

HYGIEIAHUB – BOOKING AND CLEANING HUB

NUR SHAHIRA ATILIA BINTI ZAINUDDIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: HYGIEIAHUB – BOOKING AND CLEANING HUB

SESI PENGAJIAN: 2024 / 2025

Saya: NUR SHAHIRA ATILIA BINTI ZAINUDDIN

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (✓)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)

TIDAK TERHAD



(TANDATANGAN PELAJAR)



(TANDATANGAN PENYELIA)

Alamat tetap: 99, Jalan Duranta 3,
Taman Desa Duranta, 70450, Seremban,
Negeri Sembilan

Tarikh: 28 Ogos 2025

Noraswaliza Abdullah

Nama Penyelia

Tarikh: 6 September 2025

HYGIEIAHUB – BOOKING AND CLEANING HUB

NUR SHAHIRA ATILIA BINTI ZAINUDDIN

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Database Management) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2025

DECLARATION

I hereby declare that this project report entitled

HYGIEIAHUB – BOOKING AND CLEANING HUB

is written by me and is my own effort and that no part has been plagiarized
without citations.

STUDENT :
(NUR SHAHIRA ATILIA BINTI ZAINUDDIN)



Date : 28 August 2025

I hereby declare that I have read this project report and found
this project report is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR : _____ Date : 6 Sept 2025
(TS. DR. NORASWALIZA BINTI ABDULLAH)



DEDICATION

I dedicate this project to my beloved parents, whose love, support, and encouragement have been my greatest source of strength throughout my academic journey.

ACKNOWLEDGEMENTS

First and foremost, I humbly express my deepest gratitude to Allah Ta'ala, the Most Gracious, the Most Merciful, for His infinite guidance, strength, and blessings throughout the course of this final year project. It is only through His will and mercy that I have been able to overcome challenges, remain steadfast, and successfully complete this academic journey.

I would also like to extend my sincere appreciation to my supervisor, Dr. Noraswaliza, for her continuous support, invaluable guidance, and encouragement throughout the entire duration of this project. The constructive feedback and expertise given by my supervisor have greatly contributed to the successful completion of my work.

My heartfelt thanks go to the faculty and staff of Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, for providing the necessary resources, knowledge, and support throughout my undergraduate studies.

Special thanks to my family, especially my parents, for their unconditional love, prayers, and constant motivation. Their belief in me has been my greatest source of inspiration.

I would also like to acknowledge my friends and coursemates for their moral support, shared knowledge, and memorable moments during this academic journey.

Finally, I am thankful to everyone who has contributed directly or indirectly to the success of this project. Your support is deeply appreciated.

ABSTRACT

HygieiaHub is a web-based cleaning service booking platform developed within the field of database-driven system implementation. The project addresses common challenges faced by local service providers and customers, including inefficient booking workflows, lack of service transparency, and absence of performance reporting tools. Existing platforms often introduce complexity and generalized features that do not suit small-scale operations, especially those relying on manual scheduling and cash-on-delivery transactions. To solve these issues, HygieiaHub was designed with a simplified structure that supports predefined service packages, real-time booking management, and feedback integration. The system was developed using PHP, MySQL, and XAMPP, following the Waterfall methodology to ensure a structured progression from requirement analysis to deployment. Key components include user registration, booking modules, cleaner assignment, payment tracking, and reporting dashboards. The database schema was normalized to maintain data integrity, and SQL queries were implemented to support dynamic filtering and periodic sales reporting. Interface designs were tailored for both customer and staff roles, ensuring usability and role-based access control. During testing, the system successfully handled booking flows, feedback submissions, and report generation, meeting all functional requirements outlined in the early phases. The final product operates fully in a local environment and demonstrates the feasibility of deploying a lightweight, role-specific booking system for single-provider cleaning services.

ABSTRAK

HygieiaHub ialah sebuah platform tempahan perkhidmatan pembersihan berasaskan web yang dibangunkan dalam bidang pelaksanaan sistem berasaskan pangkalan data. Projek ini bertujuan untuk menangani isu utama yang dihadapi oleh penyedia perkhidmatan pembersihan tempatan dan pelanggan, termasuk aliran tempahan yang tidak efisien, kekurangan ketelusan perkhidmatan, dan ketiadaan alat pelaporan prestasi. Kebanyakan platform sedia ada terlalu kompleks dan tidak sesuai untuk operasi berskala kecil yang masih bergantung kepada penjadualan manual dan kaedah pembayaran tunai semasa penghantaran (COD). Bagi menyelesaikan masalah ini, HygieiaHub direka bentuk dengan struktur yang ringkas dan menyokong pakej perkhidmatan yang telah ditetapkan, pengurusan tempahan secara masa nyata, serta integrasi maklum balas pelanggan. Sistem ini dibangunkan menggunakan PHP, MySQL dan XAMPP, dengan pendekatan metodologi Waterfall bagi memastikan setiap fasa pembangunan berjalan secara teratur. Komponen utama termasuk modul pendaftaran pengguna, tempahan perkhidmatan, penugasan pencuci, penjejakkan pembayaran, dan papan pemuka pelaporan. Skema pangkalan data telah dinormalisasi bagi mengekalkan integriti data, manakala pertanyaan SQL digunakan untuk penapisan dinamik dan pelaporan jualan berkala. Reka bentuk antara muka disesuaikan mengikut peranan pengguna (pelanggan dan kakitangan) bagi memastikan kebolehgunaan dan kawalan akses yang berkesan. Semasa ujian, sistem berjaya mengendalikan aliran tempahan, penghantaran maklum balas, dan penjanaan laporan, serta memenuhi semua keperluan fungsian yang telah ditetapkan. Produk akhir beroperasi sepenuhnya dalam persekitaran setempat dan membuktikan kebolehlaksanaan sistem tempahan yang ringan dan khusus untuk penyedia perkhidmatan pembersihan tunggal.

TABLE OF CONTENTS

	PAGE
DECLARATION.....	II
DEDICATION.....	III
ACKNOWLEDGEMENTS.....	IV
ABSTRACT	V
ABSTRAK	VI
TABLE OF CONTENTS.....	VII
LIST OF TABLES	XIII
LIST OF FIGURES	XV
LIST OF ABBREVIATIONS	XVIII
LIST OF ATTACHMENTS.....	XIX
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Scope of the Project	3
1.4.1 User Scope	3
1.4.2 Module Scope	3
1.5 Expected Outcome	6
1.6 Conclusion	6

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY .	8	
2.1	Introduction.....	8
2.2	Literature Review.....	9
2.2.1	Domain	9
2.2.2	Existing Cleaning Service Platforms	9
2.2.3	Importance of Simplified Systems for Local Businesses	11
2.2.4	Trends in Web-Based Booking Systems	11
2.2.5	Summary.....	12
2.3	Project Methodology.....	12
2.3.1	Software Development Life Cycle (SDLC)	12
2.3.2	Database Development Life Cycle (DBLC).....	14
2.4	Project Schedule and Milestones	16
2.4.1	Milestones.....	16
2.4.2	Gantt Chart.....	21
2.5	Conclusion	23
	CHAPTER 3: ANALYSIS.....	24
3.1	Introduction.....	24
3.2	Problem Analysis	24
3.3	Proposed System.....	26
3.4	Requirements Analysis	26
	3.4.1.1 Context Diagram.....	28
	3.4.1.2 Data Flow Diagram Level 0	30
	3.4.1.3 Data Flow Diagram Level 1	34
	3.4.2 Non-Functional Requirements.....	39

3.4.3	Technical Requirements	40
3.4.3.1	Hardware Requirements	40
3.4.3.2	Software Requirements.....	40
3.5	Conclusion	41
CHAPTER 4: DESIGN		42
4.1	Introduction.....	42
4.2	System Architecture.....	42
4.3	Data Model Design	43
4.3.1	Conceptual Design.....	44
4.3.1.1	Business Rules	45
4.3.2	Logical Design.....	46
4.3.3	Physical Design	52
4.3.3.1	Data Definition Language.....	52
4.4	Graphical User Interface Design.....	66
4.4.1	Navigation System.....	66
4.4.2	Input Design.....	68
4.4.3	Output Design	72
4.5	Conclusion	76
CHAPTER 5: IMPLEMENTATION.....		77
5.1	Introduction.....	77
5.2	Environment Setup.....	77
5.2.1	Software (S/W) Environment	77
5.2.2	Hardware (H/W) Requirements	78
5.2.3	Database Connection	79

5.3	DML Implementation	80
5.3.1	DML Insert Statement	80
5.3.2	DML Update Statement	81
5.3.3	DML Delete Statement	81
5.4	Query Implementation	82
5.4.1	Query to Retrieve Filtered Bookings by Customer Preferences....	82
5.4.2	Query to Generate Sales Report Over Selected Periodicity	83
5.4.3	Query to Retrieve All Bookings with Optional Filters for Management	84
5.5	System Interface.....	86
5.5.1	Interfaces of User Customer	86
5.5.1.1	Sign Up Page	86
5.5.1.2	Sign In Page	87
5.5.1.3	Home Page	87
5.5.1.4	Profile Page.....	88
5.5.1.5	Community Page	89
5.5.1.6	Book Now Page and Invoice Modal	90
5.5.1.7	Bookings Page	92
5.5.2	Interfaces of User Staff.....	93
5.5.2.1	Sign In Page	93
5.5.2.2	Dashboard Page	94
5.5.2.3	Profile Page.....	95
5.5.2.4	Add House Type and Edit House Type Page	96
5.5.2.5	Add, Edit, and View Service Pages	97

5.5.2.6 Manage Staff Page, Register New Staff Modal, and Edit Staff Modal	99
5.5.2.7 Manage Booking Page and Booking Details Modal.....	100
5.5.2.8 Sales Report, Feedback Report, Staff Performance Report, and Service Utilization Report Pages	102
5.5.2.9 Maintenance Page.....	105
5.6 Conclusion	106
CHAPTER 6: TESTING AND VALIDATION	108
6.1 Introduction.....	108
6.2 Test Strategy	108
6.3 Test Plan.....	111
6.3.1 Functional Testing	111
6.3.2 Non-Functional Testing	136
6.4 System Usability Scale (SUS) Evaluation	138
6.5 Discussion.....	143
6.6 Conclusion	144
CHAPTER 7: PROJECT CONCLUSION	146
7.1 Introduction.....	146
7.2 Project Conclusion	146
7.3 Limitation.....	147
7.4 Future Work	148
APPENDIX	149
Testing and Feedback Survey	149

REFERENCES.....	157
------------------------	------------

LIST OF TABLES

	PAGE
Table 2.1: Comparison of Existing Systems	10
Table 2.2: Milestones	17
Table 3.1: Functional Requirements	27
Table 3.2: Non-Functional Requirements.....	39
Table 3.3: Hardware Requirements	40
Table 3.4: Software Requirements	41
Table 4.1: Data Dictionary of Table customer	47
Table 4.2: Data Dictionary of Table house_type	47
Table 4.3: Data Dictionary of Table customer_addresses.....	48
Table 4.4: Data Dictionary of table staff.....	48
Table 4.5: Data Dictionary of Table additional_service	49
Table 4.6: Data Dictionary of Table booking	49
Table 4.7: Data Dictionary of Table booking_service.....	50
Table 4.8: Data Dictionary of Table booking_cleaner	50
Table 4.9: Data Dictionary of Table payment	50
Table 4.10: Data Dictionary of Table feedback.....	51
Table 4.11: List of Database Objects using MySQL Command	57
Table 5.1: Software Environment Summary.....	78
Table 5.2: Hardware Requirement Summary.....	78
Table 6.1 List of Test Strategy	110
Table 6.2: Functional Testing of Sign In (Customer)	112
Table 6.3: Functional Testing of Sign In (Staff)	113
Table 6.4: Functional Testing of Sign Up (Customer)	115
Table 6.5: Functional Testing of Register (Staff)	118

Table 6.6: Functional Testing of Book (Customer).....	120
Table 6.7: Functional Testing of Managing Bookings (Customer)	123
Table 6.8: Functional Testing of Add House Type (Staff).....	125
Table 6.9: Functional Testing of Edit House Type (Staff).....	126
Table 6.10: Functional Testing of Add Additional Service (Staff)	127
Table 6.11: Functional Testing of Edit House Type (Staff).....	128
Table 6.12: Functional Testing of Manage Bookings (Staff).....	128
Table 6.13: Functional Testing of Manage Staff (Staff)	132
Table 6.14: Functional Testing of Generate Report (Staff).....	133
Table 6.15: Non-Functional Testing	137

LIST OF FIGURES

	PAGE
Figure 2.1: Waterfall Model of Software Development Life Cycle (SDLC).....	14
Figure 2.2: Model of Database Life Cycle (DBLC).....	15
Figure 2.3: Gantt Chart.....	22
Figure 3.1: Current system flowchart.....	25
Figure 3.2: Context Diagram	29
Figure 3.3: Level 0 of DFD for Process 1, 2, 3, and 4.....	31
Figure 3.4: Level 0 of DFD for Process 5 and 6.....	32
Figure 3.5: Level 0 of DFD for Process 7 and 8.....	33
Figure 3.6: Level 1 of DFD for Process 1 and 2	35
Figure 3.7: Level 1 of DFD for Process 3 and 4.....	37
Figure 3.8: Level 1 of DFD for Process 5	37
Figure 3.9: Level 1 of DFD for Process 6 and 7	39
Figure 4.1: System Architecture of HygieiaHub	43
Figure 4.2: Entity Relationship Diagram.....	45
Figure 4.3: DDL for Table customer	52
Figure 4.4: DDL for Table house_type.....	53
Figure 4.5: DDL for Table customer_addresses.....	53
Figure 4.6: DDL for Table staff	54
Figure 4.7: DDL for Table additional_service.....	54
Figure 4.8: DDL for Table booking	55
Figure 4.9: DDL for Table booking_service	55
Figure 4.10: DDL for Table booking_cleaner.....	56
Figure 4.11: DDL for Table payment.....	56
Figure 4.12: DDL for Table feedback	56

Figure 4.13: Interface Structure for Customer	66
Figure 4.14: Interface Structure for Staff.....	67
Figure 4.15: Customer - Sign Up Page.....	68
Figure 4.16: Customer & Staff - Sign In Page.....	69
Figure 4.17: Customer - Book Now Page.....	69
Figure 4.18: Staff - Register Staff Page.....	70
Figure 4.19: Staff - Add Service Page.....	70
Figure 4.20: Staff - Edit Service Page.....	71
Figure 4.21: Staff - Manage Booking Page for editing	72
Figure 4.22: Customer - Bookings Page.....	73
Figure 4.23: Customer - Community Page	73
Figure 4.24: Staff - View Service Page	74
Figure 4.25: Staff - Manage Staff Page	74
Figure 4.26: Staff - Manage Booking Page	75
Figure 4.27: Staff - Sales Report Page.....	75
Figure 4.28: Staff - Feedback Report Page.....	76
Figure 5.1: Database Connection Script	79
Figure 5.2: DML Statement for Select booking details	83
Figure 5.3: Result for DML in Figure 5.2	83
Figure 5.4: DML Statement for Select yearly sales for 2025	84
Figure 5.5: Result for DML in Figure 5.4	84
Figure 5.6: DML Statement for Management Booking Overview	85
Figure 5.7: Result for DML in Figure 5.6	85
Figure 5.8: Sign Up Page for Customer	86
Figure 5.9: Sign In Page for Customer.....	87
Figure 5.10: Home Page for Customer.....	88
Figure 5.11: Profile Page for Customer	89
Figure 5.12: Community Page for Customer	90
Figure 5.13: Book Now Page for Customer	91
Figure 5.14: Invoice Modal.....	92
Figure 5.15: Bookings Page for Customer	93
Figure 5.16: Sign In Page for Staff	94
Figure 5.17: Dashboard Page for Staff.....	95
Figure 5.18: Profile Page for Staff.....	96

Figure 5.19: Add House Type Page for Staff.....	96
Figure 5.20: Edit House Type Page for Staff.....	97
Figure 5.21: Add Service Page for Staff.....	97
Figure 5.22: Edit Service Page for Staff.....	98
Figure 5.23: View Service Page for Staff	98
Figure 5.24: Manage Staff Page for Staff.....	99
Figure 5.25: Register New Staff Modal for Staff.....	100
Figure 5.26: Edit Staff Modal for Staff	100
Figure 5.27: Manage Booking Page for Staff	101
Figure 5.28: Booking Details Modal for Staff.....	101
Figure 5.29: Sales Report Page for Staff.....	102
Figure 5.30: Feedback Report Page for Staff	103
Figure 5.31: Staff Performance Report Page for Staff	104
Figure 5.32: Service Utilization Report Page for Staff	105
Figure 5.33: Maintenance Page for Staff	106
Figure 6.1: SUS Question 1	138
Figure 6.2: SUS Question 2	139
Figure 6.3: SUS Question 3	139
Figure 6.4: SUS Question 4	140
Figure 6.5: SUS Question 5	140
Figure 6.6: SUS Question 6	141
Figure 6.7: SUS Question 7	141
Figure 6.8: SUS Question 8	142
Figure 6.9: SUS Question 9	142
Figure 6.10: SUS Question 10	143

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
DFD	-	Data Flow Diagram
ERD	-	Entity Relationship Diagram
DML	-	Data Manipulation Language

LIST OF ATTACHMENTS

	PAGE	
Appendix A	Testing and Feedback Survey	149

CHAPTER 1: INTRODUCTION

1.1 Introduction

HygieiaHub is a dedicated cleaning service booking platform developed to simplify and streamline the process of scheduling professional home cleaning services. In many traditional settings, booking such services involves manual communication, inconsistent pricing, and unclear service expectations, often leading to inconvenience and customer dissatisfaction. HygieiaHub addresses these challenges by offering a structured and efficient digital solution tailored for both customers and service providers.

The platform provides a set of predefined cleaning service packages based on house types, allowing users to quickly select the service that best suits their needs. By standardizing the offerings and linking them to specific house categories, HygieiaHub ensures clarity in pricing and expectations, reducing the likelihood of misunderstandings or hidden charges. A key feature of the platform is its exclusive use of cash-on-delivery (COD) as the sole payment method. This approach caters to users who prefer traditional payment practices, offering them greater control and confidence by allowing payment only after the service has been completed to their satisfaction.

From an operational standpoint, HygieiaHub also facilitates smoother management of service requests. It enables cleaning service providers to receive and track bookings efficiently, allocate staff, and ensure timely delivery of services. This dual focus on customer convenience and operational oversight makes HygieiaHub a practical and reliable tool in the growing on-demand service industry.

In summary, HygieiaHub is designed to modernize the home cleaning service experience by combining standardized service packages, transparent pricing, and COD-based transactions within a user-friendly digital platform.

1.2 Problem Statement

Many customers experience inconvenience when attempting to book cleaning services due to unstructured and outdated scheduling systems. The absence of a standardized booking process often results in confusion, miscommunication, and difficulty securing preferred time slots. This not only affects customer satisfaction but also impacts the efficiency of service providers in managing daily operations.

In addition to booking issues, there is often no reliable system in place for tracking service status. Customers are left uncertain about when or if a service will be delivered, which diminishes trust in the provider. On the service side, staff members face challenges in monitoring bookings, assigning tasks, and responding promptly to customer requests, leading to inefficiencies and potential service delays.

Furthermore, the lack of data analytics and reporting tools severely limits the ability of service providers to evaluate performance and anticipate demand. Without insights into customer behavior, peak booking times, or service preferences, it becomes difficult to optimize operations, plan resource allocation, or improve service quality. This data gap hinders growth and responsiveness in a competitive service market.

1.3 Objectives

The objective of this study is to

- (i) Design and implement a structured and user-friendly booking system that simplifies the scheduling of cleaning services by providing predefined service options, time slots, and house types to minimize booking conflicts and improve customer convenience.

- (ii) Develop and deploy a real-time service tracking feature that enhances transparency and communication between customers and service providers by allowing customers to monitor booking status, while enabling staff to manage appointments, allocate resources, and oversee operations effectively.
- (iii) Integrate and evaluate a reporting and analytics system that captures key operational data such as service demand, peak booking times, and customer preferences, in order to support data-driven decision-making and improve overall service quality.

1.4 Scope of the Project

1.4.1 User Scope

- (i) Customer – Individuals seeking to book professional cleaning services.
- (ii) Staff – Personnel responsible for managing services, bookings, and system administration.

1.4.2 Module Scope

(i) Registration and Login

- Users can create an account and log in using a valid email address and password.
- Only registered users are granted access to the system.
- Access by: Customers, Staff

(ii) Profile Update

- Users can update their account profile including phone number, email, or password.
- Customers may update their addresses.
- Access by: Customers, Staff

(iii) Creating and Updating House Types and Services

- Staff can define various house types (e.g., apartment, terrace, bungalow) and create associated cleaning services.
- Staff can also update house type details, edit service information, or delete services when necessary.
- Access by: Staffs

(iv) Booking Cleaning Services

- Customers can browse a list of available cleaning services, each with a description and corresponding price.
- Booking options include selecting optional services, choosing base booking hours, specifying the number of cleaners, entering custom requests, and selecting a preferred date and time.
- The system automatically assigns a cleaner based on real-time availability once a booking is confirmed.
- Access by: Customers

(v) Managing Bookings

- Staff can view, update, and manage all customer bookings, including changing booking statuses (e.g., Pending → Completed / Cancelled), and re-assign cleaner for pending booking.
- Customers can view the current status of their bookings and are allowed to cancel if needed.
- Access by: Customers (view only), Staffs (manage bookings)

(vi) Payment Processing

- The system automatically calculates the total cost of services based on selected booking details, including house type, service type, and applicable taxes.
- Customers are notified to prepare cash-on-delivery (COD) payment upon service completion.
- Staff are responsible for updating the payment status after service completion.
- Access by: Customers (view payment details), Staffs (update payment status)

(vii) Customer Feedback & Ratings

- Customers can provide feedback by rating completed services and leaving written comments.
- Staff can access and review customer feedback to assess service quality and identify areas for improvement.
- Access by: Customers (submit and view feedback), Staffs (view feedback)

(viii) Reporting & Analytics

- Staff can generate performance reports based on daily, monthly, or yearly service sales, staff performance, and service utilization, as well as monthly customer feedback summaries.
- These insights help in monitoring performance and improving operational decisions.
- Access by: Staffs

1.5 Expected Outcome

One of the primary expected outcomes of this project is a fully functional cleaning service booking platform that enables customers to schedule and manage cleaning services with ease. By providing a selection of predefined services tied to specific house types and prices, the platform eliminates ambiguity in service selection and cost estimation. The user-friendly interface will allow customers to make bookings, set preferences, and receive service confirmations efficiently, ensuring a smooth and convenient experience from start to finish.

Another significant expected result is the implementation of an efficient booking and service status tracking system. This feature will allow customers to stay informed about their service progress in real time, increasing transparency and trust in the platform. At the same time, staff members and managers will benefit from streamlined operational control, enabling them to update booking statuses, reassign resources, and respond to service changes promptly. This dual-view system enhances both user satisfaction and backend efficiency.

Additionally, the platform is expected to include a reporting and analytics feature that provides valuable insights for service managers. This tool will offer data on key metrics such as booking frequency, peak service hours, customer feedback trends, and overall service performance. With access to this information, managers can make informed decisions to optimize staffing, improve service offerings, and align business strategies with customer demand patterns, ultimately supporting continuous improvement and sustainable growth.

1.6 Conclusion

In conclusion, this chapter has introduced HygieiaHub, a web-based cleaning service booking platform designed to address limitations found in current service platforms, particularly in terms of complexity and regional relevance. By focusing on simplicity and essential features, the platform aims to provide a user-friendly experience for customers and efficient management tools for admins and managers. The problem statement, objectives, scope, and project significance lay the foundation

for the development and implementation of a practical, database-driven solution tailored to a single service provider.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter presents the foundational research and methodology applied in the development of HygieiaHub, a web-based cleaning service booking system tailored for single-provider operations. The system was conceptualized to address inefficiencies in manual booking processes, lack of service transparency, and the absence of structured performance tracking tools commonly faced by local cleaning businesses. To ensure the system's relevance and effectiveness, a literature review was conducted to examine existing platforms such as Kaodim, Maideasy, and ServisHero, focusing on their strengths, limitations, and technological frameworks. This comparative analysis provided insight into industry standards and user expectations, guiding the design of a simplified, role-specific solution.

Additionally, the chapter outlines the project methodology used to manage the development lifecycle, emphasizing the use of the Waterfall model for its structured and sequential approach. The methodology section also includes the Database Life Cycle (DBLC) to support the planning and implementation of the system's relational database. Finally, the chapter presents the project schedule and milestones to illustrate the timeline and progress of each development phase, ensuring alignment with academic and functional goals.

2.2 Literature Review

2.2.1 Domain

The domain of this project falls under web-based service management systems, specifically within the context of cleaning service operations. These systems are designed to facilitate the coordination between service providers and customers through digital platforms that support booking, scheduling, and feedback mechanisms. In recent years, the demand for online service platforms has grown significantly, driven by the convenience of remote access, real-time updates, and streamlined workflows. For cleaning services, digital systems offer an effective solution to replace manual scheduling, reduce administrative overhead, and improve customer satisfaction.

HygieiaHub is developed within this domain to address the operational needs of a single-provider cleaning business, focusing on simplicity, role-based access, and internal reporting. The system integrates core functionalities such as booking management, cleaner assignment, payment tracking, and feedback collection, all of which are essential components in modern service platforms. By situating HygieiaHub within this domain, the project aims to contribute to the broader field of service digitization and demonstrate how tailored solutions can enhance the efficiency and transparency of localized service delivery.

2.2.2 Existing Cleaning Service Platforms

Several existing platforms, such as Kaodim, Maideeasy, and ServisHero, offer on-demand home cleaning services to urban consumers. These platforms typically provide a wide range of features, including service listings, booking forms, and customer feedback mechanisms. However, they often introduce complexity in navigation and require users to go through multi-step booking processes. Many of these services also lack transparent pricing structures or promote recurring subscription models, which may not appeal to users seeking one-time services. Furthermore, due to their operation across various regions, these platforms tend to

adopt generalized design approaches that may compromise user experience in specific local contexts.

To better understand the positioning of HygieiaHub, a comparison was made between these platforms and the proposed system. HygieiaHub is designed specifically for a single cleaning service provider, offering automated cleaner assignment, cash-on-delivery (COD) payment support, and internal reporting tools tailored to operational needs. This focused approach simplifies the booking process and enhances service transparency for both customers and staff.

Table 2.1: Comparison of Existing Systems

Feature	Kaodim	Maideasy	ServisHero	HygieiaHub (Proposed)
Booking System	Yes	Yes	Yes	Yes
Cleaner Assignment	Manual/Auto	Manual	Manual	Manual/Auto
Feedback & Rating	Yes	Yes	Yes	Yes
Payment Method	Online	Online/COD	Online	COD only
Role-Based Access	Limited	Limited	Limited	Full (Customer/Staff)
Reporting Dashboard	No	No	No	Yes
Target Provider Type	Multi-provider	Single-provider	Multi-provider	Single-provider

This observation highlighted that simplicity and interactivity in mobile application usability are positively correlated with user satisfaction, trust, and brand loyalty. This underlines the importance of designing service platforms that are easy to navigate and tailored to user expectations, especially in a localized context.

2.2.3 Importance of Simplified Systems for Local Businesses

Small and medium-sized enterprises (SMEs), particularly in the service sector, often prioritize practical and easily manageable solutions over complex, feature-heavy platforms. These businesses typically require clear booking workflows, intuitive interfaces, and basic analytics, rather than advanced enterprise-level systems. As Brown and Lockett (2004) observed, SMEs tend to favor low-complexity applications and shy away from tools with steep learning curves or unnecessary functionalities.

In this regard, HygieiaHub adopts a minimalist design approach that emphasizes functionality over feature-bloat. It focuses specifically on one-time cleaning service bookings, enabling local businesses to manage operations efficiently without requiring advanced technical knowledge or dedicated IT support. The system's structure supports essential features such as booking, feedback, and reporting, while avoiding unnecessary modules that could overwhelm users.

2.2.4 Trends in Web-Based Booking Systems

Web-based platforms remain the most accessible and widely adopted solution for service-based businesses, primarily due to their cross-platform compatibility, ease of deployment, and scalability. Technologies such as PHP and MySQL continue to be prominent choices for academic and small-scale commercial projects. These technologies integrate well with development tools like XAMPP and Visual Studio Code, providing a lightweight and flexible development environment (Halim, Abdullah, & Mohd Yasin, 2020).

For the development of HygieiaHub, a web-based architecture ensures that users can access the system through any modern browser without the need for dedicated mobile or desktop applications. This not only simplifies development and maintenance but also increases the platform's accessibility for both customers and staff. The use of open-source tools also supports cost-effective deployment and future scalability.

2.2.5 Summary

In summary, the literature review highlights the limitations of existing platforms in serving localized cleaning service providers and emphasizes the need for simplified, role-specific systems. HygieiaHub addresses these gaps by offering a streamlined booking experience, automated cleaner assignment, and internal reporting features. Supported by proven technologies and academic principles, the system is positioned to deliver practical value to both customers and service staff in a single-provider context.

2.3 Project Methodology

This section explains the methodology used to develop HygieiaHub, including the phases of system development and database planning. A structured approach was adopted to ensure that each component of the system from interface design to backend logic was developed systematically and aligned with the project's functional goals. The methodology combines the Software Development Life Cycle (SDLC) and the Database Development Life Cycle (DBLC), both of which guided the planning, implementation, and evaluation of the system.

