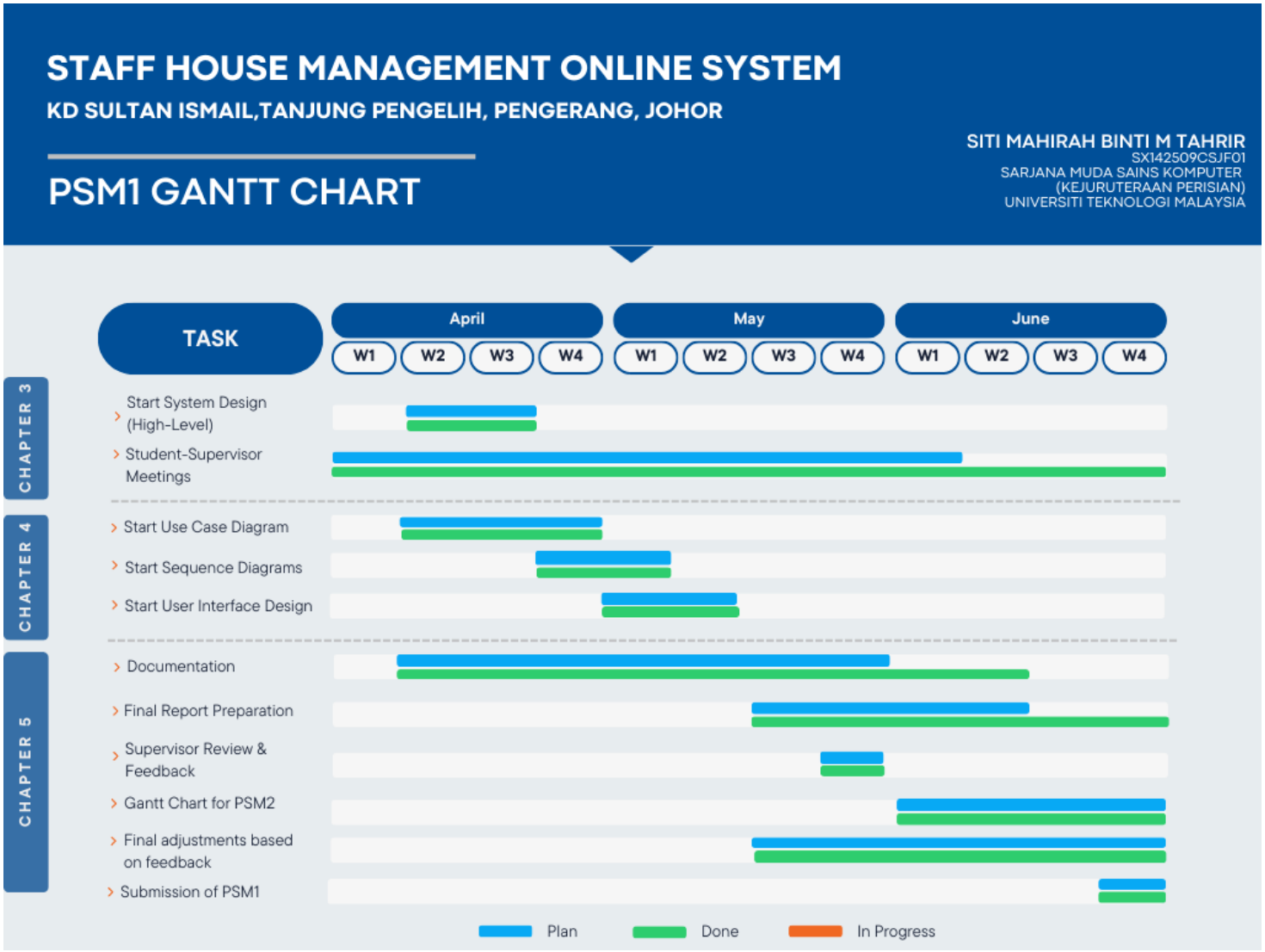
Soylemezoglu, A., Jagannathan, S., & Saygin, C. (2011). Mahalanobis-Taguchi system as a multi-sensor based decision making prognostics tool for centrifugal pump failures. IEEE Transactions on Reliability, 60(4), 864-878.

Theodoridis, S., Koutroumbas, K., Holmstrom, L., & Koistinen, P. (2009). Pattern Recognition. Wiley Interdisciplinary Reviews: Computational Statistics.

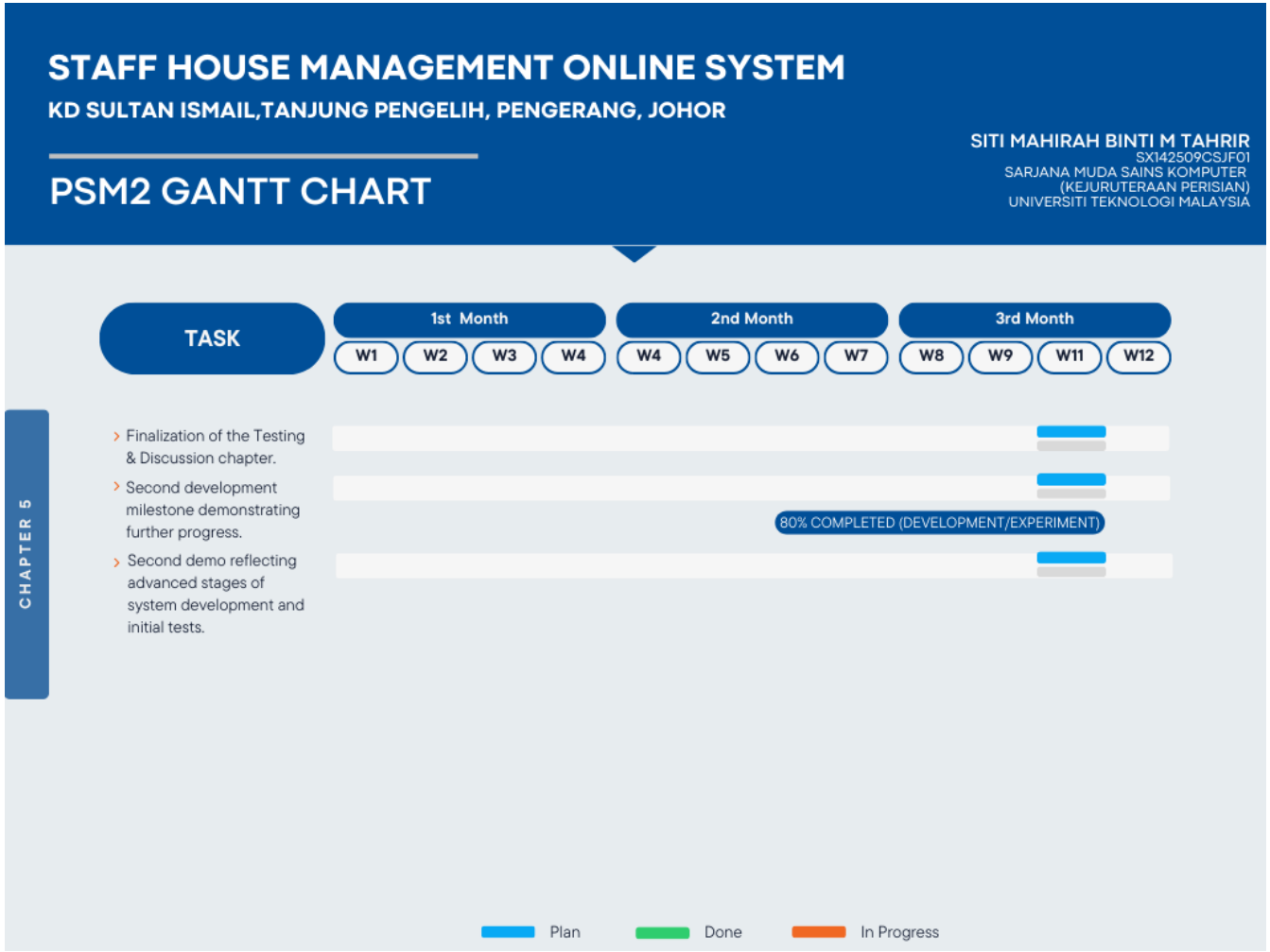
Zaki, M. J., Wong, L., Berry, M. J. A., Linoff, G. S., Hegland, M., Zaki, M. J., & Wong, L. (2003). Data Mining Techniques. WSPC/Lecture Notes Series: 9in x 6in, 10(1–2), 54.