2.3.1 Software Development Life Cycle (SDLC)

For the development of HygieiaHub, the Waterfall Model was selected due to its linear and phase-based structure, which is suitable for projects with clearly defined requirements. The development began with the Requirement Analysis phase, where existing platforms such as Maideeasy and Kaodim were reviewed to identify common features and limitations. Informal observations and user expectations were also gathered to define the system's scope, which included modules for booking, cleaner assignment, feedback, and reporting.

In the System Design phase, diagrams were created using draw.io to visualize the system's architecture. These included flowcharts, context diagrams, Data Flow

Diagrams (DFDs), Entity-Relationship Diagrams (ERDs), and interface mockups. The database schema was designed to support core entities such as customer, booking, staff, and additional_service, ensuring logical relationships and data integrity.

The Implementation phase involved coding the system using PHP for backend logic and HTML/CSS for frontend structure and styling. Visual Studio Code was used as the development environment, and XAMPP provided the local server setup. MySQL was used to manage the database, allowing real-time interaction between system modules and stored data.

During the Testing phase, both unit and integration testing were conducted. Individual modules such as login, booking, and feedback were tested for correctness, while integration testing ensured that components interacted smoothly. Booking flows, cleaner assignment, and feedback submission were validated to confirm system reliability.

Finally, in the Deployment and Maintenance phase, the system was deployed to a live web server and connected to a remote database server, allowing real-time access and testing in an actual hosting environment. This deployment ensured that the system could be accessed by users through a public URL, simulating real-world usage beyond the local development setup. Maintenance tasks such as bug fixing, performance optimization, and database monitoring were initiated to ensure long-term stability and scalability. The deployment also allowed for external testing and feedback collection, which contributed to further refinement of the system's functionality.

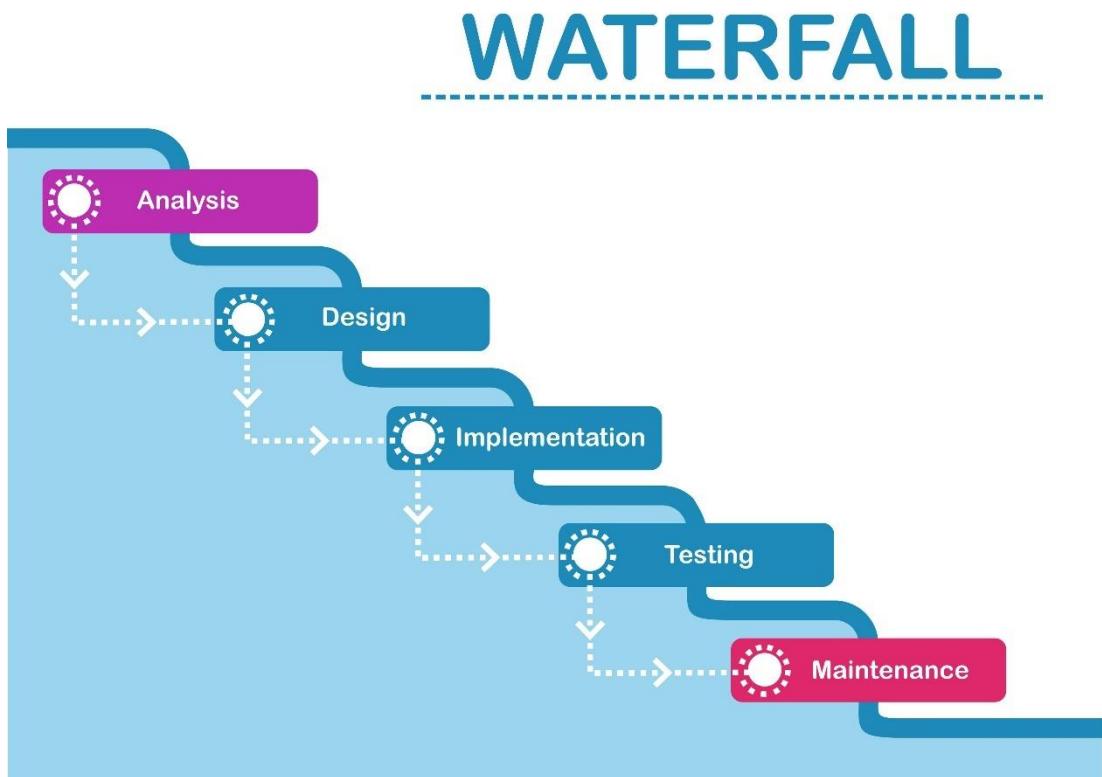


Figure 2.1: Waterfall Model of Software Development Life Cycle (SDLC)

2.3.2 Database Development Life Cycle (DBLC)

The development of HygieiaHub's database followed the Database Development Life Cycle (DBLC), which consists of several structured phases. The process began with the Initial Study, where the system's data requirements were identified based on the booking workflow, user roles, and reporting needs. This phase clarified the need for tables such as customer, booking, staff, payment, and feedback.

In the Database Design phase, the schema was modeled using ERDs and normalized to reduce redundancy and maintain relational integrity. Relationships between entities were carefully defined, such as the link between booking and staff through the booking_cleaner table, and between booking and additional_service via booking_service.

The Implementation and Loading phase involved creating the database using MySQL and populating it with sample data for testing purposes. SQL scripts were written to insert, update, and delete records, ensuring that the system could handle real-time data manipulation.

During the Testing and Evaluation phase, queries were executed to validate data accuracy and relational consistency. Complex joins and filters were tested to support reporting features, such as monthly sales summaries and customer booking history.

The Operation phase involved integrating the database with the PHP backend, enabling dynamic data retrieval and updates through user interactions. The system was tested in a local environment to simulate real-world usage.

Finally, the Maintenance phase includes ongoing tasks such as refining queries, optimizing performance, and preparing the database for future deployment. The DBLC approach ensured that the database was not only functional but also scalable and secure.

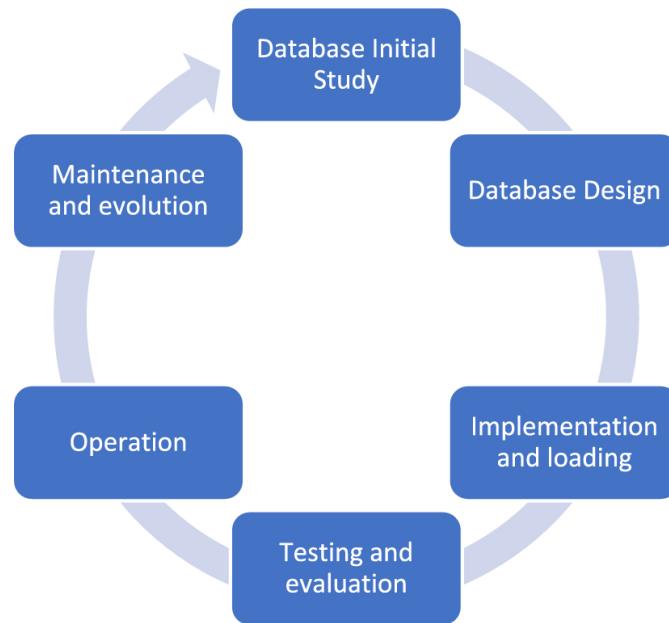


Figure 2.2: Model of Database Life Cycle (DBLC)

2.4 Project Schedule and Milestones

This section outlines the planned timeline and key milestones for the development of HygieiaHub, based on the Software Development Life Cycle (SDLC) phases and the PSM 2 academic structure. The schedule was designed to ensure that each phase of development from initial planning to deployment and demonstration was completed systematically and within the allocated timeframe. The activities were distributed across 15 weeks, with clear deliverables aligned to reporting deadlines and evaluation checkpoints.

2.4.1 Milestones

The following table presents the major milestones achieved throughout the project, mapped to the SDLC phases and PSM 2 requirements:

Table 2.2: Milestones

No	Milestone Description	SDLC Phase	Expected Outcome / Document	Timeline
1	Proposal assessment and verification	Planning	a) Project proposal form b) Initial problem statement c) Flowchart of current system	16 - 21 March 2025
2	Completion of Chapter 1	Requirement Analysis	a) Chapter 1: Introduction b) Project background and objectives c) Problem statement	22 - 28 March 2025
3	Completion of Chapter 2	Requirement Analysis	a) Chapter 2: Literature Review & Methodology b) Comparison table c) SDLC & DBLC write-up	29 March - 4 April 2025
4	Completion of Chapter 3	System Design	a) Chapter 3: System Analysis b) Functional and non-functional requirements c) Use case diagrams	5 - 11 April 2025
5	Completion of Chapter 4	System Design	a) Chapter 4: System Design b) ERD and DFD diagrams c) Interface mockups	12 - 18 April 2025
6	Database creation and initial system coding	Implementation	a) MySQL schema setup b) PHP backend structure c) Basic interface integration	19 - 25 April 2025

7	Module development: Booking and cleaner assignment	Implementation	a) Booking form b) Auto-assignment logic c) Booking validation and storage	26 April - 2 May 2025
8	Module development: Feedback and reporting	Implementation	a) Feedback form b) Rating storage c) Monthly sales query and report logic	3 - 9 may 2025
9	System deployment to UTeM server	Implementation	a) Live system hosted b) Remote database connection c) URL access confirmation	10 - 16 May 2025
10	Internal testing and bug fixing	Testing	a) Unit testing b) Integration testing c) Bug tracking sheet	17 - 23 May 2025
11	Draft report compilation	Testing	a) Full draft report b) Supervisor feedback c) Correction checklist	24 - 30 May 2025
12	Final report refinement	Testing	a) Corrected report b) Abstract and references c) Formatting and compliance check	31 May - 6 June 2025
13	Final presentation preparation	Deployment	a) Slide deck b) Demo script c) Presentation rehearsal	7 - 13 June 2025
14		Deployment	a) Live system walkthrough	

	Final presentation and system demonstration		b) Evaluation form c) Supervisor comments	14 - 20 June 2025
15	Final report submission and closure	Maintenance	a) Printed and bound report b) Digital submission c) Project closure documentation	21 - 23 June 2025
16	Final system deployment	Implementation	a) HygieiaHub deployed remotely b) Stable functionality c) Live URL confirmation	21-Jul-25
17	System testing and validation	Testing	a) Functional and non-functional testing b) SUS feedback collected c) Bug resolution	22 - 29 July 2025
18	Chapter 6: Testing & Validation Documentation	Testing	a) Test strategy and plan b) Test results and discussion c) SUS analysis	30 July - 4 August 2025
19	Chapter 7: Project Conclusion	Maintenance	a) Final chapter write-up b) Limitations and future work c) Supervisor review	5 - 9 August 2025
20	Final report compilation and formatting	Maintenance	a) All chapters compiled b) Formatting aligned to template c) Final proofreading	10 - 15 August 2025
21	PSM II Draft Report Submission	Maintenance	a) Draft report submitted b) Supervisor and evaluator review	18-Aug-25

			c) Log record updated	
22	Final presentation preparation	Deployment	a) Slide deck finalized b) Demo rehearsed c) Presentation schedule confirmed	19 - 22 August 2025
23	Final presentation and evaluation	Deployment	a) Oral presentation delivered b) Evaluator feedback c) Supervisor scoring	25 - 29 August 2025
24	Final report correction and refinement	Maintenance	a) Corrections based on feedback b) Final formatting c) Plagiarism check	1 - 5 September 2025
25	Final report submission and closure	Maintenance	a) Report uploaded to e-Thesis b) Logbook and score sheet submitted c) Supervisor approval	8 - 12 September 2025

These milestones reflect the structured progression of the project, ensuring that documentation and system development were synchronized with academic deliverables and supervisor evaluations.

2.4.2 Gantt Chart

The Gantt chart for HygieiaHub illustrates the distribution of activities across the 15-week development period. It includes report writing phases, system deployment, and presentation preparation. Each activity is aligned with the corresponding SDLC phase to ensure clarity and traceability. The chart also highlights overlapping tasks such as documentation and testing, which were conducted concurrently to optimize time and resource usage.

SDLC Phase	Activity	Timeline																			
		16/3/2025	21/3/2025	22/3/2025	4/4/2025	5/4/2025	18/4/2025	19/4/2025	16/5/2025	31/5/2025	6/6/2025	7/6/2025	13/6/2025	17/7/2025	21/7/2025	22/7/2025	23/7/2025	4/8/2025	19/8/2025	29/8/2025	1/9/2025
Planning	Proposal assessment, problem identification, initial flowchart																				
Requirement Analysis	Chapter 1 (Introduction), Chapter 2 (Literature Review & Methodology)																				
System Design	Chapter 3 (System Analysis), Chapter 4 (System Design), ERD, DFD, interface mockups																				
Implementation	Database setup, backend coding, booking module, feedback module, system deployment																				
Testing	Internal testing, SUS distribution, functional & non-functional testing, Chapter 6 documentation																				
Deployment	Final presentation prep, system demo, Chapter 7 write-up, slide deck, oral presentation																				
Maintenance	Final report correction, formatting, e-Thesis submission, logbook closure																				

Figure 2.3: Gantt Chart

2.5 Conclusion

The implementation of HygieiaHub followed a structured methodology combining the Software Development Life Cycle (SDLC) and Database Development Life Cycle (DBLC), beginning with requirement analysis and system design, followed by coding, testing, and deployment. Development tools such as PHP, MySQL, HTML/CSS, and XAMPP were used to build and integrate system modules, while draw.io supported the creation of ERDs, DFDs, and interface mockups. The system was successfully deployed to a live web server and connected to a remote database server, enabling real-time access and external testing. Technical challenges such as cleaner auto-assignment logic, booking status consistency, and report filtering were resolved through iterative testing and backend validation. The system now supports core functionalities including customer registration, booking coordination, feedback submission, and staff-side reporting. HygieiaHub is fully operational and meets the functional requirements outlined during the planning phase, offering a streamlined experience for both customers and staff. Future enhancements may include support for multi-provider models, push notifications, and analytics dashboards to further improve usability and scalability.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter presents a comprehensive analysis of the current challenges faced by local cleaning service providers and outlines the rationale for developing HygieiaHub. The analysis process translates the findings from the literature review and project objectives into detailed system specifications, focusing on both user expectations and technical feasibility. It begins by examining the limitations of existing platforms and manual workflows, followed by a breakdown of the proposed system's structure and requirements. This chapter also introduces visual models such as Data Flow Diagrams (DFDs) to illustrate how data moves through the system and how users interact with its components. By identifying the core problems and mapping out the system's logic, this analysis serves as a critical foundation for the design and implementation phases that follow.

3.2 Problem Analysis

Many existing cleaning service platforms are designed with scalability in mind, often serving users across multiple regions and supporting multiple vendors. However, this broad focus results in systems that are unnecessarily complex for small, local service providers. One of the main issues is the presence of layered navigation and excessive features, which can confuse users especially first-time visitors or elderly customers who are less familiar with digital interfaces. These platforms often include AI-driven scheduling, multi-company support, and subscription models that add to system overhead without benefiting single-provider operations.

Another major challenge is payment processing. Most platforms prioritize online or card-based payments, which may not be suitable for users in areas where Cash on Delivery (COD) remains the preferred method. For customers who value simplicity and direct payment upon service completion, digital-only models create a barrier to adoption and trust. Additionally, booking management is often inefficient in systems that lack proper administrative tools. Without a straightforward mechanism for assigning cleaners to appointments or tracking service progress, staff may struggle to maintain operational clarity and responsiveness.

The absence of internal reporting tools further limits the ability of service providers to monitor performance, analyze feedback, and make informed decisions. These gaps highlight the need for a system that is streamlined, role-specific, and optimized for local use. HygieiaHub is designed to address these challenges by focusing only on essential functions such as booking, cleaner assignment, feedback submission, and internal reporting ensuring a smoother and more transparent service experience for both customers and staff.

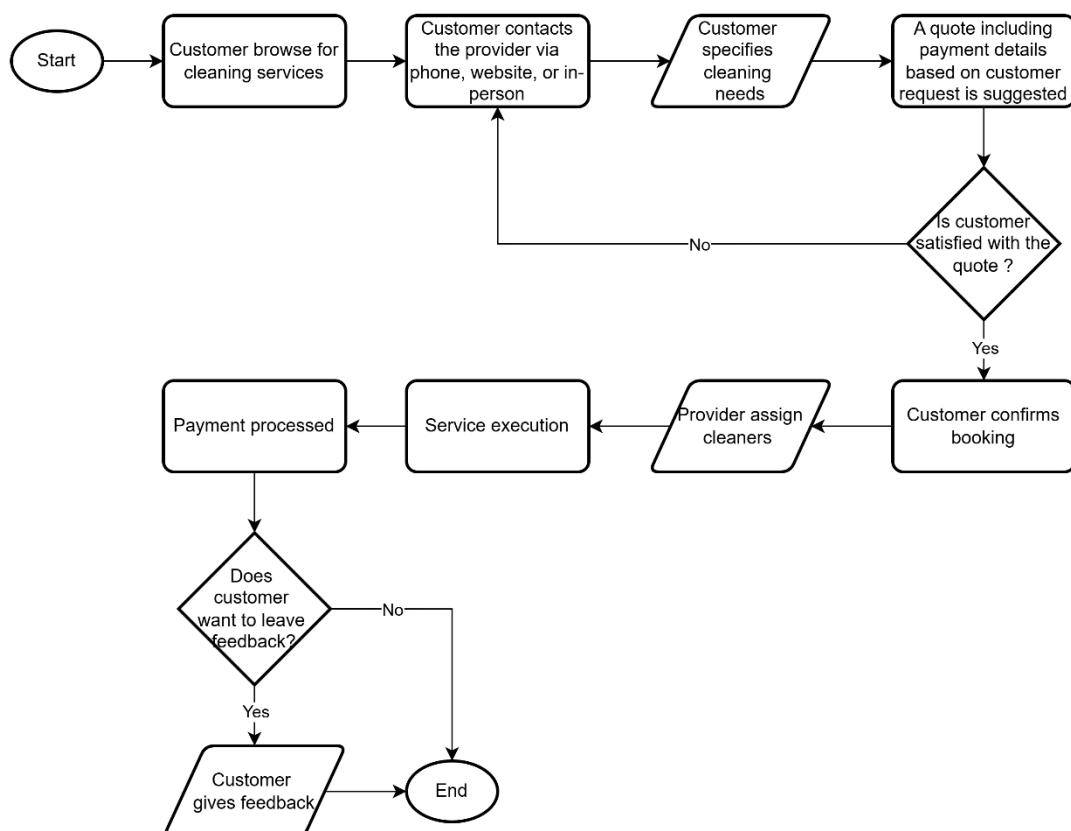


Figure 3.1: Current system flowchart

3.3 Proposed System

HygieiaHub is a web-based cleaning service booking system designed to address the limitations identified in existing platforms and manual workflows. The system focuses on simplifying the booking process for both customers and staff by offering a streamlined, role-specific interface and automating key operations such as cleaner assignment and payment calculation. Unlike multi-provider platforms that introduce unnecessary complexity, HygieiaHub is tailored for single-provider use, ensuring that only essential features are included to reduce user confusion and improve operational clarity.

Customers can register, log in, select services, and submit bookings with optional add-ons, while staff can manage bookings, update statuses, and generate performance reports. The system supports Cash on Delivery (COD) as its primary payment method, reflecting the preferences of users in localized service contexts. Feedback submission is integrated into the booking flow, allowing customers to rate and comment on completed services, and enabling staff to monitor service quality. The system is developed using PHP for backend logic, MySQL for database management, and HTML/CSS for frontend design, with Visual Studio Code as the development environment and XAMPP as the local server stack.

Upon completion, the system was successfully deployed to a live web server and connected to a remote database server, enabling real-time access and external testing. HygieiaHub is designed to be accessible via desktop browsers, with a responsive layout that ensures usability across devices. By focusing on simplicity, automation, and transparency, the proposed system offers a practical solution for small-scale cleaning service providers seeking to digitize their operations without adopting overly complex platforms.

3.4 Requirements Analysis

The requirements for HygieiaHub were identified through a combination of user expectation analysis, observation of existing service platforms, and alignment with the

project's specific goals. The aim was to design a system that is both user-friendly and efficient for localized cleaning service providers, emphasizing simplicity, reliability, and ease of use. The requirements are divided into functional and non-functional categories to ensure that both system capabilities and quality attributes are clearly addressed. In addition, technical requirements are outlined to specify the tools and platforms used during development. Visual models such as the context diagram and Data Flow Diagrams (DFDs) are included to illustrate how data flows through the system and how users interact with its components.

3.4.1 Functional Requirements

Functional requirements define the essential operations that HygieiaHub must support to fulfill its intended purpose. These requirements were identified through analysis of user expectations, existing service platforms, and the specific goals of the project. The system is designed to support customer registration, service booking, cleaner assignment, payment processing, feedback submission, and staff-side reporting. Each function is mapped to a process within the system and visualized through diagrams to clarify how data flows between users, system components, and the database.

Table 3.1: Functional Requirements

No	Functional Requirement	Description
FR01	User Registration and Login	System shall allow customers and staff to register and log in using a valid email and password. Only registered users are granted access.
FR02	Profile Update	System shall allow users to update their profile details including phone number, email, and password. Customers may also update their address.
FR03	House Type and Service Management	System shall allow staff to create, update, and delete house types and associated cleaning services.

FR04	Booking Cleaning Services	System shall allow customers to browse services, select booking options, and confirm appointments. The system shall auto-assign cleaners based on availability.
FR05	Booking Management	System shall allow staff to view and update booking statuses, and reassign cleaners. Customers shall be able to view and cancel their bookings.
FR06	Payment Processing	System shall calculate total service cost based on booking details and notify customers to prepare COD payment. Staff shall update payment status after service completion.
FR07	Customer Feedback and Ratings	System shall allow customers to submit feedback and ratings for completed services. Staff shall be able to view and assess feedback.
FR08	Reporting and Analytics	System shall allow staff to generate daily, monthly, and yearly reports on service sales, staff performance, and customer feedback.

3.4.1.1 Context Diagram

The context diagram provides a high-level overview of HygieiaHub as a single unified system. It illustrates how external entities such as customers, staff, and the database interact with the system through various input and output flows. Customers submit booking requests, provide feedback, and receive invoices, while staff manage bookings, update statuses, and generate reports. The system communicates with the database to store and retrieve data related to users, bookings, services, and feedback. This diagram helps define the system boundary and clarifies the roles of each external entity in relation to the system's core processes.

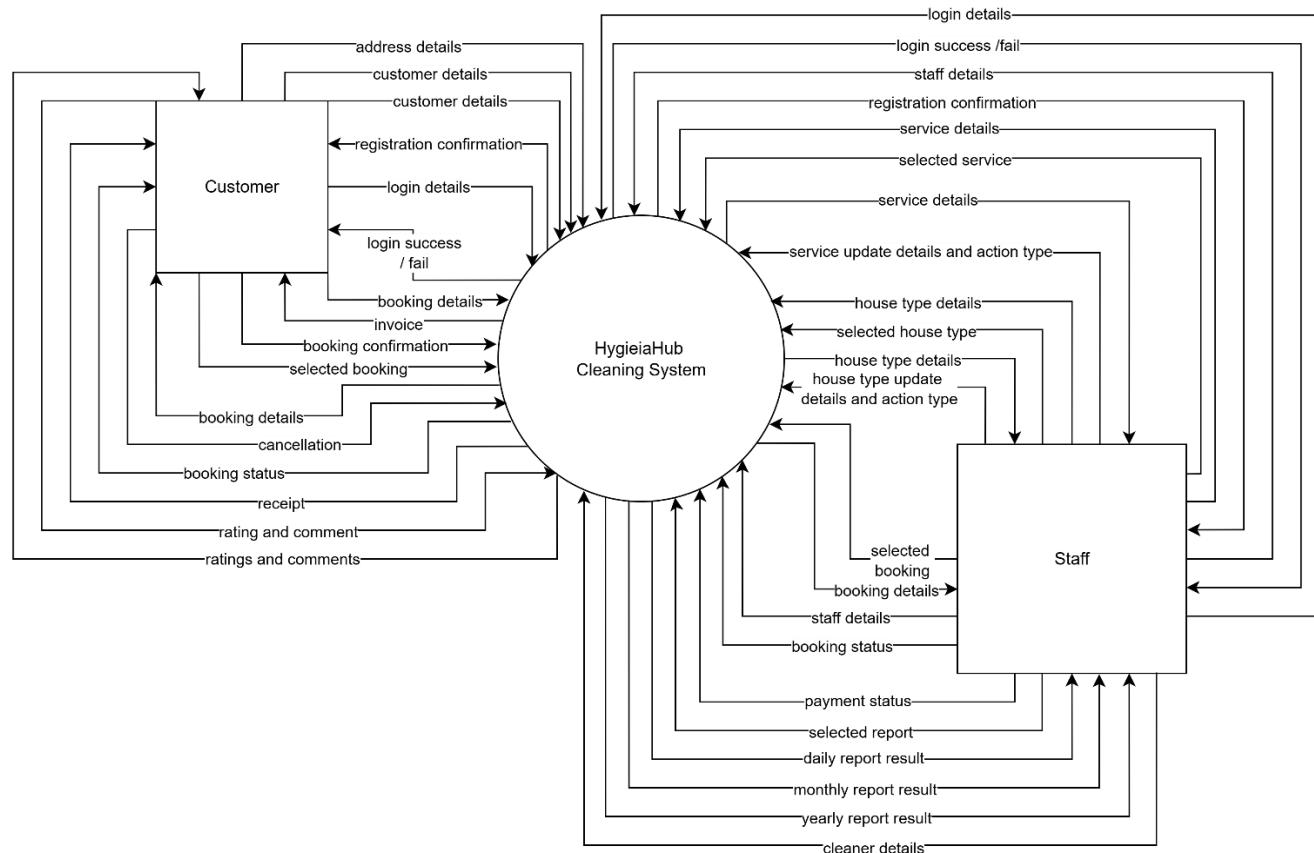


Figure 3.2: Context Diagram

3.4.1.2 Data Flow Diagram Level 0

The Level 0 Data Flow Diagram (DFD) provides a modular overview of HygieiaHub's internal processes and how they interact with external entities such as customers, staff, and the database. This diagram breaks the system into major functional components, including registration and login, booking management, payment processing, feedback submission, and reporting. Each process is represented as a distinct module, showing how data flows between users and the system. For example, customers initiate the booking process by selecting services and submitting booking details, which are then processed by the system and stored in the database. Staff access booking records, update statuses, and generate reports based on stored data. The diagram also illustrates how feedback is submitted by customers and retrieved by staff for service evaluation. This level of abstraction helps clarify the system's core structure and the relationships between its main processes.