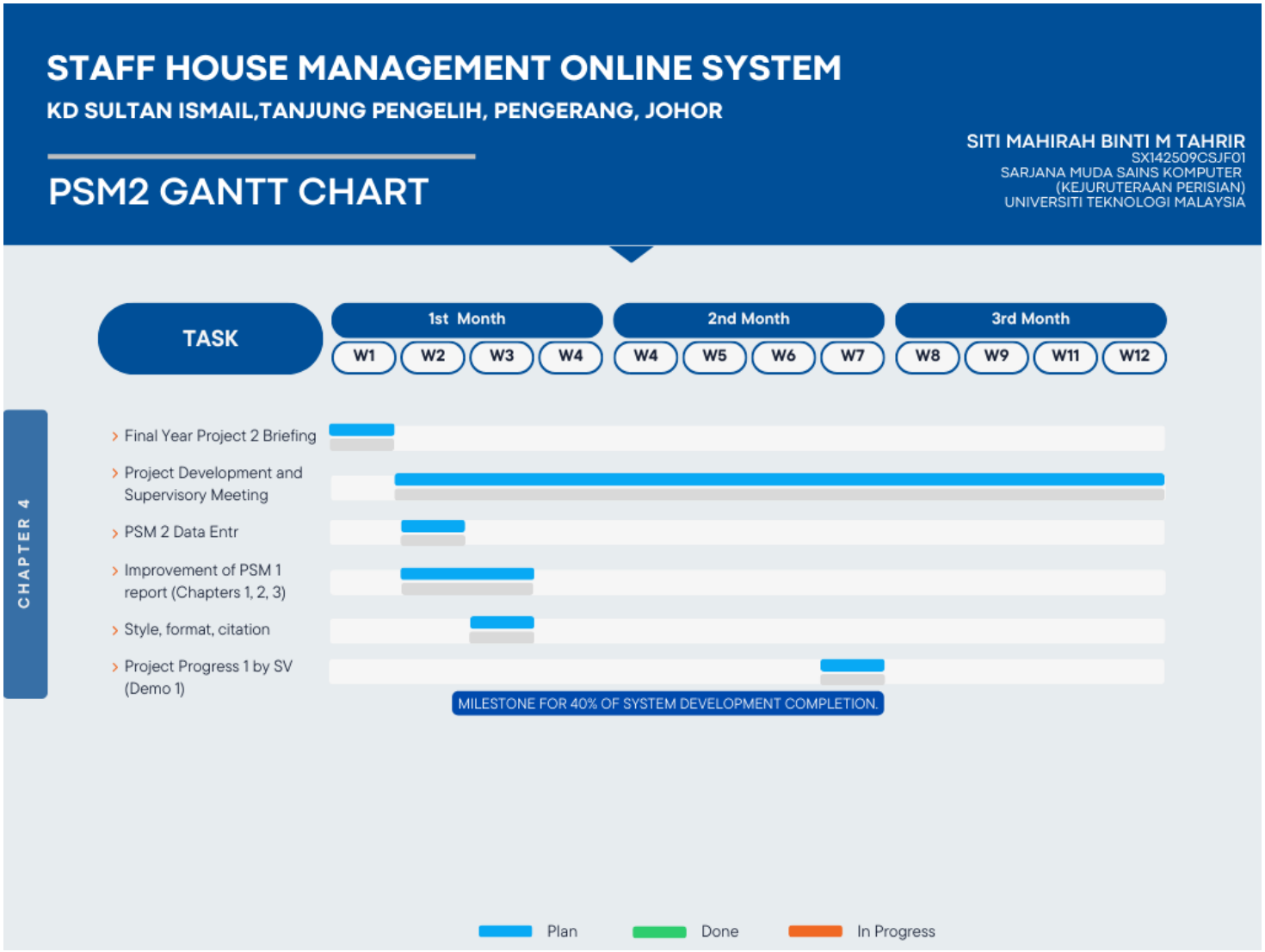
APPENDIX A GANTT CHART FOR PSM1

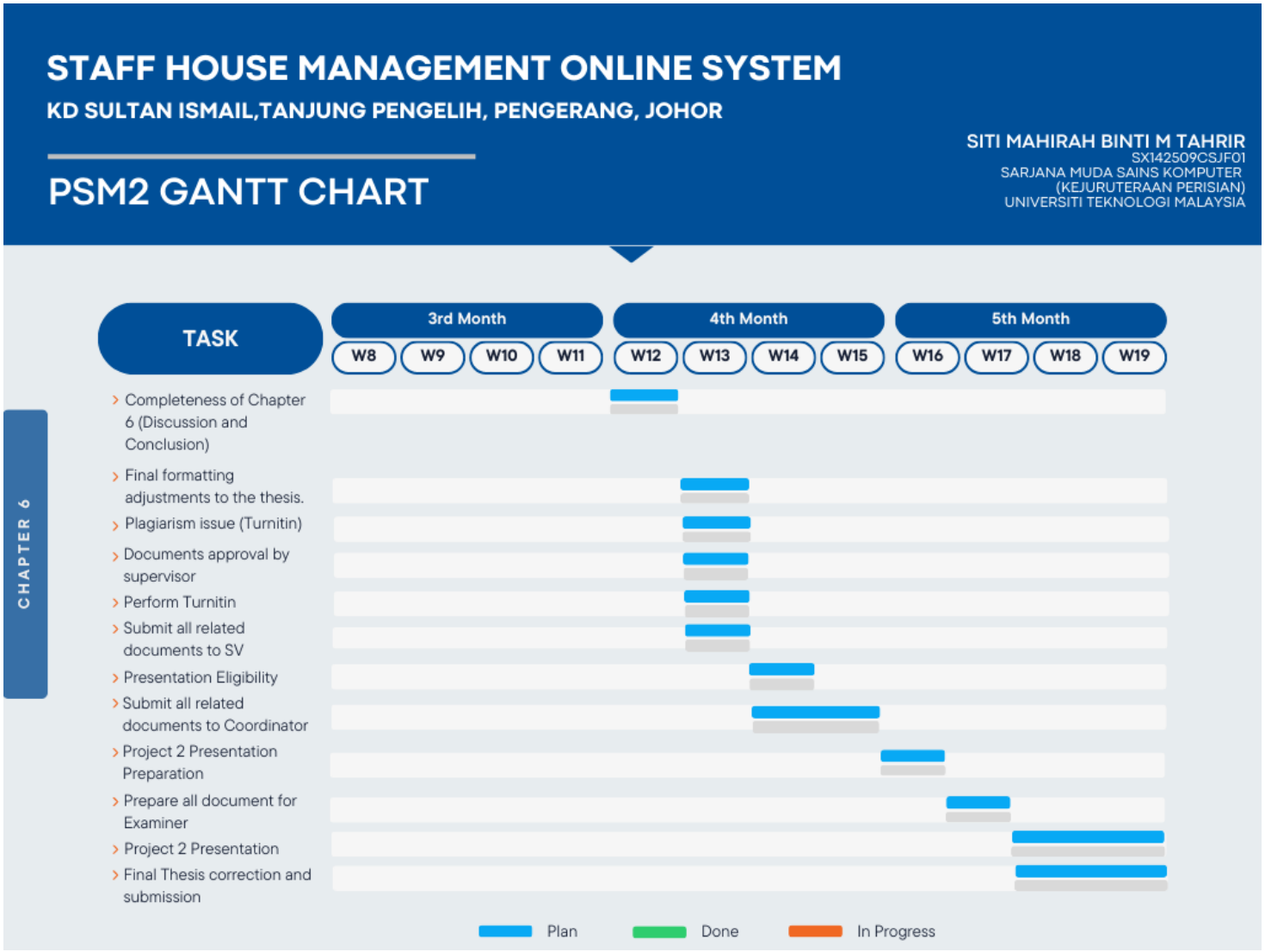



STAFF HOUSE MANAGEMENT ONLINE SYSTEM
KD SULTAN ISMAIL,TANJUNG PENGELIH, PENGERANG, JOHOR
PSM1 GANTT CHART
SITI MAHIRAH BINTI M TAHRIR
SX142509CSJF01
SARJANA MUDA SAINS KOMPUTER
(KEJURUTERAAN PERISIAN)
UNIVERSITI TEKNOLOGI MALAYSIA
TASK
April
May
June
W1
W2
W3
W4
CHAPTER 3
Start System Design (High-Level)
Student-Supervisor Meetings
CHAPTER 4
Start Use Case Diagram
Start Sequence Diagrams
Start User Interface Design
CHAPTER 5
Documentation
Final Report Preparation
Supervisor Review & Feedback
Gantt Chart for PSM2
Final adjustments based on feedback
Submission of PSM1
Plan