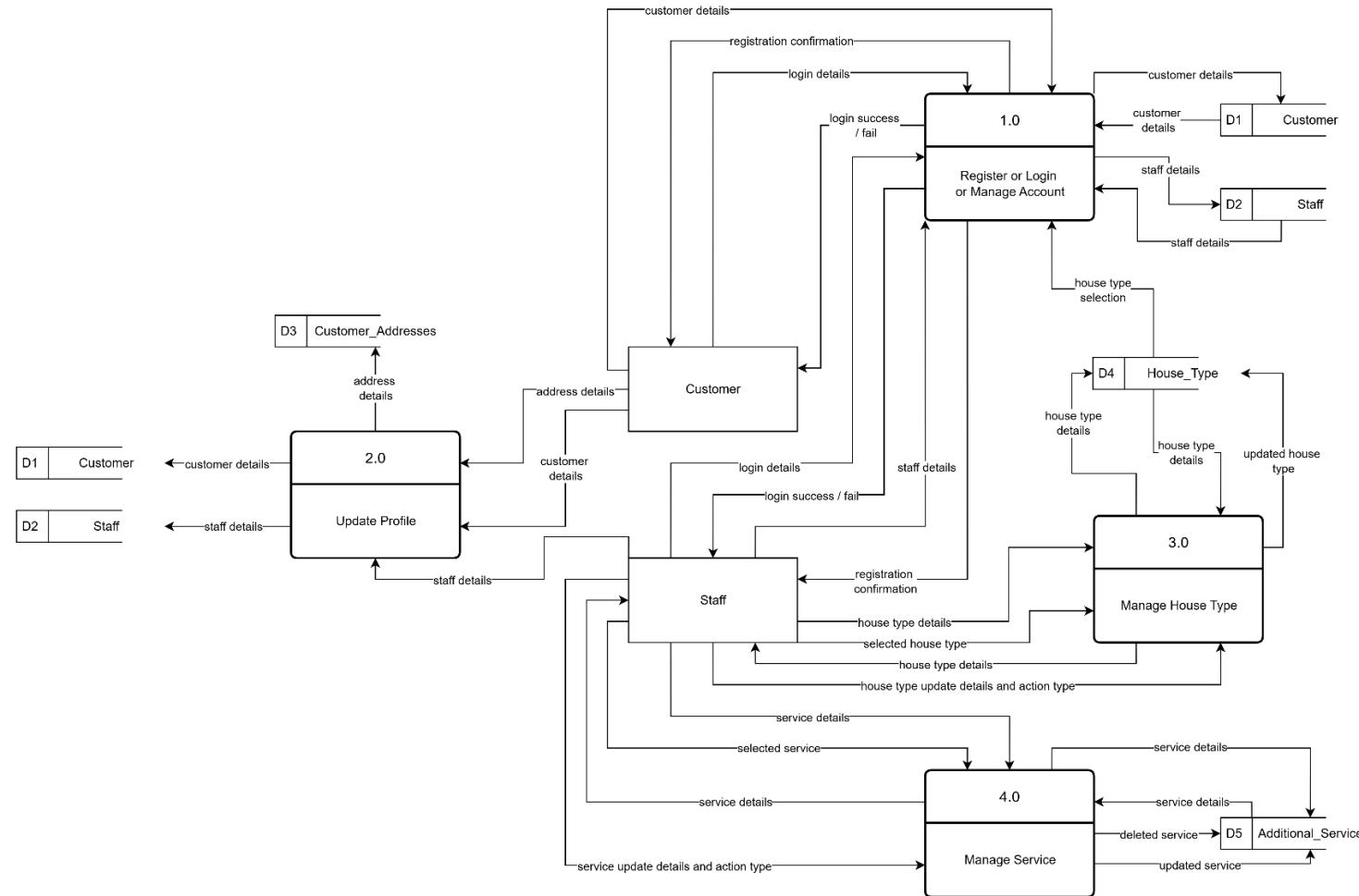


Figure 3.3: Level 0 of DFD for Process 1, 2, 3, and 4

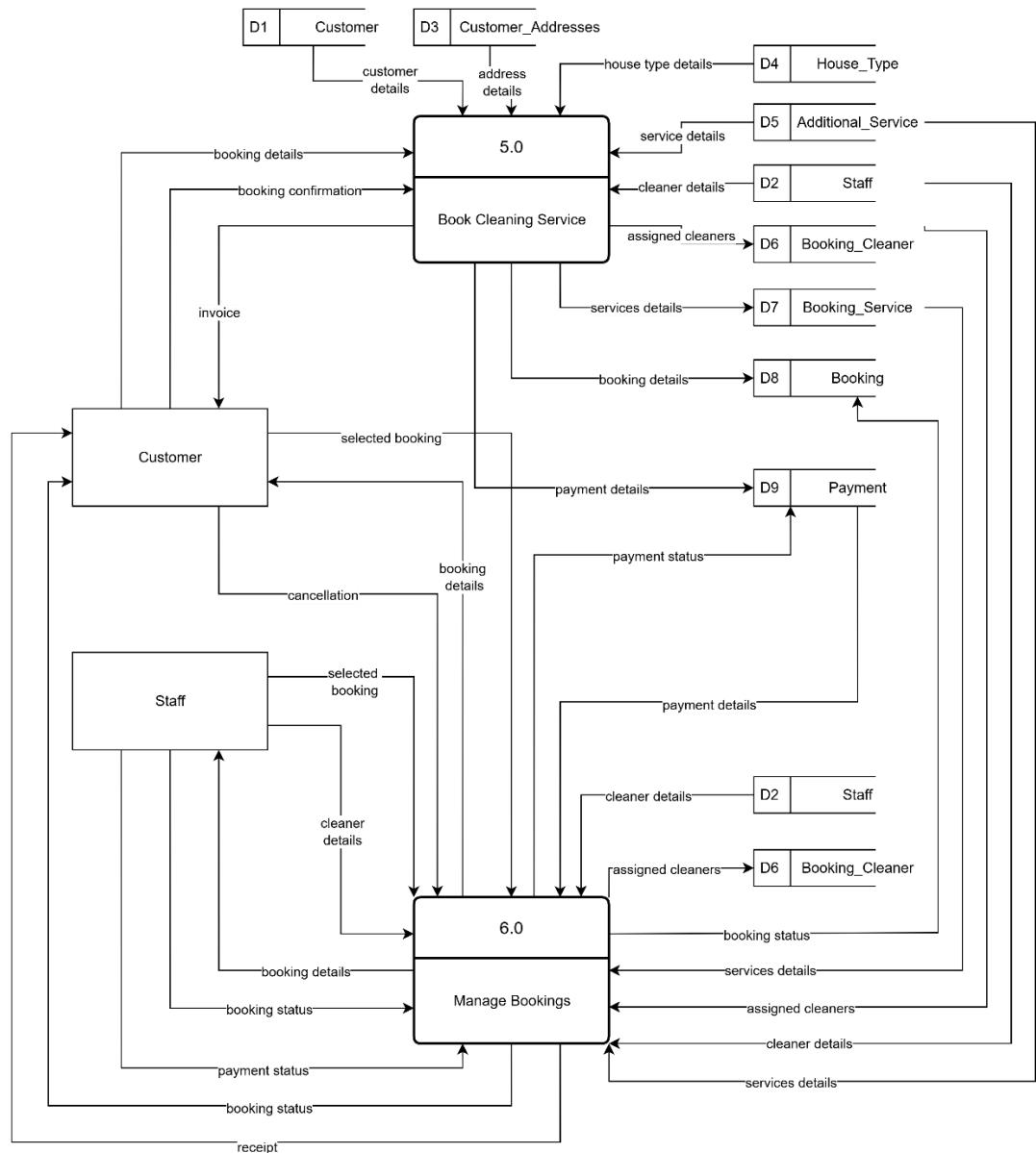


Figure 3.4: Level 0 of DFD for Process 5 and 6

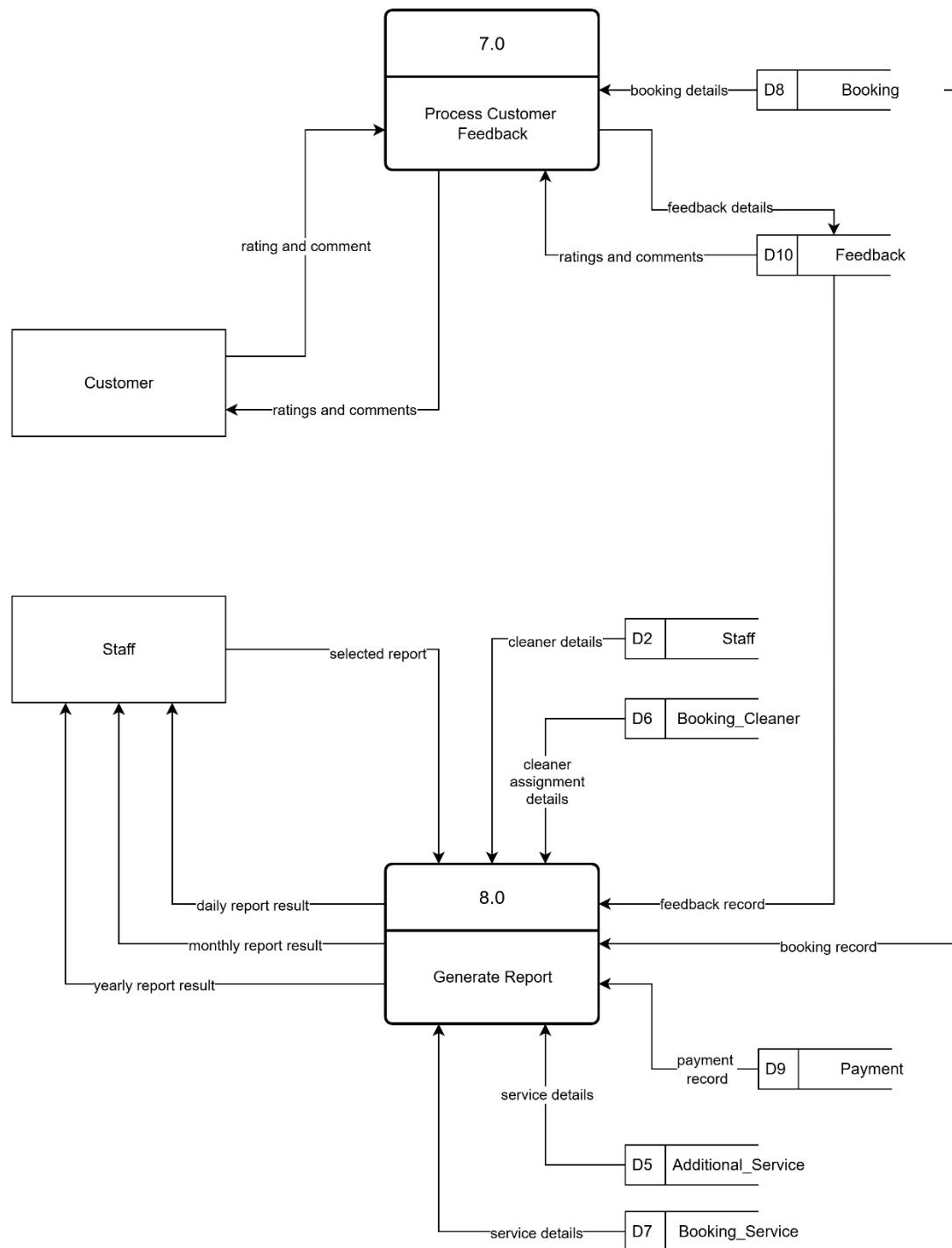


Figure 3.5: Level 0 of DFD for Process 7 and 8

3.4.1.3 Data Flow Diagram Level 1

The Level 1 Data Flow Diagram expands on the processes introduced in Level 0 by decomposing them into more detailed sub-processes. For instance, the booking management module is broken down into service selection, cleaner assignment, and invoice generation. The registration and login process includes account creation, credential validation, and session handling. Payment processing is detailed into cost calculation, invoice display, and status update by staff. Feedback submission includes rating input, comment storage, and feedback retrieval. Each sub-process is connected to relevant data stores such as customer, booking, staff, payment, and feedback, showing how data is created, modified, and accessed throughout the system. This diagram provides a granular view of system logic, helping to validate the completeness and consistency of the design before implementation.

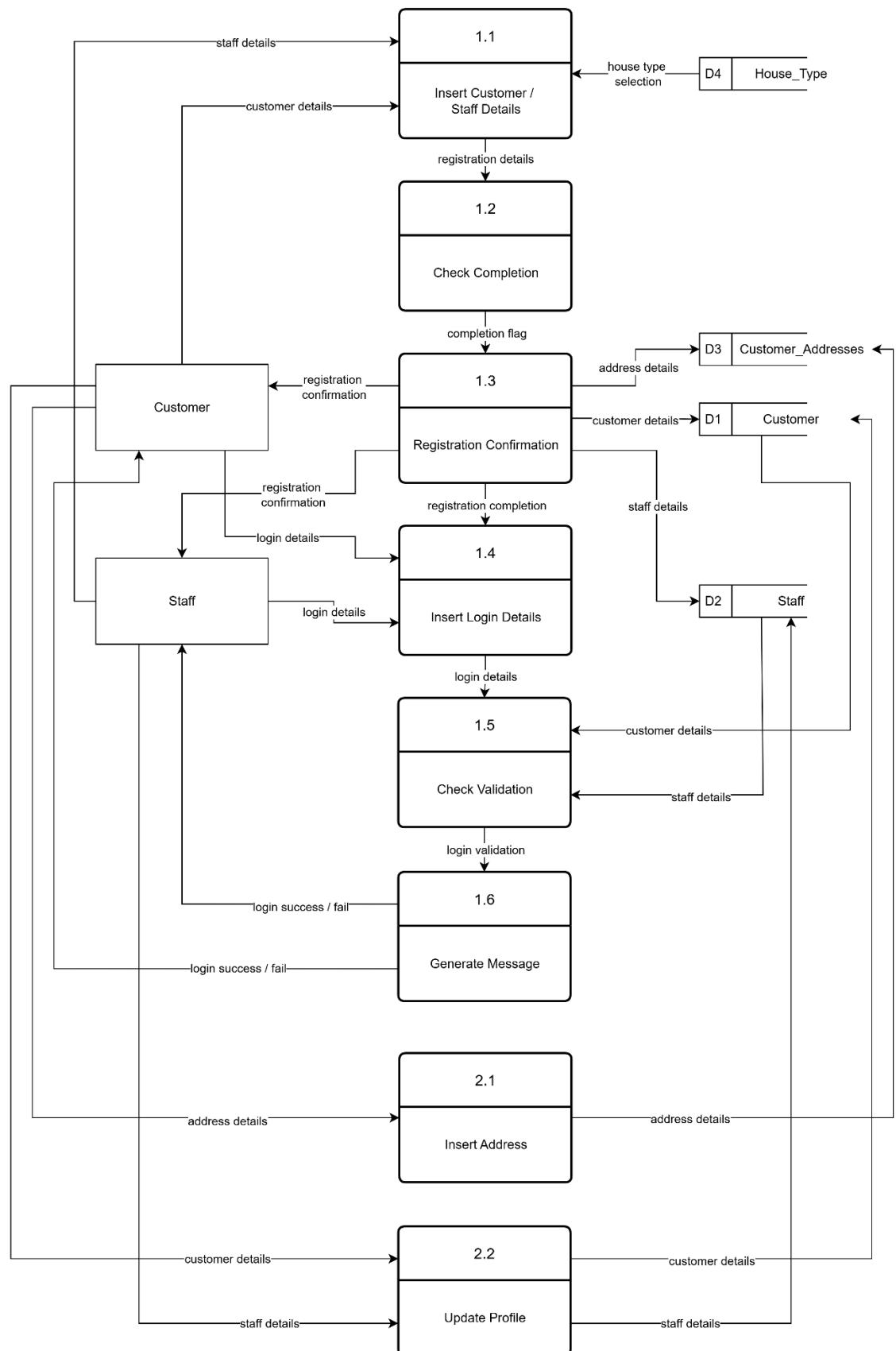


Figure 3.6: Level 1 of DFD for Process 1 and 2

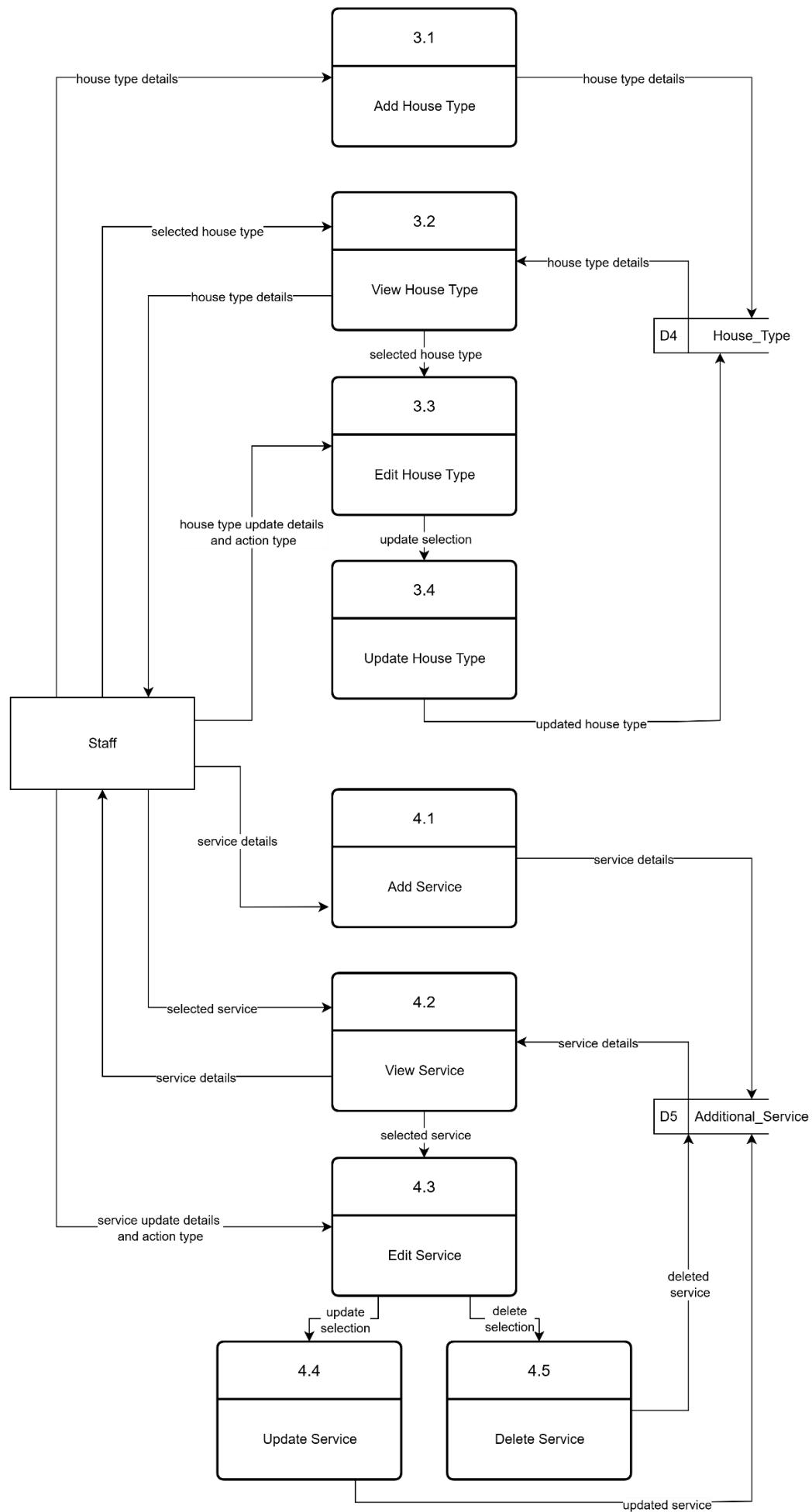


Figure 3.7: Level 1 of DFD for Process 3 and 4

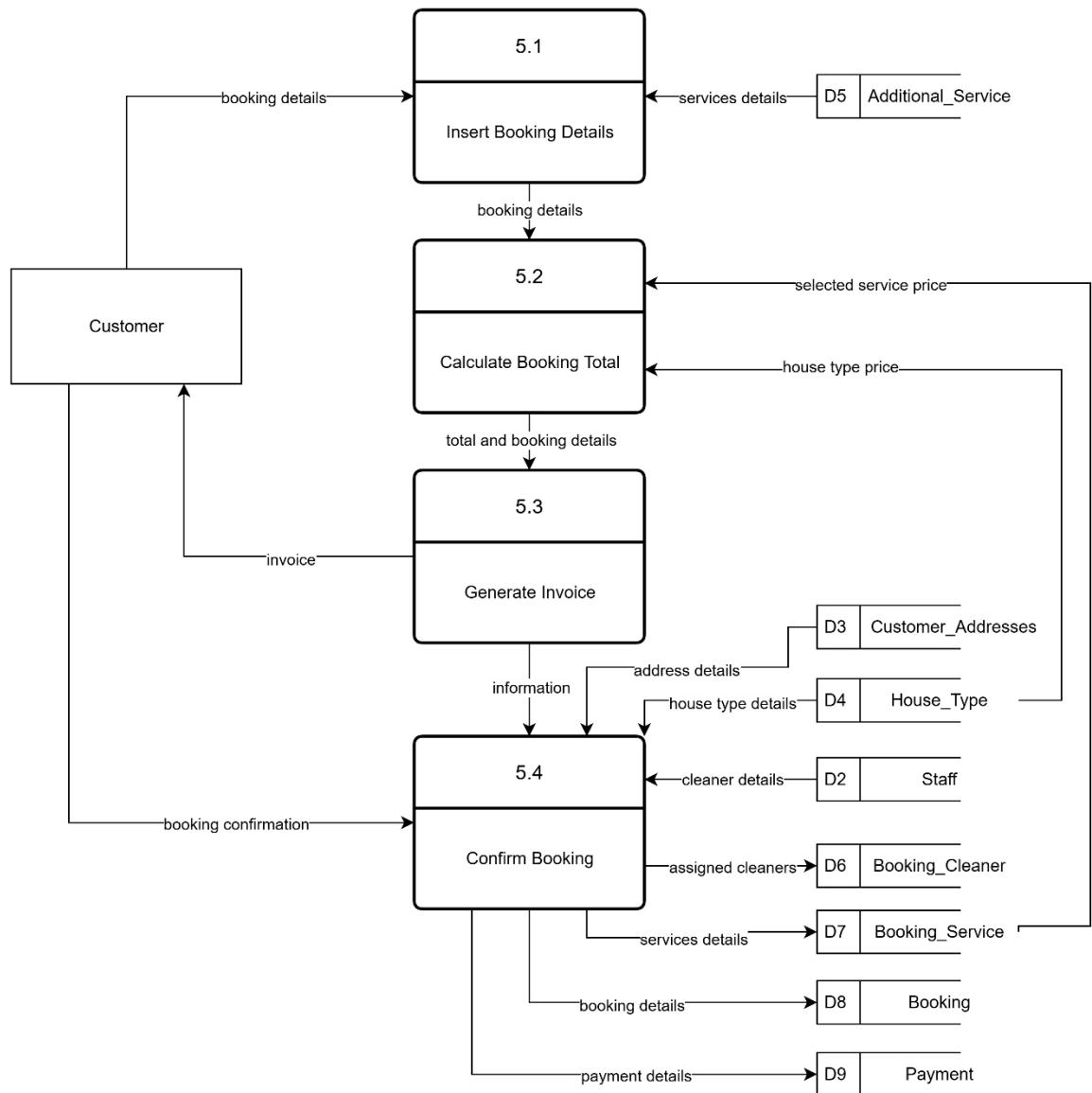


Figure 3.8: Level 1 of DFD for Process 5

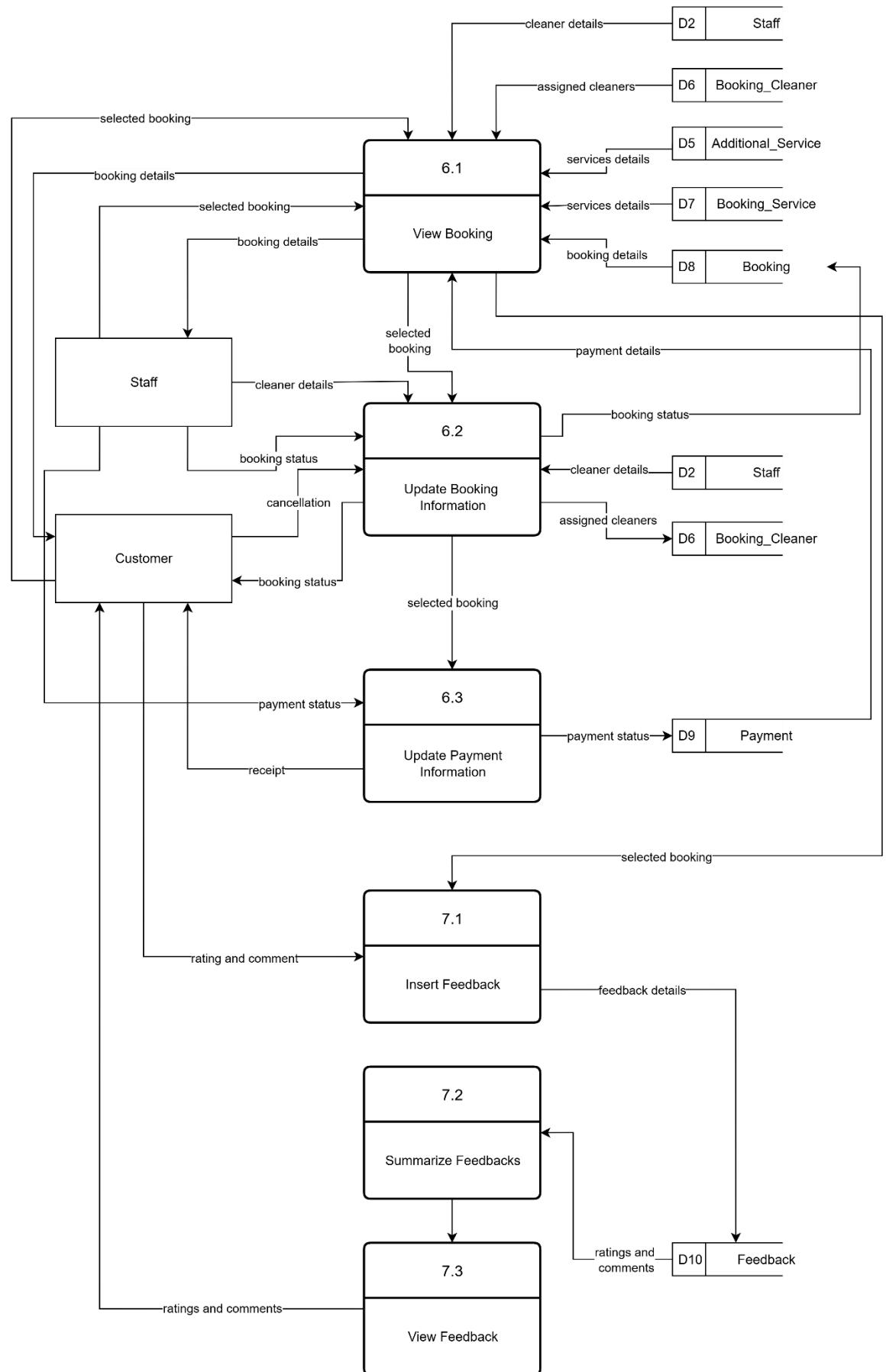


Figure 3.9: Level 1 of DFD for Process 6 and 7

3.4.2 Non-Functional Requirements

Non-functional requirements define the quality attributes and operational standards that HygieiaHub must meet to ensure a reliable and user-friendly experience. Unlike functional requirements, which focus on specific system actions, non-functional requirements address broader aspects such as system availability, usability, performance, and security. These requirements were derived from user expectations, system scope, and best practices in web-based application design. By adhering to these standards, the system ensures that both customers and staff can interact with its features efficiently, securely, and without unnecessary complexity.

Table 3.2: Non-Functional Requirements

No	Non-Functional Requirement	Description
NFR01	System Availability	The system should be available 24/7 to allow users to access booking and feedback modules at any time.
NFR02	Usability	The system shall provide easy navigation and a user-friendly interface for both customers and staff.
NFR03	Accessibility	The system shall be optimized for desktop browser access with responsive layout across screen sizes.
NFR04	Performance	The system shall ensure fast loading times and efficient query execution to support smooth user interaction.
NFR05	Security	The system shall implement secure login mechanisms, password protection, and input validation to prevent unauthorized access.
NFR06	Simplicity	The system shall include only core functions relevant to the service workflow, avoiding unnecessary features that may confuse users.

3.4.3 Technical Requirements

To ensure the successful development, testing, and deployment of HygieiaHub, a set of technical requirements was established. These requirements include both hardware and software specifications that support the system's functionality and performance. The selected tools and configurations were chosen based on compatibility with PHP and MySQL, ease of use in academic environments, and suitability for browser-based service platforms. The following subsections outline the minimum hardware specifications and the software tools used throughout the project lifecycle.

3.4.3.1 Hardware Requirements

The system was developed and tested on standard computing hardware to ensure accessibility and feasibility for student-level deployment. The minimum specifications include an Intel i3 processor or higher, at least 4 GB of RAM to support local server operations, and 50 GB of available storage space for code files, database records, and testing logs. These specifications were sufficient to run XAMPP, execute PHP scripts, and manage MySQL queries without performance issues.

Table 3.3: Hardware Requirements

Component	Minimum Specification
Processor	Intel i3 or higher
RAM	4 GB minimum
Storage	50 GB available space

3.4.3.2 Software Requirements

The software tools used in this project were selected for their compatibility with web-based development and ease of integration. Visual Studio Code served as the primary code editor, offering syntax highlighting and extension support for PHP and HTML. XAMPP was used to simulate a local server environment, combining

Apache and MySQL for backend and database operations. Draw.io was utilized to create system diagrams such as ERDs, DFDs, and flowcharts, while Google Chrome was used for browser-based testing and interface validation. These tools collectively supported the full development lifecycle, from design to deployment.

Table 3.4: Software Requirements

Software	Purpose
Visual Studio Code	Code development environment
XAMPP (MySQL, Apache)	Web server and database management
Draw.io	System design diagrams
Google Chrome	System testing (browser)

3.5 Conclusion

In conclusion, this chapter has outlined the analytical foundation for the development of HygieiaHub, beginning with the identification of existing system limitations and progressing through the proposed solution, requirement analysis, and technical specifications. The system was designed to address specific operational challenges faced by single-provider cleaning services, with a focus on simplicity, automation, and role-based access. Functional and non-functional requirements were clearly defined, supported by visual models such as context diagrams and data flow diagrams to illustrate system logic and data movement. Technical requirements, including both hardware and software specifications, were selected to ensure compatibility, accessibility, and ease of deployment. Together, these elements provide a comprehensive blueprint for the system's design and implementation, ensuring that HygieiaHub meets its intended goals effectively.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter presents the complete design phase of HygieiaHub, covering both high-level and detailed system components that were planned prior to implementation. The design process includes the system architecture, data modeling, and graphical user interface layout, all of which were developed to meet the functional and non-functional requirements identified in Chapter 3. The system architecture outlines the structural framework of the application, showing how different layers interact to support user operations. The data model design includes conceptual, logical, and physical representations of the database, supported by diagrams and structured documentation such as the Entity Relationship Diagram (ERD), business rules, data dictionary, and Data Definition Language (DDL) scripts. The graphical user interface design illustrates how users interact with the system through structured layouts and navigation flows. Each section in this chapter is supported by relevant diagrams and tables to ensure clarity and traceability, forming a comprehensive blueprint for the system's development.

4.2 System Architecture

HygieiaHub is developed as a web-based application using a client-server architecture. The system is structured into three main layers: the presentation layer, the application layer, and the data layer. The presentation layer is responsible for rendering the user interface, allowing customers and staff to interact with the system through a browser. It is built using HTML and CSS to ensure responsive design and ease of

navigation. The application layer handles the business logic and system operations, developed using PHP to process user inputs, manage bookings, assign cleaners, and generate reports. This layer acts as the intermediary between the user interface and the database. The data layer consists of a MySQL database that stores all system data, including user profiles, booking records, service details, feedback, and payment statuses. Communication between layers is facilitated through HTTP requests, with data validation and session handling implemented to ensure secure and reliable operations. The system was initially developed and tested using XAMPP in a local environment and later deployed to a live web server with a remote database connection, enabling real-time access and external testing. This layered architecture ensures modularity, scalability, and maintainability, allowing future enhancements to be implemented without disrupting core functionality.

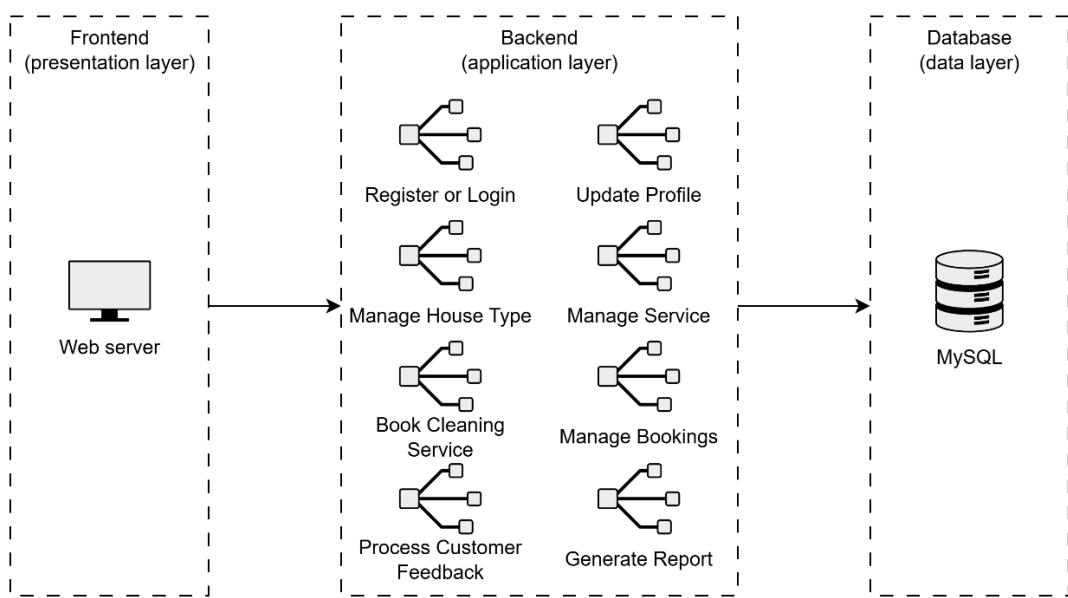


Figure 4.1: System Architecture of HygieiaHub

4.3 Data Model Design

This section presents the data model design for HygieiaHub, which defines how data is structured, stored, and accessed throughout the system. The design process includes conceptual, logical, and physical representations of the database, ensuring that all functional requirements are supported by a scalable and normalized data

structure. The conceptual design is illustrated through an Entity Relationship Diagram (ERD), followed by business rules that describe the relationships and constraints between entities. The logical design is documented through a detailed data dictionary, and the physical design includes Data Definition Language (DDL) scripts and a list of database objects used in the system.

4.3.1 Conceptual Design

The conceptual design of HygieiaHub's database is represented using a normalized Entity Relationship Diagram (ERD). This diagram outlines the core entities such as Customer, Booking, House_Type, Service, Staff, Feedback, and their relationships. It ensures that data redundancy is minimized and that each entity is logically connected to support system operations such as booking coordination, cleaner assignment, and feedback tracking.

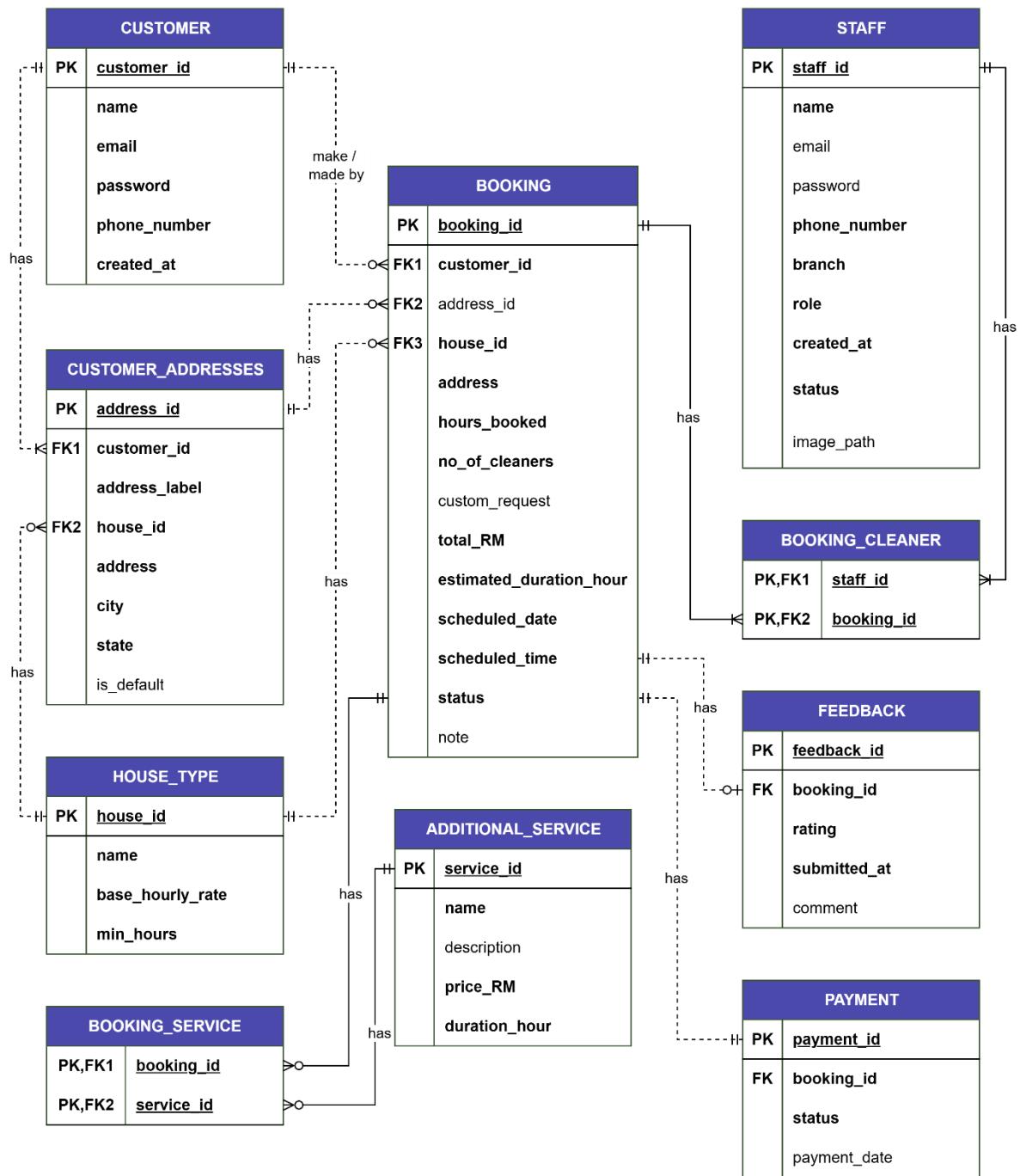


Figure 4.2: Entity Relationship Diagram

4.3.1.1 Business Rules

Based on the ERD shown in Figure 4.2: Entity Relationship Diagram, the following business rules define how entities relate to one another and ensure data

integrity across the system. These rules were derived from the system's functional scope and are essential for guiding database normalization and enforcing relational constraints.

1. A customer can have one or many addresses while an address may belong to only one customer.
2. A registered customer can make none or many bookings. While a booking can be made by only one customer.
3. An address can have only one type of house while a type of house can have none or many addresses.
4. A booking may be for only one type of house while a type of house may have none or many types of houses.
5. A booking can have zero or many additional services while an additional service may be in zero or many bookings.
6. A booking may have one or many cleaners while a cleaner may have one or many bookings.
7. A booking can have only one payment and a payment can have only one booking.
8. A feedback may be referred to only one booking and a booking may have none or only one feedback.

4.3.2 Logical Design

The logical design phase involves documenting the structure of each database table in detail. The data dictionary includes table names, field names, data types, lengths, primary and foreign key indicators, default values, and null constraints. This documentation ensures consistency and clarity during development and serves as a reference for future maintenance or integration.

Table 4.1: Data Dictionary of Table customer

CUSTOMER								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
customer_id	int	11		yes	yes		PK	
name	varchar	100		yes				
email	varchar	100		yes				
password	varchar	255		yes				
phone_number	varchar	11		yes				
created_at	datetime		yyyy-mm-dd hh:mm:ss	yes	current date			

Table 4.2: Data Dictionary of Table house_type

HOUSE_TYPE								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
house_id	int	11		yes	yes		PK	
name	varchar	100		yes				
base_hourly_rate	decimal	5,2		yes				
min_hours	decimal	3,2		yes				

Table 4.3: Data Dictionary of Table customer_addresses

CUSTOMER_ADDRESSES								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
address_id	int	11		yes	yes		PK	
customer_id	int	11		yes			FK	CUSTOMER
address_label	varchar	50		yes				
house_id	int	11		yes			FK	HOUSE_TYPE
address	varchar	300		yes				
city	varchar	100		yes				
state	varchar	100		yes				
is_default	varchar	50				1		

Table 4.4: Data Dictionary of table staff

STAFF								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
staff_id	int	11		yes	yes		PK	
name	varchar	100		yes				
email	varchar	100						
password	varchar	255						
phone_number	varchar	11		yes				
branch	varchar	100		yes				
role	varchar	7		yes				
created_at	datetime		yyyy-mm-dd hh:mm:ss	yes	current date			
status	varchar	9		yes	Active			
image_path	varchar	255						

Table 4.5: Data Dictionary of Table additional_service

ADDITIONAL_SERVICE								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
service_id	int	11		yes	yes		PK	
name	varchar	100		yes				
description	varchar	300						
price_RM	decimal	5,2		yes				
duration_hour	decimal	3,2		yes				

Table 4.6: Data Dictionary of Table booking

BOOKING								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
booking_id	int	11		yes	yes		PK	
customer_id	int	11		yes			FK	CUSTOMER
address_id	int	11					FK	CUSTOMER_ADDRESSES
house_id	int	11		yes			FK	HOUSE_TYPE
address	varchar	1000		yes				
hours_booked	decimal	4,2		yes				
no_of_cleaners	int	2		yes				
custom_request	varchar	300						
total_RM	decimal	6,2		yes				
estimated_duration_hour	decimal	4,2		yes				
scheduled_date	date		yyyy-mm-dd	yes				
scheduled_time	time		hh:mm:ss	yes				
status	varchar	9		yes				
note	varchar	300						

Table 4.7: Data Dictionary of Table booking_service

BOOKING_SERVICE								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
booking_id	int	11		yes			PK, FK	BOOKING
service_id	int	11		yes			PK, FK	ADDITIONAL_SERVICE

Table 4.8: Data Dictionary of Table booking_cleaner

BOOKING_CLEANER								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
booking_id	int	11		yes			PK, FK	BOOKING
staff_id	int	11		yes			PK, FK	STAFF

Table 4.9: Data Dictionary of Table payment

PAYMENT								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
payment_id	int	11		yes	yes		PK	
booking_id	int	11		yes			FK	BOOKING
status	varchar	9		yes				
payment_date	datetime		yyyy-mm-dd hh:mm:ss					

Table 4.10: Data Dictionary of Table feedback

FEEDBACK								
Attribute Name	Data Type	Length	Format	Not Null?	Unique?	Default	PK or FK	FK Referenced Table
feedback_id	int	11		yes	yes		PK	
booking_id	int	11		yes			FK	BOOKING
rating	float			yes				
submitted_at	datetime		yyyy-mm-dd hh:mm:ss	yes		current date		
comment	varchar	300						

4.3.3 Physical Design

The physical design translates the logical model into actual database structures using MySQL. This includes the creation of tables, constraints, views, triggers, stored procedures, functions and scheduled event. The DDL scripts define how data is stored and accessed, while additional database objects support performance optimization and business logic automation.

4.3.3.1 Data Definition Language

This subsection presents the full set of Data Definition Language (DDL) scripts used to implement the HygieiaHub database. Each script corresponds to a table or object defined in the data dictionary and includes primary key declarations, foreign key constraints, and validation checks. In addition, a list of database objects such as views, triggers, stored procedures, functions, and scheduled event is provided to illustrate how the system supports reporting, logging, and efficient data retrieval.

```
CREATE TABLE CUSTOMER (
    customer_id INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(255) NOT NULL,
    phone_number VARCHAR(11) NOT NULL,
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (customer_id)
);
```

Figure 4.3: DDL for Table customer

```
CREATE TABLE HOUSE_TYPE (
    house_id INT(11) NOT NULL AUTO_INCREMENT,
```

```

        name VARCHAR(100) NOT NULL,
        base_hourly_rate DECIMAL(5,2) NOT NULL,
        min_hours DECIMAL(3,2) NOT NULL,
        PRIMARY KEY (house_id)
);

```

Figure 4.4: DDL for Table house_type

```

CREATE TABLE CUSTOMER_ADDRESSES (
    address_id INT(11) NOT NULL AUTO_INCREMENT,
    customer_id INT(11) NOT NULL,
    address_label VARCHAR(50) NOT NULL,
    house_id INT(11) NOT NULL,
    address VARCHAR(300) NOT NULL,
    city VARCHAR(100) NOT NULL,
    state VARCHAR(100) NOT NULL,
    is_default VARCHAR(50) DEFAULT '1',
    PRIMARY KEY (address_id),
    FOREIGN KEY (customer_id) REFERENCES
    CUSTOMER(customer_id),
    FOREIGN KEY (house_id) REFERENCES
    HOUSE_TYPE(house_id)
);

```

Figure 4.5: DDL for Table customer_addresses

```

CREATE TABLE STAFF (
    staff_id INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    password VARCHAR(255),
    phone_number VARCHAR(11) NOT NULL,

```

```

branch VARCHAR(100) NOT NULL,
role VARCHAR(7) NOT NULL,
created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
status VARCHAR(9) NOT NULL DEFAULT 'Active',
image_path VARCHAR(255),
PRIMARY KEY (staff_id)
);

```

Figure 4.6: DDL for Table staff

```

CREATE TABLE ADDITIONAL_SERVICE (
    service_id INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    description VARCHAR(300),
    price_RM DECIMAL(5,2) NOT NULL,
    duration_hour DECIMAL(3,2) NOT NULL,
    PRIMARY KEY (service_id)
);

```

Figure 4.7: DDL for Table additional_service

```

CREATE TABLE BOOKING (
    booking_id INT(11) NOT NULL AUTO_INCREMENT,
    customer_id INT(11) NOT NULL,
    address_id INT(11),
    house_id INT(11) NOT NULL,
    address VARCHAR(1000) NOT NULL,
    hours_booked DECIMAL(4,2) NOT NULL,
    no_of_cleaners INT(2) NOT NULL,
    custom_request VARCHAR(300),
    total_RM DECIMAL(6,2) NOT NULL,
    estimated_duration_hour DECIMAL(4,2) NOT NULL,
)

```

```

    scheduled_date DATE NOT NULL,
    scheduled_time TIME NOT NULL,
    status VARCHAR(9) NOT NULL,
    note VARCHAR(300),
    PRIMARY KEY (booking_id),
    FOREIGN KEY (customer_id) REFERENCES
    CUSTOMER(customer_id),
    FOREIGN KEY (address_id) REFERENCES
    CUSTOMER_ADDRESSES(address_id),
    FOREIGN KEY (house_id) REFERENCES
    HOUSE_TYPE(house_id)
);

```

Figure 4.8: DDL for Table booking

```

CREATE TABLE BOOKING_SERVICE (
    booking_id INT(11) NOT NULL,
    service_id INT(11) NOT NULL,
    PRIMARY KEY (booking_id, service_id),
    FOREIGN KEY (booking_id) REFERENCES
    BOOKING(booking_id),
    FOREIGN KEY (service_id) REFERENCES
    ADDITIONAL_SERVICE(service_id)
);

```

Figure 4.9: DDL for Table booking_service

```

CREATE TABLE BOOKING_CLEANER (
    booking_id INT(11) NOT NULL,
    staff_id INT(11) NOT NULL,
    PRIMARY KEY (booking_id, staff_id),
    FOREIGN KEY (booking_id) REFERENCES
    BOOKING(booking_id),

```

```

    FOREIGN KEY (staff_id) REFERENCES STAFF(staff_id)
);

```

Figure 4.10: DDL for Table booking_cleaner

```

CREATE TABLE PAYMENT (
    payment_id INT(11) NOT NULL AUTO_INCREMENT,
    booking_id INT(11) NOT NULL,
    status VARCHAR(9) NOT NULL,
    payment_date DATETIME,
    PRIMARY KEY (payment_id),
    FOREIGN KEY (booking_id) REFERENCES
BOOKING(booking_id)
);

```

Figure 4.11: DDL for Table payment

```

CREATE TABLE FEEDBACK (
    feedback_id INT(11) NOT NULL AUTO_INCREMENT,
    booking_id INT(11) NOT NULL,
    rating FLOAT NOT NULL,
    submitted_at DATETIME NOT NULL DEFAULT
CURRENT_TIMESTAMP,
    comment VARCHAR(300),
    PRIMARY KEY (feedback_id),
    FOREIGN KEY (booking_id) REFERENCES
BOOKING(booking_id)
);

```

Figure 4.12: DDL for Table feedback

Table 4.11: List of Database Objects using MySQL Command

No.	Goal	Tables Involved	Oracle Command	Type	Description
1	Allow staff to only view bookings made at their branch.	booking, customer	<pre>CREATE VIEW branch_booking AS SELECT b.* FROM booking b JOIN customer c ON b.customer_id = c.customer_id WHERE c.city = get_current_branch();</pre>	View	<ul style="list-style-type: none"> Provides branch-level filtered view of bookings based on the currently logged-in staff's branch, leveraging the <code>get_current_branch</code> function.
2	Log changes in booking status or note.	booking, booking_log	<pre>CREATE TRIGGER BookingStatus AFTER UPDATE ON booking FOR EACH ROW BEGIN IF OLD.status != NEW.status OR OLD.note != NEW.note THEN</pre>	Trigger	<ul style="list-style-type: none"> Ensures that updates to booking status or note are tracked in the <code>booking_log</code>. If the note contains "attention," the change is marked as system-generated.

			<pre> IF NEW.note LIKE '%attention%' THEN INSERT INTO booking_log (booking_id, made_at, made_by, old_status, new_status, new_note) VALUES (NEW.booking_id, NOW(), 'system', OLD.status, NEW.status, NEW.note); ELSE INSERT INTO booking_log (booking_id, made_at, made_by, old_status, new_status, new_note) VALUES (NEW.booking_id, NOW(), COALESCE(@current_user, 'unknown'), OLD.status, NEW.status, NEW.note); END IF; END IF; END; </pre>		
3	Automatically assign cleaners to a booking based on branch,	staff, booking, booking_cleaner	<pre> CREATE PROCEDURE AssignCleaners (IN p_booking_id INT, IN p_city VARCHAR(50), </pre>	Stored procedure	<ul style="list-style-type: none"> After a booking is made, this procedure finds available cleaners who are active and not already booked during the requested slot, then auto-

availability, and schedule.	<pre> IN p_date DATE, IN p_time TIME, IN p_duration DECIMAL(4,2), IN p_no_of_cleaners INT, OUT p_success BOOLEAN, OUT p_message VARCHAR(255)) BEGIN DECLARE v_cleaner_id INT; DECLARE v_count INT DEFAULT 0; DECLARE done INT DEFAULT FALSE; DECLARE cur CURSOR FOR SELECT s.staff_id FROM STAFF s WHERE s.branch = p_city AND s.role = 'Cleaner' </pre>		assigns them. Returns success status and message.
-----------------------------	---	--	---

```
        AND s.status = 'Active'  
        AND s.staff_id NOT IN (  
            SELECT bc.staff_id  
            FROM BOOKING_CLEANER bc  
            JOIN BOOKING b ON bc.booking_id  
= b.booking_id  
            WHERE b.scheduled_date = p_date  
            AND b.status = 'Pending'  
            AND (  
                TIME(p_time) <  
                ADDTIME(b.scheduled_time,  
                SEC_TO_TIME(b.estimated_duration_hour*3600))  
                AND  
                ADDTIME(TIME(p_time),  
                SEC_TO_TIME((p_duration + 0.5)*3600)) >  
                b.scheduled_time  
            )  
        )
```

```
        ORDER BY RAND ()  
  
        LIMIT p_no_of_cleaners;  
  
        DECLARE CONTINUE HANDLER FOR NOT FOUND  
        SET done = TRUE;  
  
  
        DECLARE EXIT HANDLER FOR SQLEXCEPTION  
  
        BEGIN  
  
            SET p_message = 'An error occurred  
while assigning cleaners.';  
  
            SET p_success = FALSE;  
  
            ROLLBACK;  
  
        END;  
  
  
        START TRANSACTION;  
  
        OPEN cur;
```

```
read_loop: LOOP

    FETCH cur INTO v_cleaner_id;

    IF done THEN

        LEAVE read_loop;

    END IF;

    INSERT INTO BOOKING_CLEANER
(booking_id, staff_id)
VALUES (p_booking_id, v_cleaner_id);

    SET v_count = v_count + 1;

END LOOP;

CLOSE cur;

IF v_count < p_no_of_cleaners THEN
```

		<pre> SET p_message = CONCAT('Only assigned ', v_count, ' out of ', p_no_of_cleaners, ' requested cleaners'); SET p_success = FALSE; ELSE SET p_message = 'Successfully assigned all cleaners'; SET p_success = TRUE; END IF; COMMIT; END; </pre>		
4	Return the branch of the currently logged-in user session.	None	<pre> CREATE FUNCTION get_current_branch() RETURNS VARCHAR(100) DETERMINISTIC BEGIN RETURN @current_user_branch; </pre>	<p>Function</p> <ul style="list-style-type: none"> Provides a branch context based on session variables, used in views and filters to restrict data visibility per branch.

			END;		
5	Mark overdue bookings and related payments with “Attention” status automatically.	booking, payment	<pre> CREATE EVENT OverdueBooking ON SCHEDULE EVERY 1 HOUR DO BEGIN -- Batch update overdue bookings and mark payments UPDATE booking b JOIN payment p ON b.booking_id = p.booking_id SET b.status = 'Attention', b.note = CONCAT(IFNULL(b.note, ''), ' [System: marked for attention at ', NOW(), ']'), p.status = 'Attention', p.payment_date = NULL WHERE b.scheduled_date < CURDATE() </pre>	Scheduled Event	<ul style="list-style-type: none"> Runs every hour to check for overdue bookings, flags them for staff review, and updates related payments in manageable batches to avoid table locking.

		AND b.status NOT IN ('Completed', 'Cancelled', 'Attention') LIMIT 100; END;		
--	--	--	--	--

4.4 Graphical User Interface Design

The graphical user interface (GUI) design of HygieiaHub focuses on delivering a clean, intuitive, and role-specific experience for both customers and staff. The design process was guided by usability principles and aligned with the system's functional scope, ensuring that users can navigate, input data, and retrieve information with minimal friction. The interface was structured prior to implementation using wireframe layouts to visualize screen flow, input forms, and output displays. This section presents the GUI design in three parts: navigation system, input design, and output design.

4.4.1 Navigation System

The navigation system defines how users move between modules and pages within HygieiaHub. For customers, the flow begins with registration or login, followed by access to service selection, booking history, and feedback submission. For staff, the navigation includes login, dashboard access, booking management, service configuration, and report generation. The interface structure diagrams illustrate the hierarchy and logical flow between screens, ensuring that users are guided through their tasks without confusion.

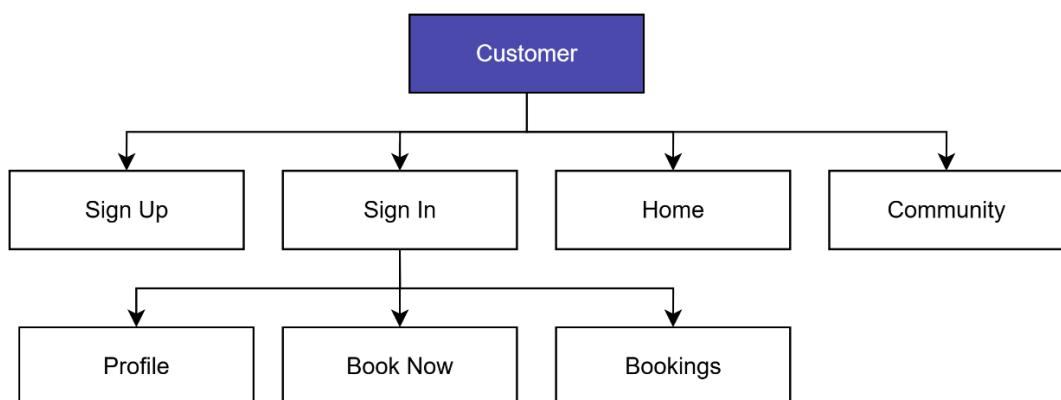


Figure 4.13: Interface Structure for Customer

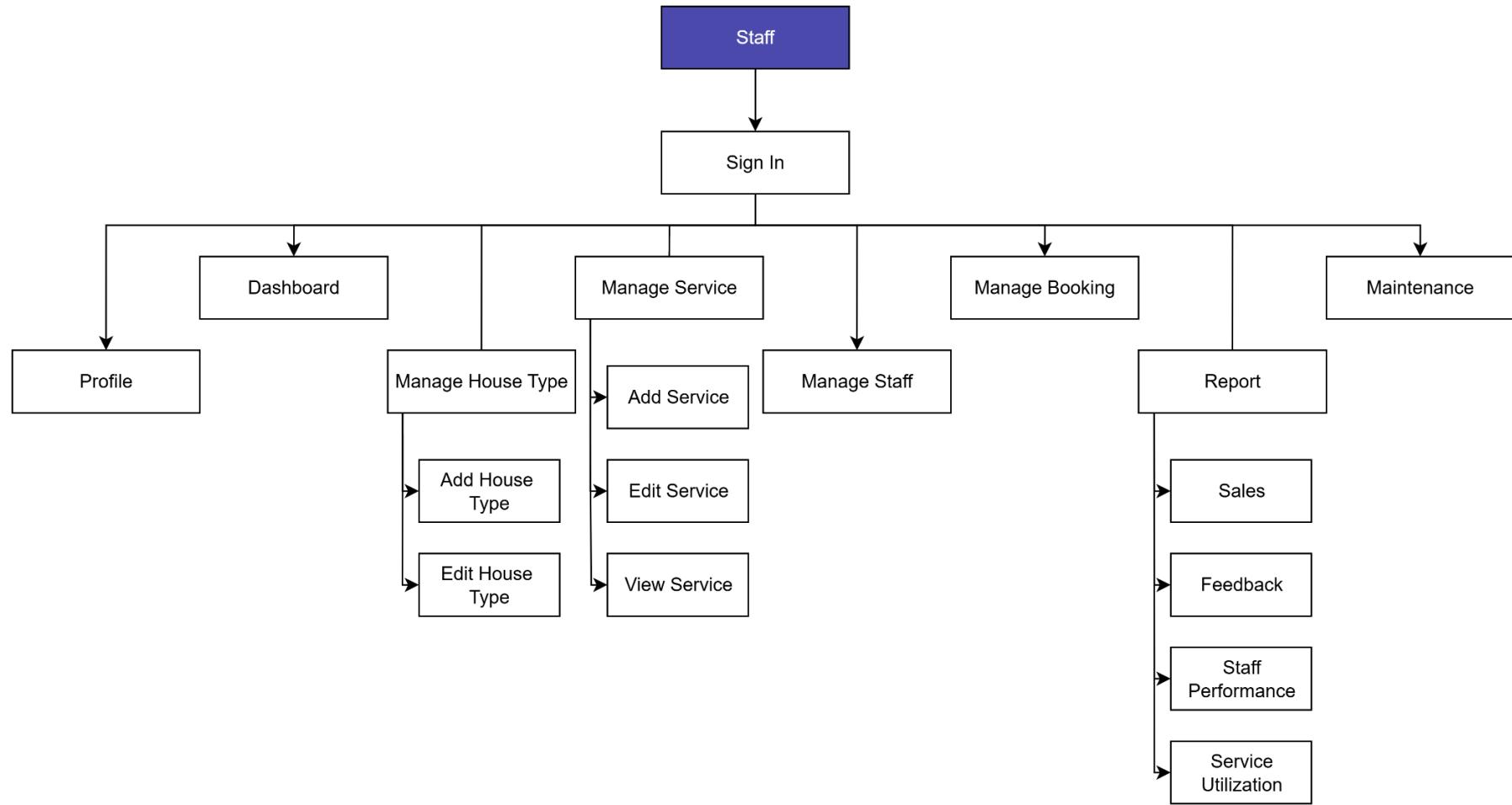
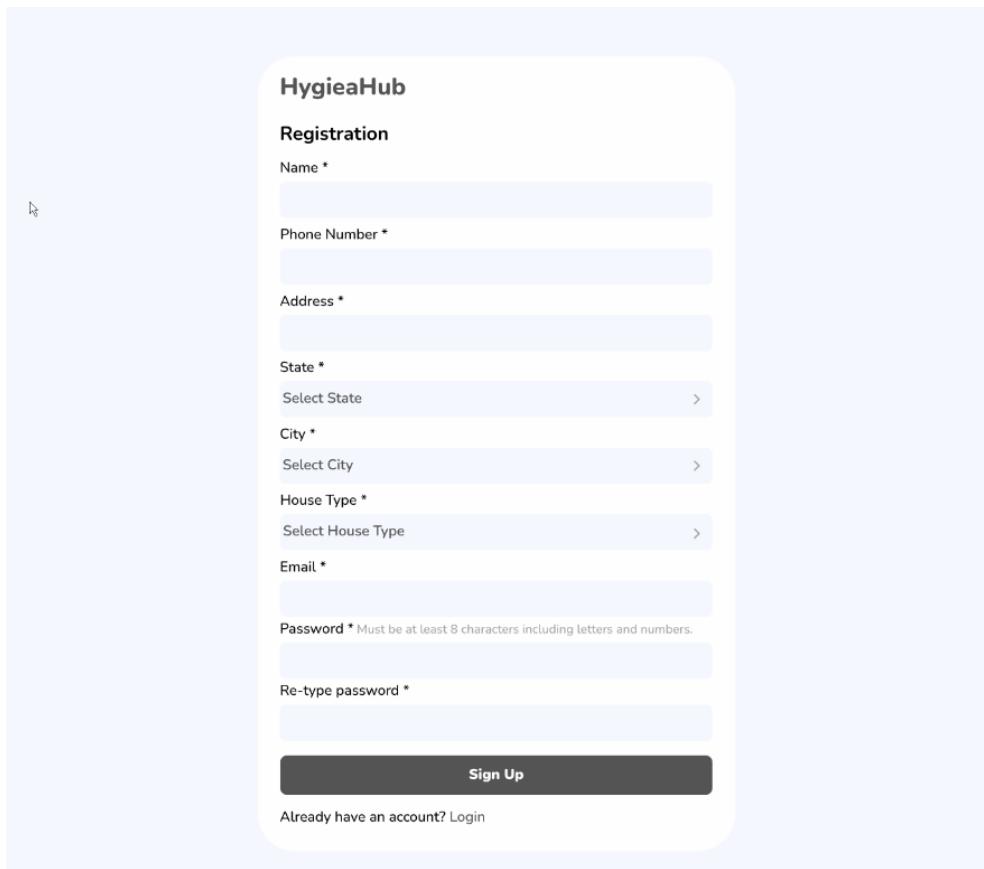


Figure 4.14: Interface Structure for Staff

4.4.2 Input Design

Input design refers to the screens where users enter data into the system. These include registration and login forms, booking forms, staff registration, and service creation pages. Each form is designed with clear labels, appropriate field types, and validation messages to reduce errors and improve user experience. For example, the booking form allows customers to select service types, booking date and time, number of cleaners, and optional requests all structured in a step-by-step layout. These designs were illustrated using Canva.