Done
In Progress

APPENDIX B GANTT CHART FOR PSM2




STAFF HOUSE MANAGEMENT ONLINE SYSTEM
KD SULTAN ISMAIL,TANJUNG PENGELIH, PENGERANG, JOHOR
PSM2 GANTT CHART
SITI MAHIRAH BINTI M TAHRIR
SX142509CSJF01
SARJANA MUDA SAINS KOMPUTER
(KEJURUTERAAN PERISIAN)
UNIVERSITI TEKNOLOGI MALAYSIA
CHAPTER 4
TASK
1st Month
2nd Month
3rd Month
W1
W2
W3
W4
W4
W5
W6
W7
W8
W9
W11
W12
Final Year Project 2 Briefing
Project Development and Supervisory Meeting
PSM 2 Data Entr
Improvement of PSM 1 report (Chapters 1, 2, 3)
Style, format, citation
Project Progress 1 by SV (Demo 1)
MILESTONE FOR 40% OF SYSTEM DEVELOPMENT COMPLETION.
Plan
Done
In Progress


STAFF HOUSE MANAGEMENT ONLINE SYSTEM
KD SULTAN ISMAIL,TANJUNG PENGELIH, PENGERANG, JOHOR
PSM2 GANTT CHART
SITI MAHIRAH BINTI M TAHRIR
SX142509CSJF01
SARJANA MUDA SAINS KOMPUTER
(KEJURUTERAAN PERISIAN)
UNIVERSITI TEKNOLOGI MALAYSIA
CHAPTER 6
TASK
3rd Month
4th Month
5th Month
W8
W9
W10
W11
W12
W13
W14
W15
W16
W17
W18
W19
Completeness of Chapter 6 (Discussion and Conclusion)
Final formatting adjustments to the thesis.
Plagiarism issue (Turnitin)
Documents approval by supervisor
Perform Turnitin
Submit all related documents to SV
Presentation Eligibility
Submit all related documents to Coordinator
Project 2 Presentation Preparation
Prepare all document for Examiner
Project 2 Presentation
Final Thesis correction and submission
Plan
Done
In Progress