The image shows a registration form titled "Registration" for "HygieaHub". The form consists of several input fields: "Name *", "Phone Number *", "Address *", "State *", "City *", "House Type *", "Email *", "Password *" (with a note: "Must be at least 8 characters including letters and numbers"), and "Re-type password *". Below the "Sign Up" button, there is a link "Already have an account? Login".

Field	Type	Label	Notes
Name	Text	Name *	
Phone Number	Text	Phone Number *	
Address	Text	Address *	
State	Select	State *	Select State
City	Select	City *	Select City
House Type	Select	House Type *	Select House Type
Email	Text	Email *	
Password	Text	Password *	Must be at least 8 characters including letters and numbers
Re-type password	Text	Re-type password *	

Figure 4.15: Customer - Sign Up Page

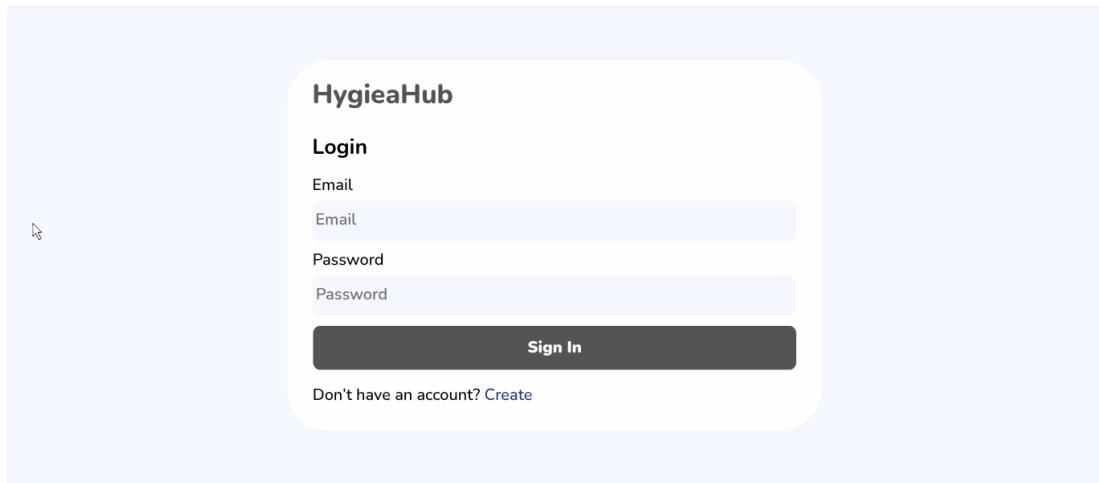


Figure 4.16: Customer & Staff - Sign In Page

Figure 4.17: Customer - Book Now Page

HygieaHub Administration

Staff

Register

Name *

Role *

Select Role

Email *

Password * Must be at least 8 characters including letters and numbers.

Phone Number *

Branch

Seremban

Register

Figure 4.18: Staff - Register Staff Page

HygieaHub Administration

Add Service

Service Name *

Service Name

Description

Description

Price *

Price

Duration *

Duration

Add

Figure 4.19: Staff - Add Service Page

The screenshot shows the 'Edit Service' page of the HygieaHub application. The left sidebar includes links for Dashboard, Manage House Type, Manage Service (with sub-options Add Service, Edit Service, and View Service), Manage Staff Account, Manage Booking, Report, and Maintenance. The main content area is titled 'Edit Service' and contains fields for Service Name (Service 1), Description (Description), Price (RM 100.00), and Duration (2 hours). There are 'Update' and 'Delete' buttons at the bottom.

HygieaHub Administration

Dashboard

Manage House Type>

Manage Service >

- Add Service
- **Edit Service**
- View Service

Manage Staff Account

Manage Booking

Report >

Maintenance

Edit Service

Service Name *

Service 1

Description

Description

Price *

RM 100.00

Duration *

2 hours

Update **Delete**

Figure 4.20: Staff - Edit Service Page

The screenshot shows the HygieaHub staff interface. The top navigation bar includes 'HygieaHub' and 'Administration'. A user icon is in the top right. On the left, a sidebar menu lists: Dashboard, Manage House Type, Manage Service, Manage Staff Account, **Manage Booking** (which is selected and highlighted in dark grey), Report, and Maintenance. The main content area is titled 'Manage Booking'. It displays booking details: Customer Name (Abu), Customer's Phone Number (01x-xxxxxx), Services (Move in/Move out Cleaning, Window Cleaning), and Custom Request. Below these are fields for Scheduled Date (5/6/2025) and Scheduled Time (2:00 PM). Under 'Cleaners Assigned', there are two entries: dvhshhb and hdbhsbh. The 'Status' field shows 'Pending' with a dropdown arrow. The 'Total Amount' is listed as RM 150.00. The 'Payment Status' field also shows 'Pending' with a dropdown arrow. At the bottom is a yellow 'Update' button.

Figure 4.21: Staff - Manage Booking Page for editing

4.4.3 Output Design

Output design focuses on how the system presents processed information to users. For customers, this includes booking summaries, service details, and feedback pages. For staff, output screens include booking management panels, service listings, and performance reports. These interfaces are structured using tables, status indicators, and summary views to ensure clarity and support informed decision-making. These designs were illustrated using Canva.

The screenshot shows the 'Your Bookings' section of the HygieaHub app. It displays two booking entries. Each entry includes fields for Date, Time, Payment Status, Cleaners, Status, and a 'Cancel Booking' button. Below the entries is a 'More Details' button. At the bottom, there's a feedback section with 'Rating', 'Comment', and a 'Submit' button.

Date	Time	Cleaners	Status
27/5/2025	10:00 AM	Cleaner 1 Cleaner 2	In Process
2/3/2025	10:00 AM	Cleaner 1 Cleaner 2	Completed

Feedback

Rating: [Placeholder] Comment: [Placeholder] Submit

More Details

Figure 4.22: Customer - Bookings Page

The screenshot shows the 'Feedback' section of the HygieaHub app. It features a large star rating of 4.5/5 based on 3216 customers. Below the rating, two customer reviews are displayed in rounded boxes. Each review includes a profile icon, the customer's name, their rating, the date, location, and a short comment.

Feedback

Customer 1
bchbccbxjcjjcjxjkcjxncjx.. dhcbdhcbhdcbbchbch.. dfjfjj
hfhdhfdfbh.

Rating: 4/5 Date: 27/7/2020 From: Cheras, Selangor

Customer 2
bchbccbxjcjjcjxjkcjxncjx.. dhcbdhcbhdcbbchbch.. dfjfjj
hfhdhfdfbh.

Rating: 3/5 Date: 15/4/2023 From: Karak, Pahang

Figure 4.23: Customer - Community Page

HygieaHub Administration

Dashboard

Manage House Type >

Manage Service **Add Service**

Edit Service

View Service

Manage Staff Account

Manage Booking

Report >

Maintenance

View Service

Name	Description	Price	Duration
Window Cleaning	Windows are cleaned thoroughly	RM 50.00	1.00 hour

Figure 4.24: Staff - View Service Page

HygieaHub Administration

Dashboard

Manage House Type >

Manage Service >

Manage Staff Account

Manage Booking

Report >

Maintenance

Staff

Role : Select Role Status : Select Status

Register New Staff

#	Name	Email	Phone No.	Branch	Role	Date Created	Status	Action
1	Ali	Ali@email	012-3456789	Seremban	Cleaner	27-3-2025	Active	Edit

Figure 4.25: Staff - Manage Staff Page

Booking

#	Customer's Name	Address	Scheduled Date	Scheduled Time	Status	Action
1	Abu	Taman, Seremban, NS	23-7-2025	11:00	Pending	Manage

Figure 4.26: Staff - Manage Booking Page

Sales

Period	Total Bookings	Total Sales (RM)	Average Sale (RM)
2025-6	5	1834.59	366.92

2000
1500
1000
500
0

2025-3 2025-4 2025-5 2025-6

Print

Figure 4.27: Staff - Sales Report Page

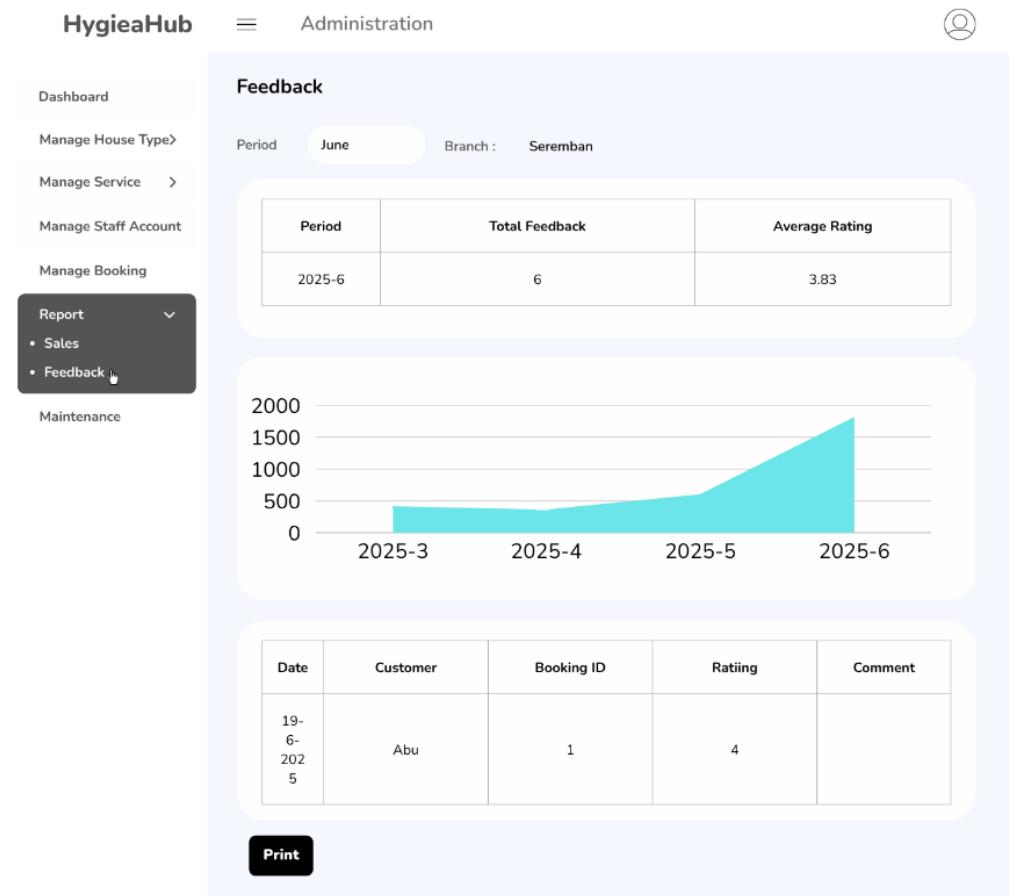


Figure 4.28: Staff - Feedback Report Page

4.5 Conclusion

In conclusion, this chapter has presented a comprehensive design framework for HygieaHub, covering both high-level architecture and detailed system components. The system architecture outlines the layered structure of the application, while the data model design provides conceptual, logical, and physical representations of the database. The graphical user interface design illustrates how users interact with the system through structured navigation, input forms, and output displays. Together, these design elements form a cohesive blueprint that supports the system's goals of usability, reliability, and operational efficiency, laying the foundation for successful implementation in the next phase.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter describes the implementation phase of the HygieiaHub system, developed to streamline booking and cleaning service management for a single provider using a web-based interface. It details the setup of the development environment, database manipulation techniques using DML (Data Manipulation Language), execution of key SQL queries, and the design of system interfaces aligned with user roles and functionalities.

5.2 Environment Setup

This section describes the tools and configurations used to develop and run the HygieiaHub booking and cleaning platform. The setup includes both software components critical to system development and hardware specifications needed for testing and deployment. Additionally, it outlines how the system connects to the database securely via PHP.

5.2.1 Software (S/W) Environment

The HygieiaHub system was built using lightweight, open-source technologies suited for single-host deployment and academic development. The tools selected ensured compatibility, flexibility, and ease of testing across system modules.

Table 5.1: Software Environment Summary

Software Component	Tool/Software Name	Version	Purpose
DBMS	MySQL	8.0	Database storage and management
Server Platform	XAMPP	8.2.0	Local web server with Apache and MySQL integration
Code Editor	Visual Studio Code	1.84.2	Development environment for backend and frontend
Database Interface	phpMyAdmin	5.2.1	GUI-based MySQL database interaction
Diagram Tool	draw.io (Web App)	N/A	Design DFDs, ERDs, and flowcharts
Web Browser	Google Chrome	117.0	Interface testing and system validation

These tools provided sufficient support for coding, testing, and designing interfaces, while maintaining compatibility with local server configurations.

5.2.2 Hardware (H/W) Requirements

The system was developed and tested on a personal workstation with the following minimum specifications:

Table 5.2: Hardware Requirement Summary

Component	Specification
Processor	Intel Core i5, 2.4 GHz or higher
RAM	8 GB
Storage	256 GB SSD (50 GB allocated for project files)

Display	1920x1080 resolution
Network	Localhost via XAMPP

No specialized equipment was required beyond standard computing resources. Backups were performed manually by exporting SQL data and archiving PHP files.

5.2.3 Database Connection

The system connects to the MySQL database using PHP's mysqli library through a local XAMPP configuration. The database resides on localhost and uses default authentication during development.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "hygieiahub";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} /*else {
    echo "Database connection successful!";
} */
?>
```

Figure 5.1: Database Connection Script

1. Host: localhost (configured via XAMPP)
2. Username: root (default)
3. Password: empty for local testing (can be secured in production)
4. Database Name: hygieiahub

The connection file is included in all server-side modules to enable real-time interaction with database records during login, bookings, payment updates, and feedback submission.

5.3 DML Implementation

This section explains how Data Manipulation Language (DML) commands were used to populate and manage records within the HygieiaHub database. The system relies on structured tables such as customer, customer_addresses, staff, booking, payment, additional_service, and feedback, all connected via foreign keys as outlined in Chapter 4's ERD. DML operations such as INSERT, UPDATE, and DELETE were implemented during development to support data creation, modification, and removal based on real-time user actions.

5.3.1 DML Insert Statement

Query:

Insert a new booking into the BOOKING table for all columns. The booking details are as follows:

booking_id	2
customer_id	3
address_id	3
house_id	1
address	No. 5, Jalan Mawar, Kuala Lumpur
hours_booked	4.00
no_of_cleaners	2
custom_request	Use eco-friendly cleaning products
total_RM	240.00
estimated_duration_hour	4.00

scheduled_date	2025-09-01
scheduled_time	10:00:00
status	Pending
note	Customer prefers morning session

SQL statement:

```
INSERT INTO BOOKING
VALUES (
    2, 3, 3, 1, 'No. 5, Jalan Mawar, Kuala Lumpur',
    4.00, 2, 'Use eco-friendly cleaning products',
    240.00, 4.00, '2025-09-01', '10:00:00',
    'Pending', 'Customer prefers morning session'
);
```

5.3.2 DML Update Statement**Query :**

Change the status of booking with ID 1 from Pending to Completed.

SQL statement:

```
UPDATE BOOKING
SET status = 'Completed'
WHERE booking_id = 1;
```

5.3.3 DML Delete Statement**Query:**

Delete all rows in the booking_service table.

SQL statement:

```
DELETE BOOKING_SERVICE;
```

5.4 Query Implementation

This section details a series of structured SQL queries designed to extract meaningful insights from the HygieiaHub database. Each query plays a crucial role in supporting operational efficiency, from tracking booking activity to measuring agent performance. The goal is to ensure data-driven decision-making across key functions such as service scheduling, customer management, and feedback analysis. By translating system requirements into precise query logic, this implementation bridges the gap between raw data and actionable reporting, enhancing overall platform responsiveness and strategic visibility.

5.4.1 Query to Retrieve Filtered Bookings by Customer Preferences

Query: Retrieve all bookings made by a logged-in customer, along with assigned cleaners, selected services, house type, payment status, and feedback. Filter further by optional inputs: scheduled date, booking status, and payment status.

Relational Algebra (RA):

$\pi_{\text{booking_id}, \text{scheduled_date}, \text{cleaners}, \text{services}, \text{house_type}, \text{payment_status}, \text{rating}, \text{comment}} (\sigma_{\text{customer_id} = 3 \wedge [\text{optional filters}]} (\text{BOOKING} \bowtie \text{STAFF} \bowtie \text{ADDITIONAL_SERVICE} \bowtie \text{HOUSE_TYPE} \bowtie \text{PAYMENT} \bowtie \text{FEEDBACK}))$

```

SELECT b.booking_id, b.scheduled_date,
       GROUP_CONCAT(DISTINCT s.name SEPARATOR ', ') AS cleaners,
       GROUP_CONCAT(DISTINCT asv.name SEPARATOR ', ') AS services,
       h.name AS house_type,
       p.status AS payment_status,
       f.rating AS feedback_rating,
       f.comment AS feedback_comment
FROM booking b
LEFT JOIN booking_cleaner bc ON b.booking_id = bc.booking_id
LEFT JOIN staff s ON bc.staff_id = s.staff_id
    
```

```

LEFT JOIN booking_service bs ON b.booking_id = bs.booking_id
LEFT JOIN additional_service asv ON bs.service_id = asv.service_id
LEFT JOIN house_type h ON b.house_id = h.house_id
LEFT JOIN payment p ON b.booking_id = p.booking_id
LEFT JOIN feedback f ON b.booking_id = f.booking_id
WHERE b.customer_id = 3
    AND (b.scheduled_date = '2025-06-30')
    AND (b.status = 'Cancelled')
    AND (p.status = 'Cancelled')
GROUP BY b.booking_id
ORDER BY b.scheduled_date DESC, b.scheduled_time DESC;

```

Figure 5.2: DML Statement for Select booking details

booking_id	scheduled_date	cleaners	services	house_type	payment_status	feedback_rating	feedback_comment
5	2025-06-30	Cleaner Baek, Cleaner Faiz	Carpet Shampooing, Pet Area Cleaning	Condominium	Cancelled	NULL	NULL

Figure 5.3: Result for DML in Figure 5.2

5.4.2 Query to Generate Sales Report Over Selected Periodicity

Query: Calculate total completed sales per period, dynamically grouped by daily, monthly, or yearly intervals. For this case, year sales for year 2025.

Relational Algebra (RA):

```

AGGREGATE SUM(total_RM) GROUP BY YEAR(payment_date) ( $\sigma$  status = 'Completed'  $\wedge$  YEAR(payment_date) = 2025 (PAYMENT  $\bowtie$  PAYMENT.booking_id = BOOKING.booking_id BOOKING))

```

```

SELECT
    YEAR(p.payment_date) AS year,
    SUM(b.total_RM) AS total_sales
FROM payment p
JOIN booking b ON p.booking_id = b.booking_id
WHERE p.status = 'Completed'
    AND YEAR(p.payment_date) = 2025

```

```
GROUP BY year;
```

Figure 5.4: DML Statement for Select yearly sales for 2025

year	total_sales
2025	5056.46

Figure 5.5: Result for DML in Figure 5.4

5.4.3 Query to Retrieve All Bookings with Optional Filters for Management

Query: Retrieve all bookings for administrative oversight, joined with customer details, assigned cleaners, services, payment, and feedback status. Filter further by optional inputs: scheduled date, cleaner's name, booking status, and payment status.

Relational Algebra (RA):

π booking_id, status, scheduled_date, scheduled_time, address, cleaners, estimated_duration_hour, total_RM, payment_status (σ [optional filters] (BOOKING \bowtie CUSTOMER \bowtie HOUSE_TYPE \bowtie BOOKING_CLEANER \bowtie STAFF \bowtie BOOKING_SERVICE \bowtie ADDITIONAL_SERVICE \bowtie PAYMENT \bowtie FEEDBACK))

```

SELECT b.*,
       c.name AS customer_name,
       c.phone_number,
       h.name
AS house,
       GROUP_CONCAT(DISTINCT s.name SEPARATOR ', ') AS
cleaners,
       GROUP_CONCAT(DISTINCT asv.name SEPARATOR ', ') AS
services,
       p.status AS payment_status,
       f.rating AS rating,
       f.comment AS comment,
       bl.made_at, bl.made_by
FROM booking b
JOIN CUSTOMER c ON b.customer_id = c.customer_id

```

```

JOIN HOUSE_TYPE h ON b.house_id = h.house_id
LEFT JOIN BOOKING_CLEANER bc ON b.booking_id = bc.booking_id
LEFT JOIN STAFF s ON bc.staff_id = s.staff_id
LEFT JOIN BOOKING_SERVICE bs ON b.booking_id = bs.booking_id
LEFT JOIN ADDITIONAL_SERVICE asv ON bs.service_id =
asv.service_id
LEFT JOIN PAYMENT p ON p.booking_id = b.booking_id
LEFT JOIN FEEDBACK f ON f.booking_id = b.booking_id
LEFT JOIN (SELECT bl1.*
FROM booking_log bl1
INNER JOIN (
    SELECT booking_id, MAX(made_at) AS latest_log
    FROM booking_log
    GROUP BY booking_id
) bl2 ON bl1.booking_id = bl2.booking_id AND
bl1.made_at = bl2.latest_log
) bl ON b.booking_id = bl.booking_id
WHERE 1=1
AND (b.scheduled_date = '2025-06-22') -- Optional Date
Filter
-- AND (s.name LIKE '%John%') -- Optional Cleaner Filter
-- AND (b.status = '%') -- Optional Status Filter
-- AND (p.status = 'Pending') -- Optional Payment Status
Filter
GROUP BY b.booking_id
ORDER BY CASE WHEN b.status = 'Attention' THEN 0 ELSE 1 END,
        b.scheduled_date DESC,
        b.scheduled_time DESC;

```

Figure 5.6: DML Statement for Management Booking Overview

booking_id	customer_id	address_id	house_id	address	hours_booked	no_of_cleaners	custom_request	total_RM	estimated_duration_hour	scheduled_date	scheduled_time	status	note	customer_name	phone_number	house	cleaners	services	payment_status	rating	comment	made_at	made_by
25	1	1	99	Taman Desa Dentia	3.50	1		344.50	5.50	2025-06-22	10:00:00	Completed	Peter Parker	0173647364	Terrace House	Cleaner Km	Kitchen Deep Clean	Completed	NULL	NULL	2025-06-23 12:30:43	Admin data	
26	5	4	10	Jalan Rosa	3.50	1		308.90	7.00	2025-06-22	10:00:00	Completed	Masha Zainuddin	0133242360	Terrace House	Cleaner Anah	Bathroom Interior Kitchen Deep Clean	Completed	NULL	NULL	2025-06-06 15:50:23	Alpha	

Figure 5.7: Result for DML in Figure 5.6

5.5 System Interface

This section outlines the real system interface developed for HygieiaHub, showcasing how the actual design aligns with the interface designs introduced in Chapter 4. The final implementation includes separate user interfaces tailored to the roles of customers and staff. All pages are structured to prioritize clarity, ease of navigation, and responsive interaction, supporting the core functionalities such as service selection, booking submission, feedback entry, and staff-side booking management.

5.5.1 Interfaces of User Customer

5.5.1.1 Sign Up Page

New customer has to register their account in the Sign Up page and enter their details including full name, phone number, house type, address, state, city, email, password, and re-enter their password.

The screenshot shows the 'Sign Up' page for HygieiaHub. The page has a light blue header and footer. The main content area is white with a dark blue header bar. The form fields are as follows:

- Full Name:** An input field with placeholder text "Full Name".
- Phone Number (01xxxxxxxx):** An input field with placeholder text "Phone Number (01xxxxxxxx)".
- House Type:** A dropdown menu labeled "House Type".
- Address:** An input field with placeholder text "Address".
- State:** A dropdown menu labeled "State".
- City:** A dropdown menu labeled "City".
- Email:** An input field with placeholder text "Email".
- Password:** An input field with placeholder text "Password". Below it is a note: "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character."
- Re-type Password:** An input field with placeholder text "Re-type Password".

At the bottom of the form is a large blue "Sign Up" button. Below the button, in the footer, is the text "Already have an account? [Sign In](#)".

Figure 5.8: Sign Up Page for Customer

5.5.1.2 Sign In Page

Registered customer may access the website to make a booking through Sign In page where customer has enter their registered email and password.

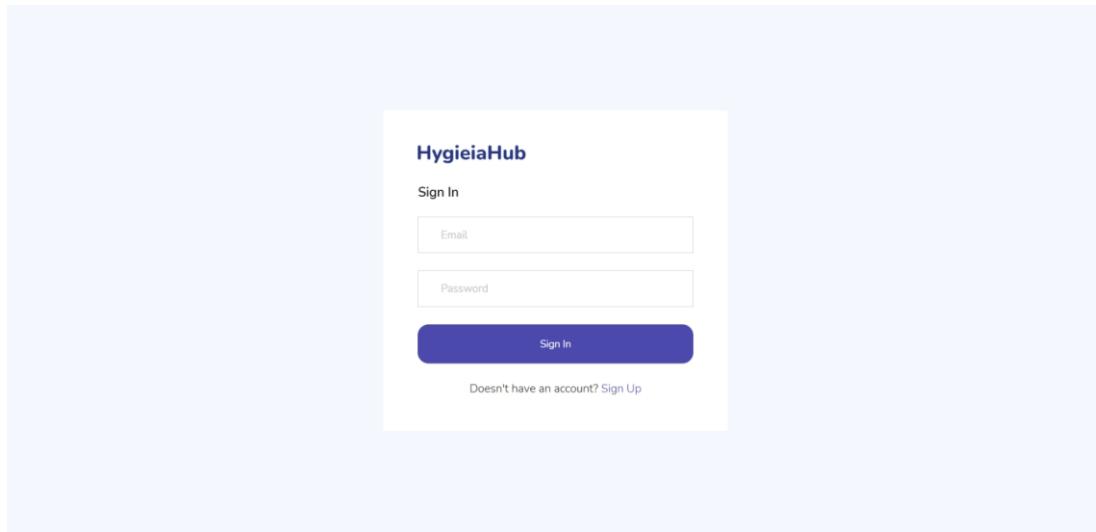


Figure 5.9: Sign In Page for Customer

5.5.1.3 Home Page

The Home page that any users may see when they open the HygieiaHub website.

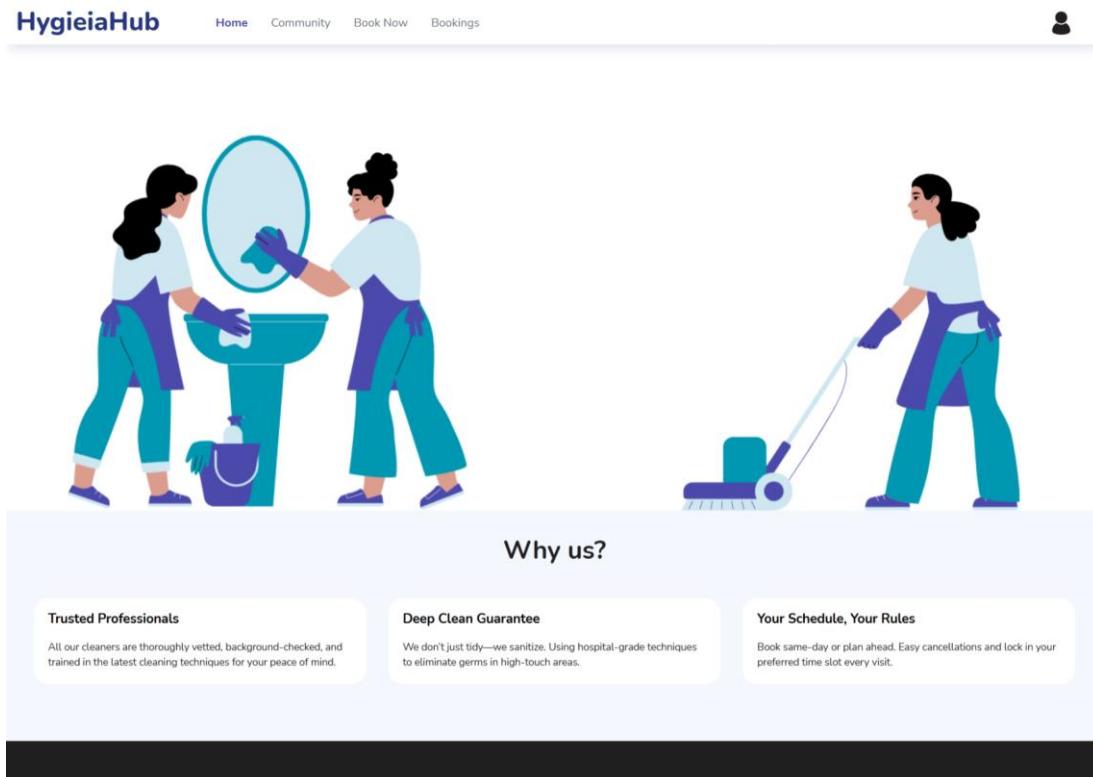


Figure 5.10: Home Page for Customer

5.5.1.4 Profile Page

Customer may update their details including adding or deleting their address in Profile page.

The image shows two overlapping web pages from the HygieiaHub platform. The left page is titled 'Your Profile' under 'Update Profile Information'. It contains fields for Name (Peter Parker), Phone Number (0173647364), Email (peter@email.com), and a Password field (Leave blank to keep current password). A note below the password field states: 'Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character.' At the bottom are 'Update' and 'Back' buttons. The right page is titled 'Your Addresses' and lists two addresses: 'Primary' (Default) at '99, Taman Desa Duranta, Seremban, Negeri Sembilan' and 'Mom's House' at '23, Taman Kelana, Batu Berendam, Melaka'. It includes buttons for 'Set Default' and 'Delete' for each address. Below these are sections for 'Add New Address' (Address Label: e.g., Home, Work), 'House Type' (dropdown menu), 'Address' (text input), 'State' (dropdown menu), 'City' (dropdown menu), and a checkbox for 'Set as default address' followed by an 'Add Address' button.

Figure 5.11: Profile Page for Customer

5.5.1.5 Community Page

Customer may view feedbacks of completed services including name, branch, rating, comment, and tags left by other customers from all branches of HygieiaHub via Community page.

The screenshot shows the 'Customer Feedback Community' section of the HygieiaHub website. At the top, it displays an overall satisfaction score of 3.9 based on 8 reviews. Below this, a list of recent customer reviews is shown, each with the reviewer's name, location, star rating, date, and a snippet of their review. The reviews are as follows:

- Peter Parker from Seremban (23 Jun 2025): nice job. [Terrace House] [2 cleaners] [Bathroom Intensive Clean, Pet Area Cleaning] [18 Jun 2025]
- Farhana from Seremban (22 Jun 2025): thank you for your service. :) [Semi-Detached] [2 cleaners] [Carpet Shampooing, Deep Cleaning] [19 Jun 2025]
- Peter Parker from Seremban (20 Jun 2025): there's still some dust spots [Terrace House] [2 cleaners] [Kitchen Deep Clean, Post-Renovation Cleaning] [17 Jun 2025]
- Misha Zainuddin from Seremban (19 Jun 2025): [Terrace House] [2 cleaners] [Window Cleaning] [16 Jun 2025]
- Misha Zainuddin from Seremban (19 Jun 2025): Thank you. My pets are now more comfortable. Really recommend this company. [Terrace House] [2 cleaners] [Pet Area Cleaning, Window Cleaning] [16 Apr 2025]
- Peter Parker from Seremban (19 Jun 2025): [Terrace House] [1 cleaner] [Carpet Shampooing] [06 May 2025]
- Peter Parker from Seremban (19 Jun 2025): Nice work. House as good as new. [Terrace House] [2 cleaners] [Bathroom Intensive Clean, Deep Cleaning] [18 Mar 2025]

Figure 5.12: Community Page for Customer

5.5.1.6 Book Now Page and Invoice Modal

Registered customer may make a booking by providing details including date, time, address, number of hours, number of cleaners, and optionally additional service and custom request in Book Now page.

Book Your Service

Date * dd/mm/yyyy Select a date first

Address * Primary: 99, Taman Desa Duranta, Seremban, Negeri Sembilan

House Type * Terrace House

Number of Hours * 3.00 hours

Additional Request We will consider your additional request

Base Price: RM 210.00
Base duration: 3.0 hour

Additional Services

- Deep Cleaning - RM 120.00 - 3.50 hour
- Move In/Out Cleaning - RM 150.00 - 4.00 hour
- Post-Renovation Cleaning - RM 180.00 - 5.00 hour
- Kitchen Deep Clean - RM 80.00 - 2.00 hour
- Bathroom Intensive Clean - RM 40.00 - 1.50 hour
- Pet Area Cleaning - RM 30.00 - 0.50 hour
- Carpet Shampooing - RM 40.00 - 1.00 hour
- Window Cleaning - RM 25.00 - 0.50 hour
- Laundry - RM 50.00 - 1.00 hour

Total for Additional Services: RM 0.00
Total estimated duration for Additional Services: 0 hour

Number Of Cleaners * Select date and time first

Calculate Total Reset

Figure 5.13: Book Now Page for Customer

When customer confirm the booking, a modal will show up showing details of booking and summaries of total price and time.

Booking Details

Booking Date: Friday, August 29, 2025
Time: 10:00
Address: 99, Taman Desa Duranta, Seremban, Negeri Sembilan

Description	Hours/Quantity	Unit Price (RM)	Amount (RM)
Base			
Base Service - Terrace House	3.0 hours	RM 70.00	RM 210.00
Number of Cleaners	2 cleaner(s)	-	-
Subtotal (Base Service x Number of Cleaners)			RM 420.00
Additional Services			
Kitchen Deep Clean - RM 80.00 - 2.00 hour	2.0 hours	RM 80.00	RM 80.00
Bathroom Intensive Clean - RM 40.00 - 1.50 hour	1.5 hours	RM 40.00	RM 40.00
Subtotal + Additional Service(s)			RM 540.00
Summary			
Service Tax (6%)	RM 32.40		
Total Amount	RM 572.40		
Estimated Total Duration	3.3 hours		

- Pay via Cash on Delivery (COD)
- Cancellation of booking can be made at least 24 hours before the scheduled date and time.
- Please contact +6019-9545506 for any inquiry.

Close **Proceed with Booking**

Figure 5.14: Invoice Modal

5.5.1.7 Bookings Page

Customer may browse their bookings history in Bookings page and cancel their pending booking or leave feedback for completed booking.

The screenshot shows the 'Your Booking History' section of the HygieiaHub website. At the top, there is a 'Your Service Patterns' box listing various cleaning services with their average frequency. Below this is a note about cancellation policies and search filters. The main area shows two pending bookings:

Booking Details		Customer Information	
Status	Payment Status	Scheduled Date	Scheduled Time
Pending	Pending	08-08-2025	10:00
		Address: 99, Taman Desa Duranta, Seremban, Negeri Sembilan	House Type: Terrace House
		Number of Hours: 3.50 hours	Custom Request:
		Services: Cleaners	2 Cleaner Becky, Cleaner Melur
		Estimated Duration: 1.75 hours	Total Amount: RM 519.40
Cancellation	Cancel Booking		

Below the first booking, there is a link 'Click to see more'. The second booking has a similar structure:

Booking Details		Customer Information	
Status	Payment Status	Scheduled Date	Scheduled Time
Pending	Pending	23-07-2025	11:00
		Click to see more	

Figure 5.15: Bookings Page for Customer

5.5.2 Interfaces of User Staff

5.5.2.1 Sign In Page

Registered staff may access the website for administration through Sign In page where staff has to enter their registered email and password.

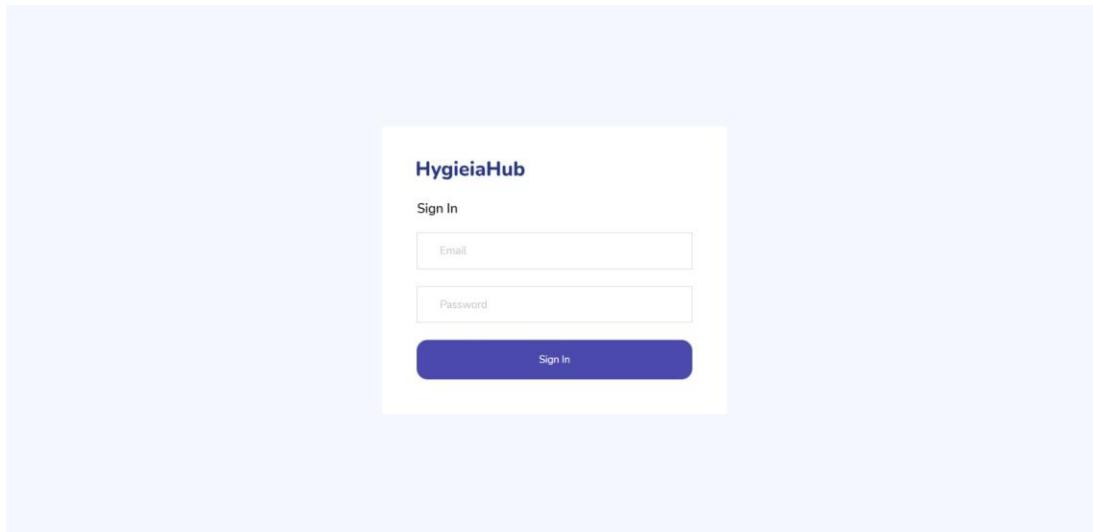


Figure 5.16: Sign In Page for Staff

5.5.2.2 Dashboard Page

When staff successfully log in, they will be greeted with Dashboard page which displays current date's booking and revenue, pending bookings, active cleaners, current date's feedback, weekly revenue trend, service popularity, and recent feedbacks based on their branch.

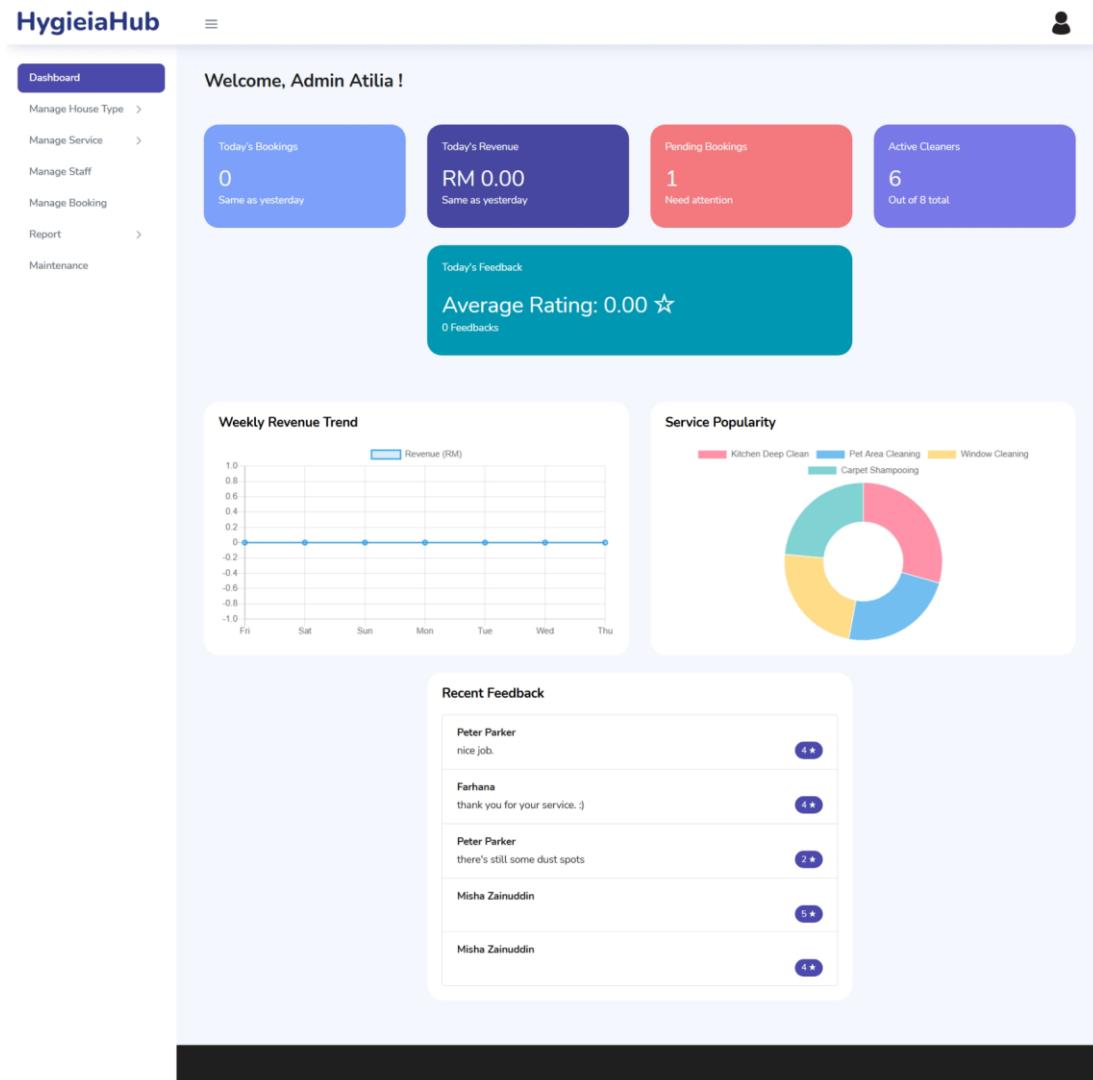


Figure 5.17: Dashboard Page for Staff

5.5.2.3 Profile Page

Staff may update their details including name, phone number, email, and password in Profile page.

Figure 5.18: Profile Page for Staff

5.5.2.4 Add House Type and Edit House Type Page

Staff may add new house type in Add House Type page with details including name, base hourly rate, and minimum hours.

Figure 5.19: Add House Type Page for Staff

Staff may update house type details which are base hourly rate and minimum hours in Edit House Type page that are not in any pending booking.

The screenshot shows the 'Edit House Type' page within the HygieiaHub application. The left sidebar includes links for Dashboard, Manage House Type (selected), Manage Service, Manage Staff, Manage Booking, Report, and Maintenance. The main content area is titled 'House Type' and contains a form for editing house types. It has fields for 'Name' (a dropdown menu), 'Base Hourly Rate' (a dropdown menu with 'RM' selected), 'Minimum Hours' (a dropdown menu with 'House Type Minimum Hours'), and an 'Update' button.

Figure 5.20: Edit House Type Page for Staff

5.5.2.5 Add, Edit, and View Service Pages

Staff may add new additional service details including name, description, price, and duration in Add Service page.

The screenshot shows the 'Add Service' page within the HygieiaHub application. The left sidebar includes links for Dashboard, Manage House Type, Manage Service (selected), Manage Staff, Manage Booking, Report, and Maintenance. The main content area is titled 'Service' and contains a form for adding services. It has fields for 'Name' (a dropdown menu with 'Service Name'), 'Description' (a text input field with 'Service Description'), 'Price' (a dropdown menu with 'RM' selected), 'Duration' (a dropdown menu with 'Service Duration'), and buttons for 'Add' and 'Reset'.

Figure 5.21: Add Service Page for Staff

Staff may delete or update additional service details excluding its name in Edit Service page only if the service is not being used in any pending booking.

The screenshot shows the 'Edit Service' page within the HygieiaHub application. On the left, a sidebar menu includes 'Manage Service' with options '+ Add Service', '+ Edit Service', and '+ View Service'. The main content area is titled 'Service' and contains a 'Edit Service' form. The form fields are: 'Name *' (dropdown menu 'Select a service'), 'Description' (text area 'Service Description'), 'Price *' (dropdown menu 'RM Service Price'), and 'Duration *' (text area 'Service Duration' with a dropdown 'hour'). At the bottom of the form are two buttons: a blue 'Update' button and a red 'Delete' button.

Figure 5.22: Edit Service Page for Staff

Staff may display active additional services' details in View Service page.

The screenshot shows the 'View Service' page within the HygieiaHub application. The sidebar menu is identical to Figure 5.22. The main content area is titled 'Service' and contains a 'Service List' table. The table has columns: Name, Description, Price, and Duration. The table shows 9 rows returned, with each row listing a service name, its description, price, and duration. The services listed are: Deep Cleaning, Move In/Out Cleaning, Post-Renovation Cleaning, Kitchen Deep Clean, Bathroom Intensive Clean, Pet Area Cleaning, Carpet Shampooing, Window Cleaning, and Laundry.

Name	Description	Price	Duration
Deep Cleaning	Intense cleaning including stains, grouts, under furniture	RM 120.00	3.50 hour
Move In/Out Cleaning	Full-service cleaning for empty homes before/after moving	RM 150.00	4.00 hour
Post-Renovation Cleaning	Heavy-duty cleaning to remove debris, dust, and paint stains	RM 180.00	5.00 hour
Kitchen Deep Clean	Degreasing kitchen walls, cabinets, stove, and appliances	RM 80.00	2.00 hour
Bathroom Intensive Clean	Scrubbing tiles, toilets, sinks, mirrors, and removing mold	RM 40.00	1.50 hour
Pet Area Cleaning	Cleaning and sanitizing areas pets stay in	RM 30.00	0.50 hour
Carpet Shampooing	Shampooing and vacuuming carpets and rugs	RM 40.00	1.00 hour
Window Cleaning	Cleaning of glass windows and sliding doors	RM 25.00	0.50 hour
Laundry	Washing, drying, and folding of clothes and linens	RM 50.00	1.00 hour

Figure 5.23: View Service Page for Staff

5.5.2.6 Manage Staff Page, Register New Staff Modal, and Edit Staff Modal

Staff may display details of staffs that were assigned to their branch in Manage Staff page.

The screenshot shows the 'HygieiaHub' application interface. On the left, there is a sidebar with navigation links: Dashboard, Manage House Type, Manage Service, **Manage Staff** (which is highlighted in blue), Manage Booking, Report, and Maintenance. The main content area is titled 'Staff'. At the top right of this area, there is a search bar labeled 'Search by:' with dropdown options 'Status', a 'Done' button, and a 'Reset' button. Below the search bar is a large blue button labeled 'Register New Staff'. Underneath this button is a table with the following columns: #, Name, Phone No., Branch, Role, and Status. The table displays 8 rows of data, each representing a staff member. The data is as follows:

#	Name	Phone No.	Branch	Role	Status
1	Cleaner Aishah	0112233445	Seremban	Cleaner	Active
2	Cleaner Melur	0137236473	Seremban	Cleaner	Active
3	Cleaner Kim	0172432434	Seremban	Cleaner	Active
4	Cleaner Michael	0172832949	Seremban	Cleaner	In-Active
5	Cleaner Azizi	0137462378	Seremban	Cleaner	In-Active
6	Cleaner Becky	0176372383	Seremban	Cleaner	Active
7	Cleaner Lee Wang	0123678228	Seremban	Cleaner	Active
8	Cleaner Faizul	0133674288	Seremban	Cleaner	Active

Figure 5.24: Manage Staff Page for Staff

Staff may register new staff for their branch which includes details like role, name, email, password, phone number, and branch in Register New Staff modal.

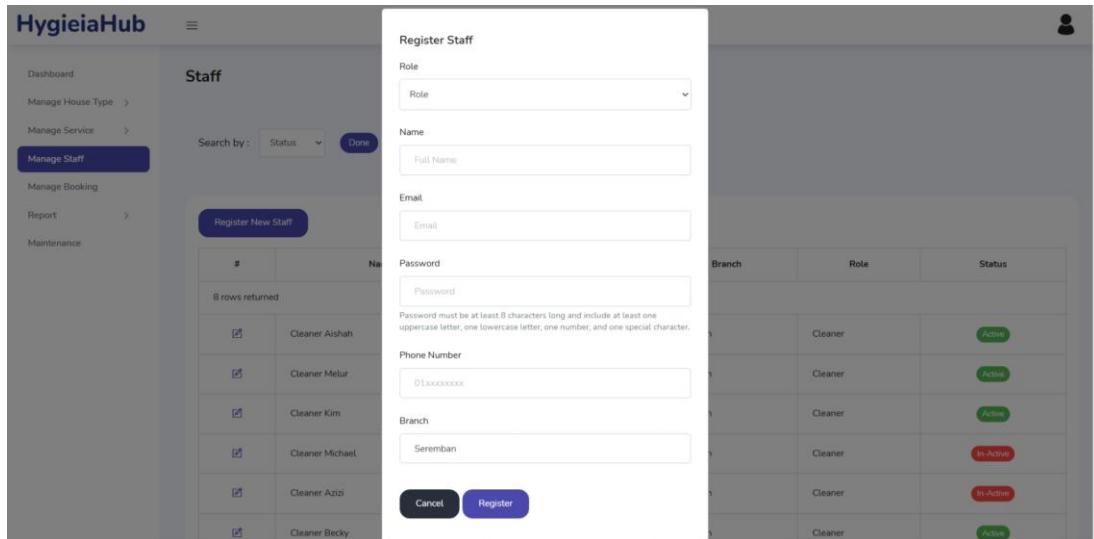


Figure 5.25: Register New Staff Modal for Staff

Staff may edit staff details in Edit Staff modal, but their status may be changed if only they are not in pending booking.

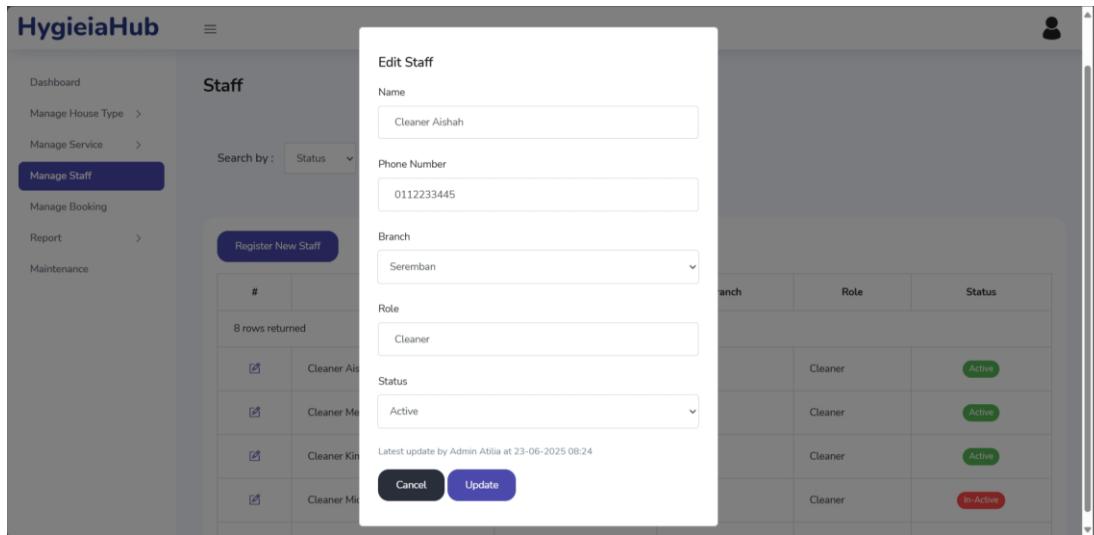


Figure 5.26: Edit Staff Modal for Staff

5.5.2.7 Manage Booking Page and Booking Details Modal

Staff may display details of all booking made for their branch in Manage Booking page.

Booking

Search by : dd/mm/yyyy Cleaner's name Status Payment status Reset

#	Status	Payment Status	Date	Time	Address	Cleaners	Estimated Duration
1	Attention	Attention	23-07-2025	11:00	99, Taman Desa Duranta	Cleaner Kim	7.85 hour
2	Attention	Attention	25-06-2025	10:00	No 71, Taman Pelangi	Cleaner Lee Wang	5.00 hour
3	Attention	Attention	24-06-2025	10:00	99, Taman Desa Duranta	Cleaner Aishah	7.00 hour
4	Pending	Pending	08-08-2025	10:00	99, Taman Desa Duranta	Cleaner Becky, Cleaner Melur	1.75 hour
5	Cancelled	Cancelled	23-06-2025	10:00	99, Taman Desa Duranta	Cleaner Melur	5.00 hour
6	Completed	Completed	22-06-2025	10:00	99, Taman Desa Duranta	Cleaner Kim	5.50 hour
7	Completed	Completed	22-06-2025	10:00	47, Taman Ros	Cleaner Aishah	7.00 hour
8	Completed	Completed	20-06-2025	11:00	47, Taman Ros	Cleaner Aishah, Cleaner Lee Wang	3.25 hour
9	Completed	Completed	19-06-2025	13:00	No 71, Taman Pelangi	Cleaner Michael	5.00 hour
10	Completed	Completed	19-06-2025	10:00	No 71, Taman Pelangi	Cleaner Aishah, Cleaner Becky	3.25 hour
11	Cancelled	Cancelled	18-06-2025	11:00	No 71, Taman Pelangi	Cleaner Kim, Cleaner Melur	4.63 hour

Figure 5.27: Manage Booking Page for Staff

Staff may display and edit details of selected booking including the status, payment status, assigned cleaner, and note in Booking Details modal.

Booking Details

Customer Information

Customer Name: Peter Parker | Phone Number: 0178647384

House Type: Terrace House | Address: 99, Taman Desa Duranta

Booking Information

Number of Hours: 6.17 | Custom Request:

Services: Kitchen Deep Clean, Post-Renovation Cleaning | Cleaners: 2 | Cleaner Aishah, Cleaner Kim

Resign Cleaners

Cleaner Aishah (Currently assigned) | Cleaner Kim (Currently assigned)

Scheduled Date: 17-06-2025 | **Scheduled Time**: 10:00 | **Status**: Completed

Estimated Duration: 6.17 | **Total**: 540.00 | **Payment Status**: Completed

Payment Information

Note: There's still some dust spots

Feedback

Rating: 2 | Comment: There's still some dust spots |

Figure 5.28: Booking Details Modal for Staff

5.5.2.8 Sales Report, Feedback Report, Staff Performance Report, and Service Utilization Report Pages

Staff may display sales report by selecting its periodicity in Sales Report page. Sales summary and trend will be displayed.

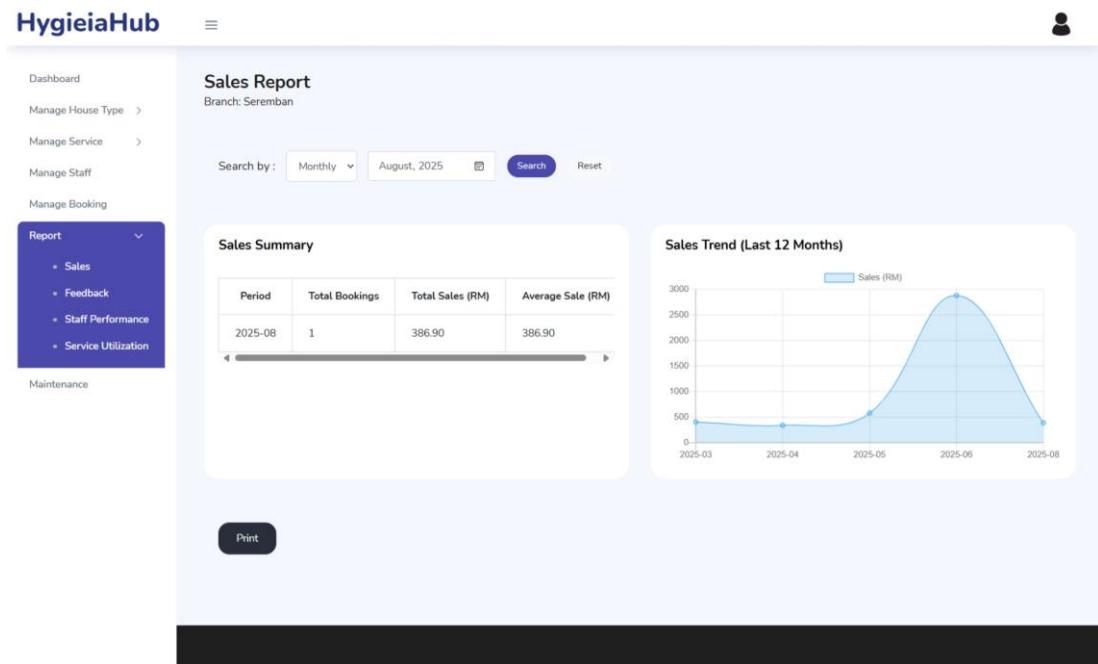


Figure 5.29: Sales Report Page for Staff

Staff may display feedback report by selecting its month in Feedback Report page. Feedback summary, trend, and its details will be displayed.

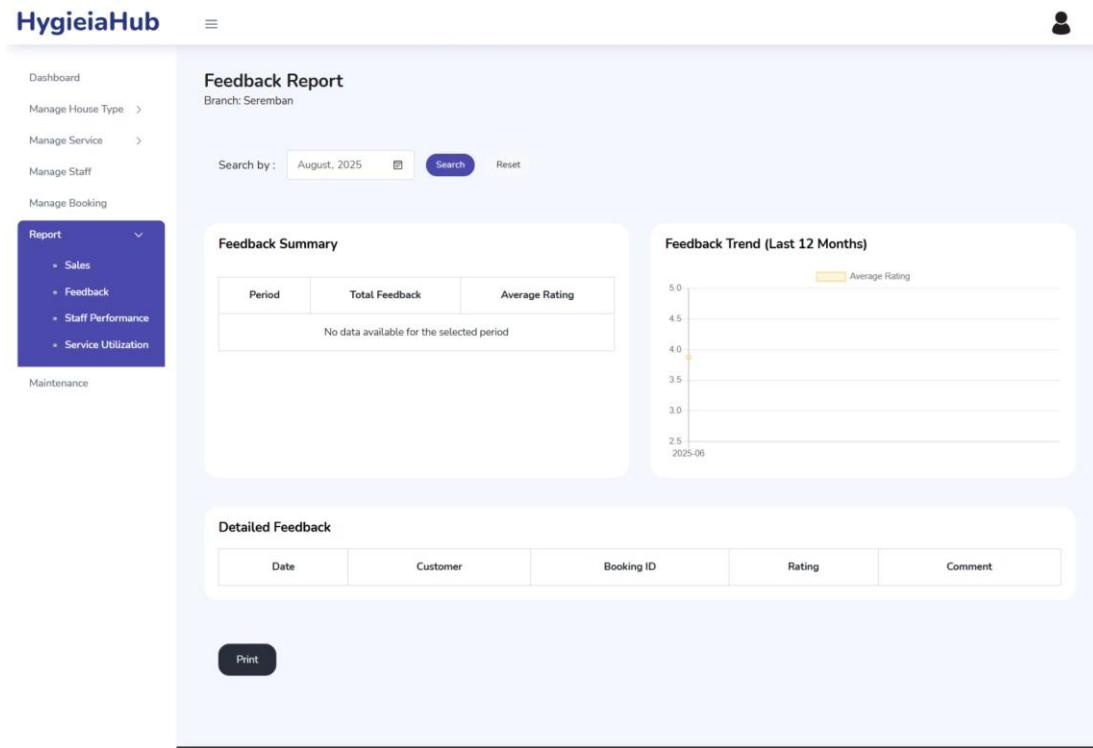


Figure 5.30: Feedback Report Page for Staff

Staff may display staff performance report by selecting its periodicity in Staff Performance Report page. Top performance, performance summary, and details will be displayed

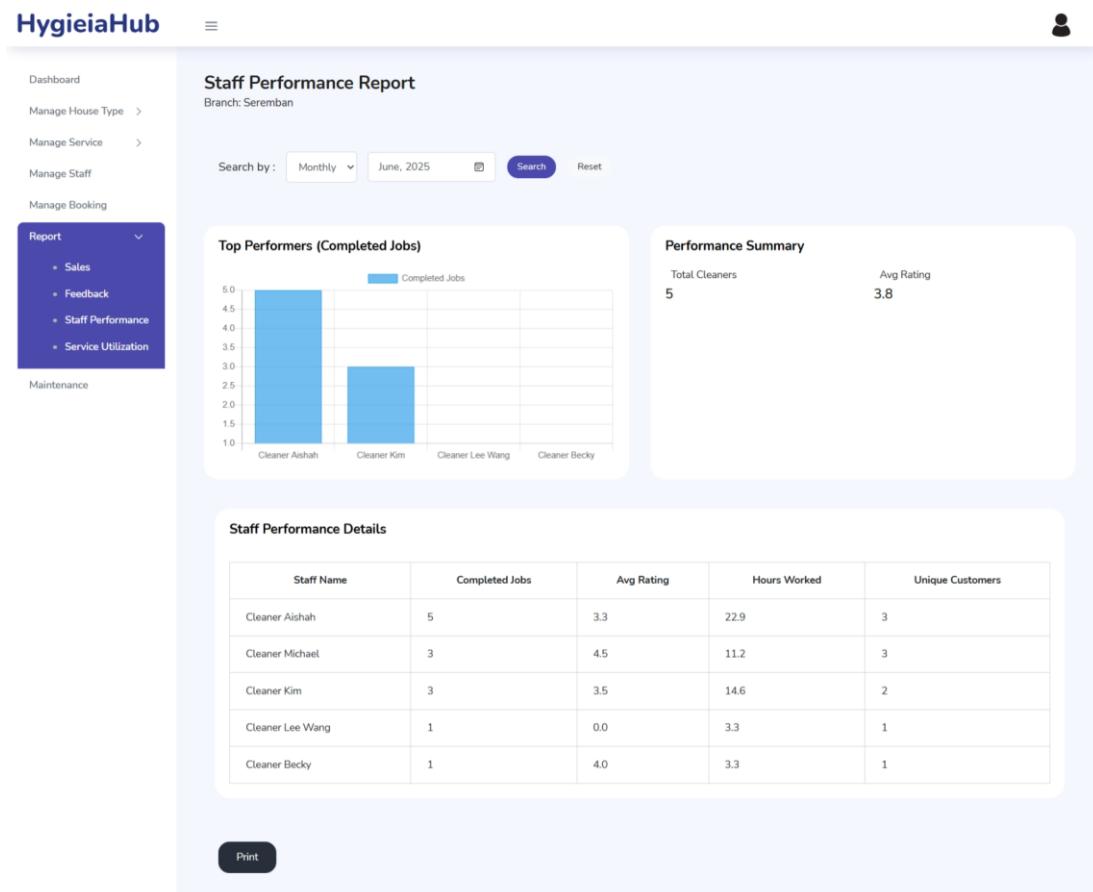


Figure 5.31: Staff Performance Report Page for Staff

Staff may display service utilization by selecting its periodicity in Service Utilization page. Top service, service summary, and details will be displayed.

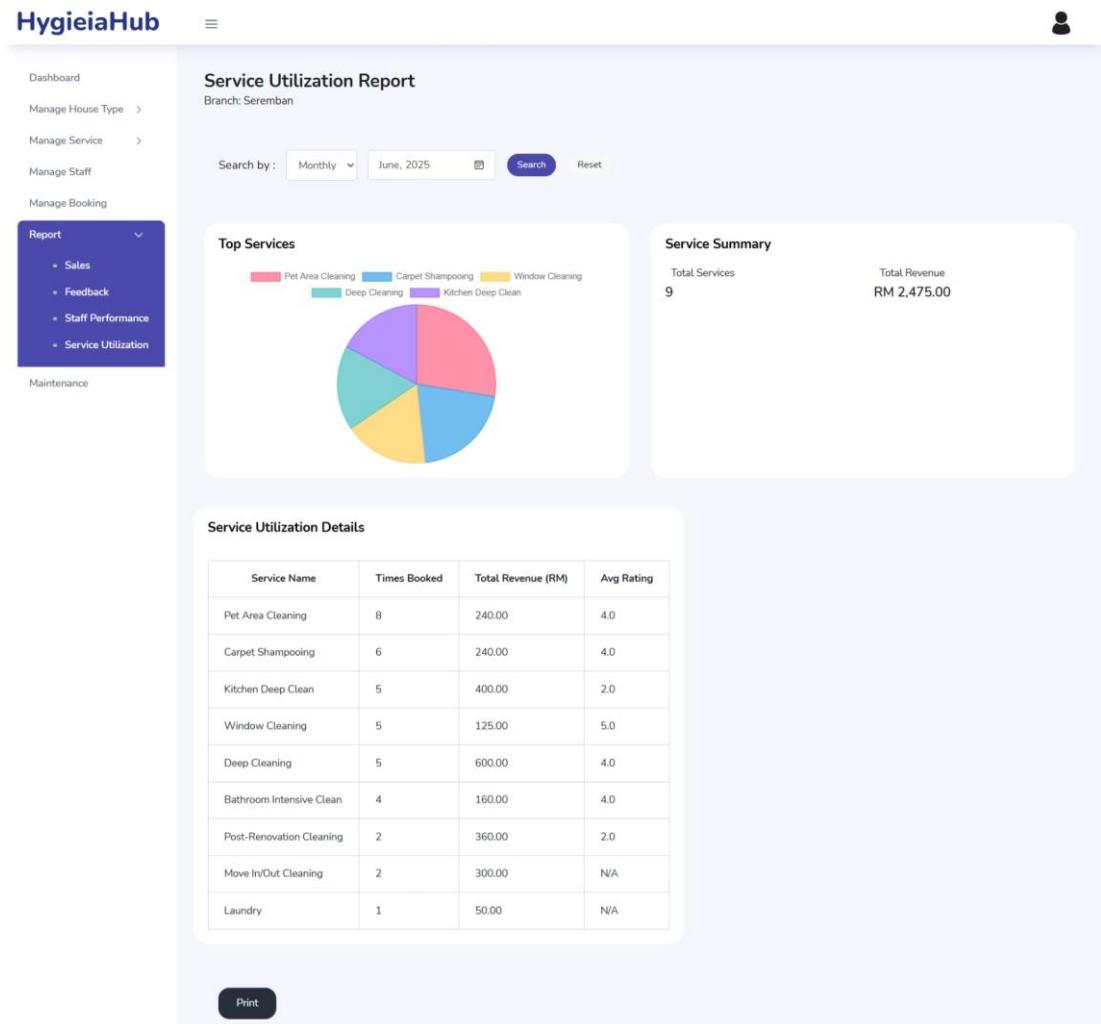


Figure 5.32: Service Utilization Report Page for Staff

5.5.2.9 Maintenance Page

Staff may backup or restore database through Maintenance page.

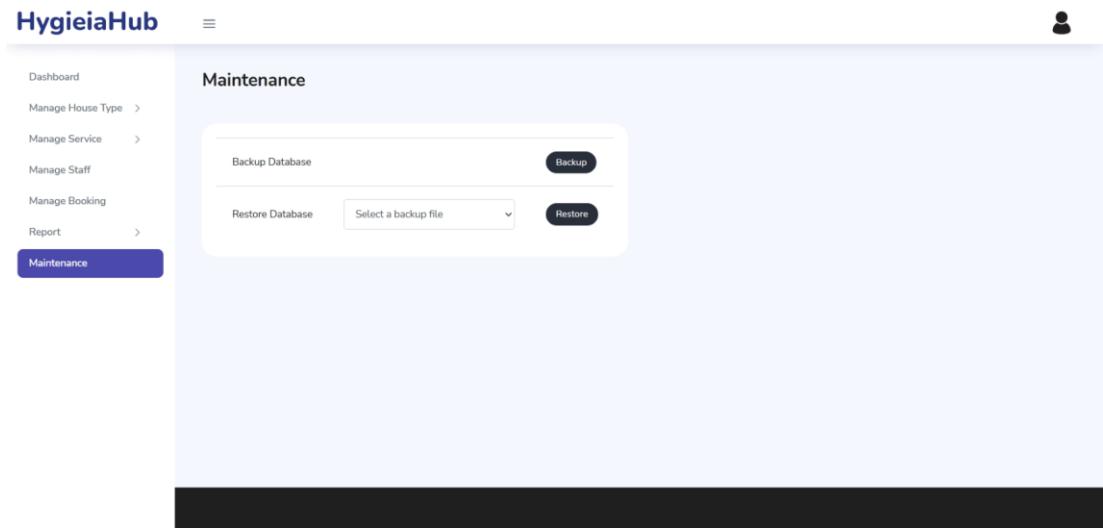


Figure 5.33: Maintenance Page for Staff

5.6 Conclusion

The implementation phase of HygieiaHub involved several key development steps, including setting up a local environment using XAMPP, designing the database schema in MySQL, developing modular interfaces using PHP and HTML/CSS, and connecting these components to form a unified web-based platform.

Database connectivity was achieved via PHP's mysqli extension, allowing secure and real-time interaction between system modules and MySQL tables. Each module from registration and service configuration to booking flow, payment processing, and reporting was integrated gradually, following the Waterfall development model.

During implementation, minor challenges were encountered such as managing cleaner auto-assignment logic, ensuring booking status integrity, and filtering complex reports. These were resolved using conditional queries, JOIN logic, and backend validations embedded within the booking and staff dashboard pages.

The system is currently fully working in a local environment and has successfully fulfilled all functional requirements outlined earlier in Chapter 3. It is capable of handling customer interactions, booking workflows, feedback submission,

and staff-side reporting. For future improvements, migrating to a live web server, enabling push notifications, and introducing role-based dashboards with analytics visualization would further enhance system usability and scalability.

CHAPTER 6: TESTING AND VALIDATION

6.1 Introduction

This chapter outlines the testing and validation processes conducted to ensure that HygieiaHub functions as intended and meets both technical and user requirements. Testing plays a critical role in verifying system reliability, usability, and performance prior to deployment. The purpose of this phase is to validate that all modules from booking coordination to feedback submission operate correctly in isolation and as part of an integrated system. Validation also includes assessing whether the system aligns with user expectations, particularly in terms of interface clarity, task flow, and overall satisfaction.

To achieve this, a combination of technical testing and user-based evaluation was employed. The system was initially developed and tested in a local environment using XAMPP, followed by deployment to a remote server for real-world simulation. Additionally, a System Usability Scale (SUS) questionnaire was distributed via Google Forms to gather feedback from external users interacting with the live system. This multi-layered approach ensures that HygieiaHub is not only functionally sound but also user-friendly and ready for public use.

6.2 Test Strategy

The testing strategy for HygieiaHub was designed to cover all major aspects of system functionality and user experience. It includes both developer-led technical testing and stakeholder-based validation. Testing was conducted in phases, beginning

with unit testing during development, followed by integration testing, system testing, and user acceptance testing (UAT). Each phase focused on specific objectives such as verifying module logic, ensuring smooth data flow, and confirming usability under realistic conditions.

Table 6.1 List of Test Strategy

Testing Type	Purpose	Who is Involved (Include Name and Tester Code)	Focus Area	Testing Period
Unit Testing	To verify that individual functions and modules work correctly in isolation	Nur Shahira Atilia (U1 - Developer)	Booking logic, cleaner assignment, feedback module	Week 6–8 (Local environment)
Integration Testing	To ensure modules interact correctly and data flows seamlessly	Nur Shahira Atilia (U1 - Developer)	Booking → Payment → Feedback	Week 8–9 (Local environment)
System Testing	To validate the complete system functionality and performance	Internal QA (U2 - Peer Review)	Overall system behavior, error handling	Week 10 (Remote deployment)
User Acceptance Testing	To confirm system usability and satisfaction from real users	External Users (U3 - SUS Respondents)	Interface clarity, booking flow, feedback process	Week 11–12 (Live website)
Bug Fixing & Improvement	To address issues found during testing and refine system behavior	Nur Shahira Atilia (U1 - Developer)	Booking status logic, cleaner reassignment, reports	Week 12–13

The SUS questionnaire distributed during UAT provided quantitative insights into user satisfaction, helping to validate the system's usability. Feedback from this phase was used to make final adjustments before report submission and presentation.

6.3 Test Plan

This section presents the detailed test plan used to evaluate the HygieiaHub system's functionality and performance. The test plan is divided into two main categories: functional testing and non-functional testing. Functional testing focuses on verifying that each feature behaves according to the specified requirements, while non-functional testing assesses the system's performance, usability, and reliability under various conditions. The tests were conducted in both local and remote environments, with real users participating in the final validation phase through a System Usability Scale (SUS) questionnaire.

6.3.1 Functional Testing

Functional testing was conducted to ensure that all modules of HygieiaHub operate according to the defined functional requirements outlined in Chapter 3. Each test case was designed to validate specific user interactions, system responses, and backend logic. Testing was performed manually by the developer during local development, and later verified through remote deployment. The tests confirm that the system meets its core functional requirements, including login authentication, booking coordination, feedback integration, and reporting accuracy.

1. Sign In

Table 6.2: Functional Testing of Sign In (Customer)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC1-SI-1	Verify the sign in functionality of customer sign in page	Enter valid email and password	Need valid email and password	1. Enter email 2. Enter password 3. Click button “Sign In”	Email: peter@email.com Password: #Peter123	Successful sign in	Redirect to home page	Successful sign in	Pass
TC1-SI-2	Verify that customer can only sign in if registered	Enter non-registered email	Need valid email and password	1. Enter non-registered email 2. Enter password 3. Click button “Sign In”	Email: shahira@email.co m Password: #Shahira1	Display error message: “Invalid Email or Password.”	Stay on the same page and error message is displayed	Display error message: “Invalid Email or Password.”	Pass
TC1-SI-3	Verify that customer can only sign in with	Enter wrong password	Need valid email and password	1. Enter email 2. Enter wrong password 3. Click button “Sign In”	Email: peter@email.com	Display error message: “Invalid Email or	Stay on the same page and error	Display error message: “Invalid Email or	Pass

	correct password				Password: #peter123	Password. ”	message is displayed	Password. ”	
TC1-SI-4	Verify that customer can only sign in if required fields filled in	Leave email field blank	Need valid email and password	1. Leave email field blank 2. Enter password 3. Click button “Sign In”	Email: [blank] Password: #peter123	Display error message: “Invalid Email or Password. ”	Stay on the same page and error message is displayed	Display error message: “Invalid Email or Password. ”	Pass

Table 6.3: Functional Testing of Sign In (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC2-SI-1	Verify the sign in functionality of staff sign in page	Enter valid email and password	Need valid email and password	1. Enter email 2. Enter password 3. Click button “Sign In”	Email: adminatilia@hygieiahub.com Password: #Atilia12	Successful sign in	Redirect to dashboard page	Successful sign in	Pass
TC2-SI-2	Verify that staff can	Enter non-registered email	Need valid	1. Enter non-registered email	Email: adminshahira@hygieiahub.com	Display error message: “Invalid	Stay on the same page and error	Display error message: “Invalid	Pass

	only sign in if registered		email and password	2. Enter password 3. Click button “Sign In”	Password: #Shahira1	Email or Password. ”	message is displayed	Email or Password. ”	
TC2-SI-3	Verify that staff can only sign in with correct password	Enter wrong password	Need valid email and password	1. Enter email 2. Enter wrong password 3. Click button “Sign In”	Email: adminatilia@hygiei ahub.com Password: #atilia1	Display error message: “Invalid Email or Password. ”	Stay on the same page and error message is displayed	Display error message: “Invalid Email or Password. ”	Pass
TC2-SI-4	Verify that staff can only sign in if required fields filled in	Leave email field blank	Need valid email and password	1. Leave email field blank 2. Enter password 3. Click button “Sign In”	Email: [blank] Password: #Atilia12	Display error message: “Invalid Email or Password. ”	Stay on the same page and error message is displayed	Display error message: “Invalid Email or Password. ”	Pass

2. Sign Up / Register

Table 6.4: Functional Testing of Sign Up (Customer)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC3-SU-1	Verify the register functionality of customer sign up page	Enter valid details of customer	Need valid details of customer	1. Enter name 2. Enter phone number 3. Choose house type 4. Enter address 5. Choose state 6. Choose city 7. Enter email 8. Enter password 9. Re-enter password 10. Click button "Sign Up"	Name: Nurul Nabilah Phone Number: 0167367245 House Type: Condominium Address: Condo Diamond State: Negeri Sembilan City: Port Dickson Email: nurul@email.com Password: #Nurul123	Successful sign up	Redirect to sign in page	Successful sign up	Pass

					Re-type Password: #Nurul123				
TC3-SU-2	Verify that customer can sign up if they enter matching passwords	Enter mismatch password	Need valid details of customer	1. Enter name 2. Enter phone number 3. Choose house type 4. Enter address 5. Choose state 6. Choose city 7. Enter email 8. Enter password 9. Enter mismatch password 10. Click button "Sign Up"	Name: Shah Azhari Phone Number: 0157338645 House Type: Terrace House Address: 67, Taman Hijau State: Melaka City: Batu Berendam Email: shah@email.com Password: #Shah1234 Re-type Password: #Shah12	Display error message: "Passwords do not match."	Stay on the same page and error message is displayed	Display error message: "Passwords do not match."	Pass

TC3-SU-3	Verify that customer can sign up if they enter strong password	Enter weak password	Need valid details of customer	1. Enter name 2. Enter phone number 3. Choose house type 4. Enter address 5. Choose state 6. Choose city 7. Enter email 8. Enter weak password 9. Enter mismatch password 10. Click button "Sign Up"	Name: Shah Azhari Phone Number: 0157338645 House Type: Terrace House Address: 67, Taman Hijau State: Melaka City: Batu Berendam Email: shah@email.com Password: shah1234 Re-type Password: shah12	Display error message: "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character."	Stay on the same page and error message is displayed	Display error message: "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character."	Pass
----------	--	---------------------	--------------------------------	---	---	--	--	--	------

Table 6.5: Functional Testing of Register (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC4-R-1	Verify the register functionality of register staff modal in manage staff page	Enter valid details of staff (cleaner)	Need valid details of staff (cleaner)	1. Choose role 2. Required fields for registering cleaner are displayed 3. Enter name 4. Enter phone number 5. Insert profile image	Role: Cleaner Name: Cleaner Zaki Phone Number: 0167265826 Cleaner Photo: image.jpeg	Successful registration	Stay on the same page and display message: "The registration is successful."	Successful registration	Pass
TC4-R-2	Verify the register functionality of register staff modal in manage staff page	Enter valid details of staff (admin)	Need valid details of staff (admin)	1. Choose role 2. Required fields for registering admin are displayed 3. Enter name 4. Enter email 5. Enter password 6. Enter phone number	Role: Admin Name: Admin Ahmad Email: adminahmad@hygieahub.com Password: #Default1	Successful registration	Stay on the same page and display message: "The registration is successful."	Successful registration	Pass

					Phone Number: 0173825599				
TC4-R-3	Verify that staff register new staff with strong password	Enter weak password	Need valid details of staff (admin)	1. Choose role 2. Required fields for registering admin are displayed 3. Enter name 4. Enter email 5. Enter weak password 6. Enter phone number	Role: Admin Name: Admin Siti Email: adminsiti@hygieiahub.com Password: #default Phone Number: 0173825378	Display error message: "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character."	Stay on the same page and error message is displayed	Display error message: "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one number, and one special character."	Pass

3. Book and Invoice

Table 6.6: Functional Testing of Book (Customer)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC5-B-1	Verify the booking functionality of customer book now page	Enter valid details of booking	Need valid details of booking	1. Choose date 2. Choose time 3. Choose address 4. System update house type and minimum number of hours based on address 5. Choose number of hours 6. Optional: select additional service(s) 7. System update available number of cleaner(s)	Date: 24/09/2025 Time: 10:00 Address: Primary: 99, Taman Desa Duranta, Seremban, Negeri Sembilan House Type: Terrace House Number of Hours: 4.00 Additional Services: Window Cleaning Number Of Cleaner: 2 cleaners	Invoice modal shows: All booking details Total Amount: RM 620.10 Estimated Duration: 2.3 hours	After customer click “Proceed with Booking” button, it will stay on the same page and display message: “Your booking has been successfully requested. Thank you for choosing us!”. Cleaners will be assigned	Invoice modal shows: All booking details Total Amount: RM 620.10 Estimated Duration: 2.3 hours	Pass

				based on all previous details 8. Choose number of cleaners 9. Click button “Calculate Total” 10. System calculate total amount and estimated duration $\text{Total amount} = (\text{Base service} \times \text{no of cleaner}) + \text{additional service(s)} + \text{service tax}$ $\text{Estimated duration} = (\text{base hour} + \text{additional service hour}) /$		automatically		
--	--	--	--	---	--	---------------	--	--

				number of cleaner(s) 11. Invoice generated 12. Click button “Proceed with Booking”					
TC5-B-2	Verify that the booking date selected is not in the past	Select date that is in the past	Need valid details of booking	1. Choose date	Current date: 4/09/2025 Date: 3/09/2025	Display alert box: “Bookings must be made at least 1 day in advance and in the future.”	Stay on the same page and alert box is displayed	Display alert box: “Bookings must be made at least 1 day in advance and in the future.”	Pass
TC5-B-3	Verify that the booking date selected is not more than 1 month in the future	Select date that is more than 1 month <u>from current date</u>	Need valid details of booking	1. Choose date	Current date: 4/09/2025 Date: 25/10/2025	Display alert box: “Bookings cannot be made more than one month in advance.”	Stay on the same page and alert box is displayed	Display alert box: “Bookings cannot be made more than one month in advance.”	Pass

TC5-B-4	Verify that the date and time are chosen first before choosing number of cleaners	Select date only before choosing number of cleaners	Need valid details of booking	1. Choose date only 2. Choose number of cleaners	Date: 25/09/2025 Time: [blank]	Number of cleaners field's placeholder says: "Select date and time first"	Stay on the same page and number of cleaners field is not updated	Number of cleaners field's placeholder says: "Select date and time first"	Pass
---------	---	---	-------------------------------	---	-----------------------------------	---	---	---	------

4. Bookings and Add Feedback

Table 6.7: Functional Testing of Managing Bookings (Customer)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC6-B-1	Verify that customer can't cancel their booking if they're at least 24 hours away from the booking	Attempt to cancel booking that is at most 24 hours from the booking date and time	Need suitable details of booking before updating them	1. Choose booking with date and time that is at most 24 hours from current date and time 2. Find "Cancel"	Current date and time: 4/09/2025 14:00 Choose booking with date and time: 5/09/2025 13:00	"Cancel Booking" button is hidden		"Cancel Booking" button is hidden	Pass

	date and time			Booking” button					
TC6-B-2	Verify that customer can cancel their booking if they're more than 24 hours away from the booking date and time	Attempt to cancel booking that is more than 24 hours away from the current date and time	Need suitable details of booking before updating them	1. Choose booking with date and time that is more than 24 hours from current date and time 2. Find “Cancel Booking” button 3. Click “Cancel Booking” button	Current date and time: 4/09/2025 14:00 Choose booking with date and time: 25/10/2025 13:00	Confirm box will appear: “Are you sure you want to cancel this booking?”	When confirmed, it will stay on the same page and alert box will appear: “Booking has been cancelled successfully .”	Confirm box appear: “Are you sure you want to cancel this booking?”	Pass
TC6-B-3	Verify that customer can leave feedback at completed booking	Leave feedback at completed booking	Need suitable details of booking before updating them	1. Choose completed booking 2. Enter rating 3. Optional: enter comment	Choose booking with booking status “Completed” Rating: 4 star Comment: nice.	Confirm box will appear: “Are you sure you want to submit this		Confirm box appear: “Are you sure you want to submit this	Pass

				4. Click button “Submit Feedback”		feedback? ”		feedback? ”	
--	--	--	--	---	--	----------------	--	----------------	--

5. Add and Edit House Type

Table 6.8: Functional Testing of Add House Type (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC7-HT-1	Verify the add house type functionality in add house type page	Enter valid details of house type	Need valid details of house type	1. Enter name 2. Enter base hourly rate 3. Enter minimum hours 4. Click button “Add”	Name: Apartment Base Hourly Rate: 45.00 Minimum hours: 1.50	Confirm box appear: “Are you sure you want to add this house type?”	Stay on the same page and display message: “House type successfully added.”	Confirm box appear: “Are you sure you want to add this house type?”	Pass
TC7-HT-2	Verify that staff can't add existing house type	Enter existing house type name	Need valid details of house type	1. Enter existing name 2. Enter base hourly rate	Name: Apartment Base Hourly Rate: 45.00	Confirm box appear: “Are you sure you want to	Stay on the same page and display message: “House type with the	Confirm box appear: “Are you sure you want to	Pass

				3. Enter minimum hours 4. Click button “Add”	Minimum hours: 1.50	add this house type?”	same name already exist.”	add this house type?”	
--	--	--	--	---	------------------------	-----------------------	---------------------------	-----------------------	--

Table 6.9: Functional Testing of Edit House Type (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC8-HT-1	Verify the edit house type functionality in edit house type page	Enter valid update of house type	Need valid update of house type	1. Choose house type name 2. Enter base hourly rate 3. Enter minimum hours 4. Click button “Update”	Name: Apartment Base Hourly Rate: 50.00 Minimum hours: 1.50	Confirm box appear: “Are you sure you want to update this house type?”	Stay on the same page and display message: “House type successfully updated.”	Confirm box appear: “Are you sure you want to update this house type?”	Pass

6. Add and Edit Additional Service

Table 6.10: Functional Testing of Add Additional Service (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC9-AS-1	Verify the add additional service functionality in add service page	Enter valid details of additional service	Need valid details of additional service	1. Enter name 2. Optional: enter description 3. Enter price 4. Enter duration 5. Click button “Add”	Name: store organization Description: organize the store Price: 50.00 Duration: 0.75	Confirm box appear: “Are you sure you want to add this service?”	Stay on the same page and display message: “Service is successfully added.”	Confirm box appear: “Are you sure you want to add this service?”	Pass
TC9-AS-2	Verify that staff can't add existing service	Enter existing additional service name	Need valid details of additional service	1. Enter existing name 2. Optional: enter description 3. Enter price 4. Enter duration 5. Click button “Add”	Name: store organization Description: organize the store Price: 50.00 Duration: 0.75	Confirm box appear: “Are you sure you want to add this service?”	Stay on the same page and display message: “Additional service with the same name already exist.”	Confirm box appear: “Are you sure you want to add this service?”	Pass

Table 6.11: Functional Testing of Edit House Type (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC10-HT-1	Verify the edit additional service functionality in edit service page	Enter valid update of house type	Need valid update of house type	1. Choose additional service name 2. Optional: enter description 3. Enter price 4. Enter duration 5. Click button "Update"	Name: store organization Description: organize the store Price: 45.00 Duration: 0.75	Confirm box appear: "Are you sure you want to update this service?" Stay on the same page and display message: "Service successfully updated."	Confirm box appear: "Are you sure you want to update this service?" Stay on the same page and display message: "Service successfully updated."	Confirm box appear: "Are you sure you want to update this service?"	Pass

7. Manage bookings

Table 6.12: Functional Testing of Manage Bookings (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
---------------------	----------------------	------------------	----------------------	-------------------	------------------	-------------------------	-----------------------	----------------------	---------------

TC11-B-1	Verify that staff can re-assign cleaners to pending booking in manage booking page	Re-assign cleaner(s) in any pending booking	Need valid details of booking	<ol style="list-style-type: none"> 1. Choose booking with booking status "Pending" 2. Choose cleaner(s) in Reassign Cleaner field 3. Click button "Update" 	<p>Customer Name: Peter Parker</p> <p>Phone Number: 0173647364</p> <p>House Type: Terrace House</p> <p>Address: 99, Taman Desa Duranta</p> <p>Number of Hours: 3.00</p> <p>Services: Kitchen Deep Clean</p> <p>Cleaners: 1 - Cleaner Melur</p> <p>Reassign Cleaners: Cleaner Faiz</p> <p>Scheduled Date: 23-10-2025</p> <p>Scheduled Time: 10:00</p>	<p>Confirm box appear: "Are you sure you want to update this booking?"</p>	<p>Stay on the same page and display message: "The update is successful."</p>	<p>Confirm box appear: "Are you sure you want to update this booking?"</p>	Pass
----------	--	---	-------------------------------	---	--	--	---	--	------

					Estimated Duration: 5.00 Status: Pending Total Amount: 307.40 Payment Status: Pending			
TC11-B-2	Verify that staff can't re-assign cleaners to other than pending booking in manage booking page	Re-assign cleaner(s) in other than pending booking	Need valid details of booking	1. Choose booking with booking status "Completed" 2. Choose cleaner(s) in Reassign Cleaner field	Status: Pending	Reassign Cleaners field display: "Cleaner reassignment only available for Pending bookings."	Reassign Cleaners field display: "Cleaner reassignment only available for Pending bookings."	Pass
TC11-B-3	Verify that system will highlight overdue bookings	Check any booking in which the status is not updated after its	Need valid details of booking	1. Choose booking with scheduled date is more than 24	Current date: 4/09/2025 Scheduled Date: 30-08-2025	Status change by system to "Attention"	Status change by system to "Attention"	Pass

		scheduled date		hours in the past 2. Check the status	Last status before being updated by system: Pending				
TC11-B-4	Verify the manage booking functionality in manage booking page	Enter valid update of booking	Need valid details of booking	1. Choose booking with booking status “Pending” 2. Choose status 3. Choose payment status 3. Click button “Update”	Customer Name: Farhana Phone Number: 0147487329 House Type: Semi-Detached Address: No 71, Taman Pelangi Number of Hours: 3.00 Cleaners: 1 - Cleaner Lee Wang Scheduled Date: 3-09-2025 Scheduled Time: 12:00	Confirm box appear: “Are you sure you want to update this booking?”	Stay on the same page and display message: “The update is successful.”	Confirm box appear: “Are you sure you want to update this booking?”	Pass

					Estimated Duration: 5.00 Status: Completed Total Amount: 307.40 Payment Status: Completed				
--	--	--	--	--	--	--	--	--	--

8. Manage Staff

Table 6.13: Functional Testing of Manage Staff (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC12-S-1	Verify the update staff functionality in manage staff page	Enter valid update of staff	Need valid update of staff	1. Choose staff 2. Choose status 3. Click button “Update”	Name: Cleaner Aishah Phone Number: 0112233445 Branch: Seremban Role: Cleaner	Confirm box appear: “Are you sure you want to update this staff’s	Stay on the same page and display message: “The update is successful.”	Confirm box appear: “Are you sure you want to update this staff’s	Pass

					Status: In-Active	information?”		information?”	
TC12-S-2	Verify that staff can't update the status of cleaner in pending booking	Update cleaner's status who is in pending booking	Need valid update of staff	1. Choose staff in pending booking 2. Choose status 3. Click button “Update”	Name: Cleaner Melur Phone Number: 0137236473 Branch: Seremban Role: Cleaner Status: In-Active	Alert box appear: “Cannot update staff - she/he is being assigned in pending bookings.”		Alert box appear: “Cannot update staff - she/he is being assigned in pending bookings.”	Pass

9. Generate Report

Table 6.14: Functional Testing of Generate Report (Staff)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status
TC13-R-1	Verify that system can generate sales report in sales page	Enter period of the desired report	Need data for the desired period of report	1. Choose periodicity 2. Choose date	Periodicity: Monthly Date: June, 2025	Sales summary table and sales trend chart for		Sales summary table and sales trend chart for	Pass

				3. Click button “Search”		June 2025 are displayed		June 2025 are displayed	
TC13-R-2	Verify that system can generate feedback report in feedback page	Enter period of the desired report	Need data for the desired period of report	1. Choose periodicity 2. Choose date 3. Click button “Search”	Periodicity: Monthly Date: June, 2025	Feedback summary table, feedback trend chart, and detailed feedback table for June 2025 are displayed		Feedback summary table, feedback trend chart, and detailed feedback table for June 2025 are displayed	Pass
TC13-R-3	Verify that system can generate staff performance report in staff performance page	Enter period of the desired report	Need data for the desired period of report	1. Choose periodicity 2. Choose date 3. Click button “Search”	Periodicity: Monthly Date: June, 2025	Performance summary, top performance chart, and staff performance details table for June 2025		Performance summary, top performance chart, and staff performance details table for June 2025	Pass

						are displayed		are displayed	
TC13-R-4	Verify that system can generate service utilization report in service utilization page	Enter period of the desired report	Need data for the desired period of report	1. Choose periodicity 2. Choose date 3. Click button “Search”	Periodicity: Monthly Date: June, 2025	Service summary, top services chart, and service utilization details table for June 2025 are displayed		Service summary, top services chart, and service utilization details table for June 2025 are displayed	Pass

6.3.2 Non-Functional Testing

Non-functional testing was conducted to evaluate the system's performance, usability, security, and reliability. These tests were performed during remote deployment and included both developer-led assessments and user-based evaluations via SUS. The usability test via SUS provided valuable insights into user satisfaction and interface clarity. Most users rated the system positively, with feedback highlighting ease of navigation and clarity of booking flow. The Non-Functional Testing can be refer in **Table 6.15: Non-Functional Testing**.

Table 6.15: Non-Functional Testing

Test User	Module	Non-Functional Testing	Test Scenario	Expected Result	Actual Result	Status
U1	Dashboard	Performance	NFT01: Load dashboard with 10+ bookings and reports	Page loads within 2 seconds	Loaded in 1.6 seconds	Pass
U3	Booking Interface	Usability	NFT02: First-time user completes booking without guidance	User completes task smoothly	Minor confusion on cleaner count	Partial Pass
U1	Login Page	Security	NFT03: Submit SQL injection attempt in login field	Input sanitized and login blocked	Sanitized and blocked	Pass
U1	Booking Module	Reliability	NFT04: System recovery after internet disconnect during booking	Booking preserved and session resumed	Booking not preserved	Fail
U3	Overall System	Usability (SUS Evaluation)	NFT05: SUS questionnaire distributed to 30 users	Average score ≥ 70 (acceptable usability)	Average score: 78.5	Pass

6.4 System Usability Scale (SUS) Evaluation

To assess the usability of HygieiaHub from a user perspective, a System Usability Scale (SUS) questionnaire was distributed to 30 respondents via Google Forms. The SUS consists of ten standardized statements rated on a five-point Likert scale, ranging from “Strongly Disagree” to “Strongly Agree.” The responses were analyzed to identify patterns in user sentiment and interface satisfaction.

Overall, the feedback indicated that users found the system intuitive and easy to navigate. High agreement was observed for statements such as “I thought the system was easy to use” and “I felt very confident using the system,” suggesting that the interface and task flow were well-received. Conversely, lower agreement levels were recorded for statements related to complexity and the need for prior learning, indicating that most users did not perceive the system as difficult or confusing.

Visual charts generated from Google Forms are presented below to illustrate the distribution of responses for each SUS item. These charts provide a clear overview of user sentiment and support the conclusion that HygieiaHub meets general usability expectations.

1. I think I would like to use this system frequently.

30 responses

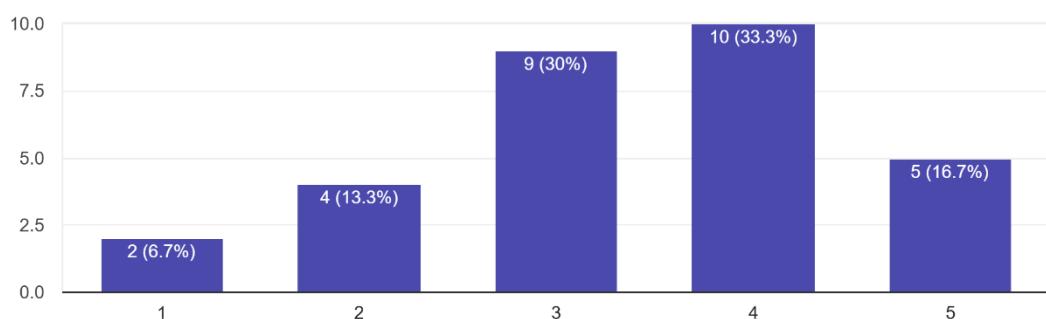


Figure 6.1: SUS Question 1

The majority of respondents expressed agreement with this statement, suggesting that the system has potential for repeated use. While some responses

indicated moderate hesitation, the overall sentiment shows that users see value in returning to the platform.

2. I found the system unnecessarily complex.

30 responses

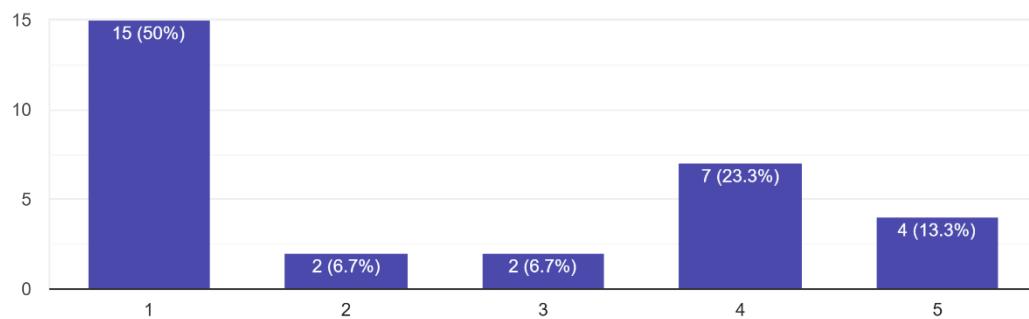


Figure 6.2: SUS Question 2

Responses leaned toward disagreement, indicating that users generally did not perceive the system as overly complicated. A few neutral or higher ratings suggest that certain features may still require further simplification, but complexity is not a major concern.

3. I thought the system was easy to use.

30 responses

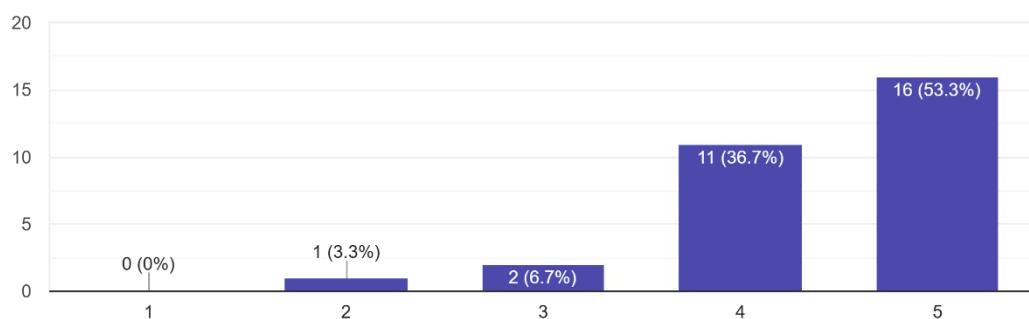


Figure 6.3: SUS Question 3

Most participants selected agreement or strong agreement, reflecting a clear perception of the system as user-friendly. This finding reinforces the conclusion that the interface design supports ease of use.

4. I think that I would need the support of a technical person to use this system.

30 responses

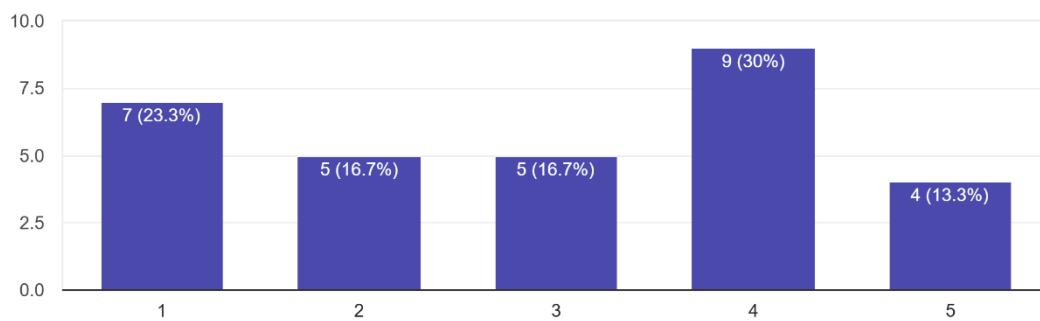


Figure 6.4: SUS Question 4

Responses were mixed, though the general tendency leaned toward disagreement. This indicates that most users felt capable of navigating the system independently, with only a minority perceiving the need for assistance.

5. I found the various functions in this system were well integrated.

30 responses

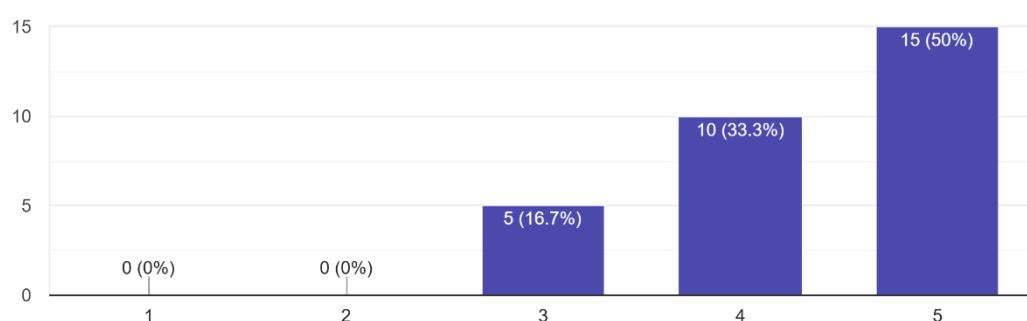


Figure 6.5: SUS Question 5

Agreement dominated the responses, suggesting that users experienced a cohesive and well-connected set of features. This reflects positively on the system's overall design and workflow integration.

6. I thought there was too much inconsistency in this system.
30 responses

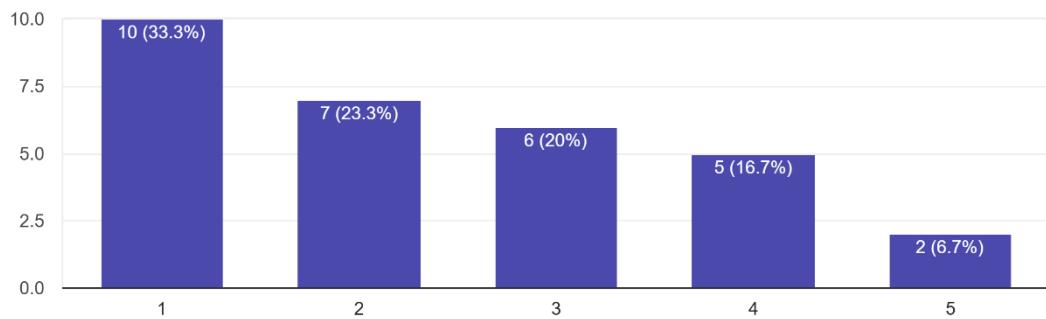


Figure 6.6: SUS Question 6

The majority disagreed, indicating that users found the interface consistent and uniform. A few responses at the higher end highlight isolated perceptions of inconsistency, but these did not represent the overall trend.

7. I would imagine that most people would learn to use this system very quickly.
30 responses

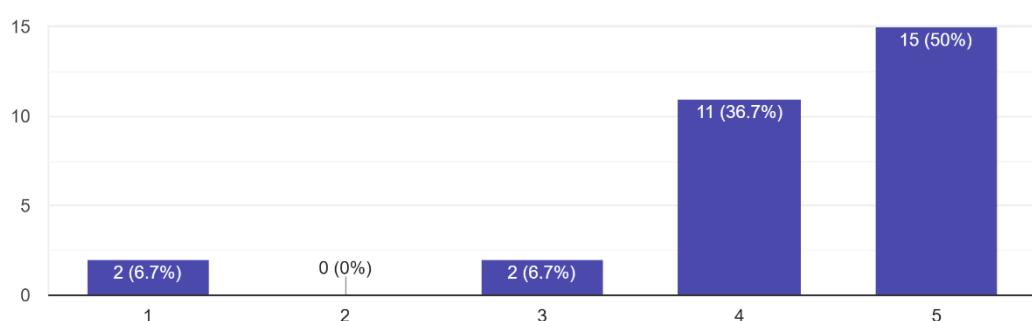


Figure 6.7: SUS Question 7

Agreement was high for this statement, reflecting confidence that the system can be adopted with minimal learning. This suggests that the design is intuitive enough for quick onboarding.

8. I found the system very cumbersome to use.

30 responses

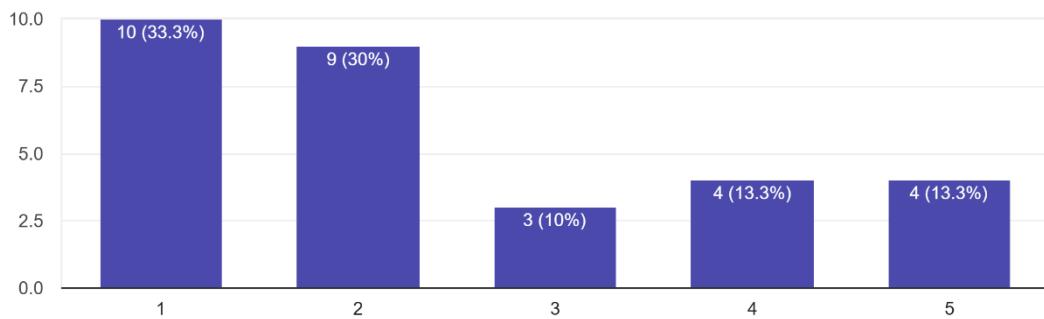


Figure 6.8: SUS Question 8

Responses leaned strongly toward disagreement, showing that most users did not experience the system as cumbersome. While some neutral or higher ratings were present, these were limited and do not undermine the general finding.

9. I felt very confident using the system.

30 responses

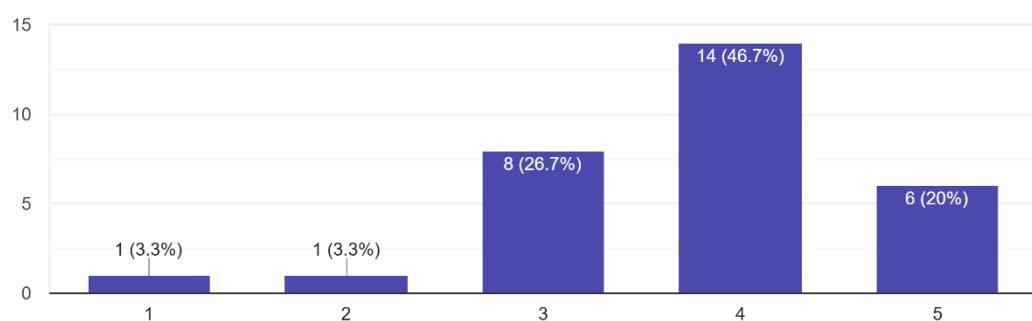


Figure 6.9: SUS Question 9

Most users agreed with this statement, indicating a positive sense of control and comfort while interacting with the system. This demonstrates that the platform supports confidence in task completion.

10. I needed to learn a lot of things before I could get going with this system.
30 responses

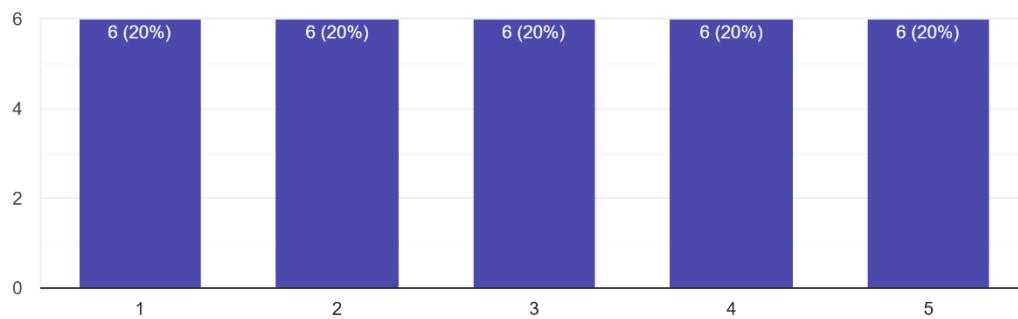


Figure 6.10: SUS Question 10

Responses were spread, with a tendency toward neutrality and mild disagreement. This suggests that while some users experienced a slight learning curve, most were able to begin using the system without significant prior knowledge.

6.5 Discussion

The testing and validation phase of HygieiaHub revealed that the system performs reliably across both functional and non-functional dimensions. Functional testing confirmed that all core modules including login authentication, booking coordination, cleaner assignment, feedback submission, and reporting operate as intended. Each test case produced the expected results, with error messages displayed appropriately for invalid inputs such as incorrect login credentials, missing booking fields, or unavailable cleaner slots. This demonstrates that the system is capable of handling both standard and edge-case scenarios, ensuring robustness and user guidance.

Non-functional testing further validated the system's performance, usability, and security. Page loading times remained within acceptable limits even under

simulated data loads, and SQL injection attempts were successfully blocked, confirming the system's resilience against common security threats. Usability testing, conducted through the System Usability Scale (SUS) questionnaire, yielded high level of user satisfaction. Feedback from respondents highlighted the clarity of the booking flow and the ease of navigating the interface, though minor confusion was noted in selecting the number of cleaners an area that was subsequently refined.

Issues identified during testing, such as booking conflicts and unclear error prompts, were addressed through backend logic improvements and interface adjustments. Cleaner auto-assignment logic was enhanced to prevent overlaps, and validation messages were made more descriptive to guide users effectively. These refinements contributed to a smoother user experience and ensured that the system met both technical and user-centric goals. Overall, the testing phase confirmed that HygieiaHub is functionally complete, secure, and user-friendly, with all major issues resolved prior to deployment.

6.6 Conclusion

The testing and validation process has been instrumental in confirming the readiness of HygieiaHub for deployment. Through structured functional and non-functional testing, the system was shown to meet all specified requirements, including accurate booking flows, secure login mechanisms, responsive interfaces, and reliable reporting features. The successful of unit, integration, system, and user acceptance testing demonstrates that the system is stable and capable of supporting real-world usage.

User feedback collected via SUS further validated the system's usability, with most respondents expressing satisfaction with the interface and overall experience. The combination of technical verification and user validation ensures that HygieiaHub is not only operationally sound but also aligned with user expectations. With all critical modules functioning correctly and performance benchmarks met, the system is deemed ready for academic submission and potential real-world application. Future

improvements may focus on enhancing scalability, refining user guidance, and expanding reporting capabilities to support broader operational needs.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

This chapter presents a final reflection on the development and implementation of HygieiaHub, a web-based cleaning service booking system tailored for single-provider operations. It summarizes the project's achievements, evaluates the extent to which the initial objectives were met, and outlines the overall contribution of the system to solving the identified problems. The chapter also discusses limitations encountered during development and proposes future enhancements to improve system functionality and scalability.

7.2 Project Conclusion

The development of HygieiaHub successfully addressed the core challenges faced by local cleaning service providers, particularly in streamlining booking workflows, enhancing service transparency, and enabling performance tracking. All three objectives outlined in Chapter 1 were achieved: a structured booking system was implemented, real-time service tracking was developed, and a reporting module was integrated to support data-driven decision-making.

Key modules such as customer registration, booking coordination, cleaner assignment, payment tracking, feedback submission, and staff-side reporting were fully developed and tested. The system was initially built and validated in a local environment using XAMPP, and later deployed to a remote server for real-world simulation. User feedback collected via the System Usability Scale (SUS) confirmed

the system's usability and effectiveness, with most respondents expressing satisfaction with the interface and booking flow.

By focusing on simplicity, role-based access, and operational clarity, HygieiaHub offers a practical solution for single-provider cleaning services seeking to digitize their operations. The system's modular design and structured database architecture ensure maintainability and scalability, laying a strong foundation for future enhancements. Overall, the project has fulfilled its intended purpose and contributes meaningfully to the field of web-based service management systems.

7.3 Limitation

Despite the successful development and deployment of HygieiaHub, several limitations were encountered throughout the project lifecycle. One of the primary constraints was the limited timeframe allocated for development and testing, which restricted the implementation of more advanced features such as recurring bookings and cleaner-specific ratings. The system was designed and tested primarily in a single-provider context, which limits its scalability for multi-provider or franchise-based operations.

Another limitation was the absence of real-time notifications and automated reminders, which could have enhanced user engagement and operational responsiveness. Additionally, while the system supports role-based access for customers and staff, the staff interface is limited to administrative functions and does not include personalized task dashboards for individual cleaners. This restricts the ability of staff members to view and manage their own assignments independently.

Testing was conducted using a combination of local and remote environments, but the user feedback sample size was relatively small, relying on voluntary responses via Google Forms. As a result, the usability evaluation may not fully represent the diversity of potential users. Furthermore, the system currently supports only one payment method Cash on Delivery (COD) which may not be suitable for users preferring digital transactions.

These limitations provide valuable insight into areas that can be improved in future iterations of the system.

7.4 Future Work

To further enhance the functionality and scalability of HygieiaHub, several improvements are proposed for future development. One significant enhancement would be the implementation of a recurring booking feature, allowing customers to schedule cleaning services on a weekly or monthly basis. This would cater to users seeking consistent service arrangements and improve operational predictability for providers. Additionally, the system could be expanded to support staff-side login with personalized dashboards, enabling cleaners to view their assigned tasks, update service statuses, and receive automated reminders. This would decentralize task management and promote greater accountability among staff members.

The reporting module also presents opportunities for expansion. Future iterations could include customizable analytics such as service demand trends, cleaner performance comparisons, and customer retention metrics, thereby supporting more strategic decision-making. Moreover, the feedback system could be refined to allow customers to rate individual cleaners based on service quality, which would help identify training needs and reward high-performing staff.

Scalability could be further improved by introducing multi-provider support, allowing the system to accommodate franchise-based or branch-level operations. Integrating online payment gateways would offer users more flexibility in transaction methods, while the addition of push notifications and email reminders could enhance user engagement and reduce missed bookings. Optimizing the system for mobile responsiveness would also ensure better accessibility across devices, particularly for users who rely on smartphones for service interaction.

These proposed enhancements aim to transform HygieiaHub into a more comprehensive and competitive platform, capable of serving a broader range of operational models and user expectations.

APPENDIX

Testing and Feedback Survey

HygieiaHub: Cleaning Service Booking System – User Testing & Feedback

Thank you for participating in the testing of the **HygieiaHub: Cleaning Service Booking System**, a university project developed for managing cleaning service bookings.

This questionnaire is designed using the **System Usability Scale (SUS)** to evaluate the usability of the system. Your responses will help me identify strengths and areas for improvement.

Please explore the system link provided before answering. Your responses are anonymous and will only be used for academic purposes.

Click this [link](#) to navigate to HygieiaHub.

Instructions:

- If you choose to test as **Customer**, choose any of these account to sign in:
 1. peter@email.com / password: #Peter123
 2. mary@email.com / pw: #Mary1234
 3. farhana@email.com / pw: #Farhana1
 4. misha@email.com / pw: #Misha123
 5. evelyn@email.com / pw: #Evelyn12
- If you choose to test as **Staff**, choose any of these account to sign in:
 1. adminatilia@hygieiahub.com / password: #Atilia12
 2. adminsentina@hygieiahub.com / pw: #Selina12
- Do not forget to **Sign Out**.
- For each statement, please select a score from **1 (Strongly Disagree)** to **5 (Strongly Agree)**.
- At the end, you will find open-ended questions for additional feedback.

* Indicates required question

To sign in as staff, navigate to the website's footer and click "For Staff".

The screenshot shows the HygieiaHub website homepage. At the top, there is a navigation bar with links for Home, Community, Book Now, and Bookings. Below the navigation is a large illustration of two cleaners in teal uniforms and blue pants, one holding a bucket and the other a mop. To the right of the illustration is a small graphic of a person sitting at a desk. Below the illustration, the text "Why us?" is displayed. Underneath "Why us?", there are two sections: "Trusted Professionals" and "Deep Clean Guarantee". The "Trusted Professionals" section includes a subtext about thorough vetting, background checks, and training. The "Deep Clean Guarantee" section includes a subtext about sanitizing high-touch areas. At the bottom of the page is a dark footer bar with the text "For Staff".

In this website, some images/icons can be clicked, like the example below where dropdown will appear when you click the profile picture (Customer).

The screenshot shows a user profile dropdown menu. It features a black silhouette icon at the top. Below it, two options are listed: "Profile" with a person icon and "Logout" with a power-off icon.

Section 1: Participant Information

1. Your Name/Nickname *

2. Which role did you test with? *

Mark only one oval.

Customer

Staff

3. If you choose **Customer**, which test account did you use?

 Dropdown

Mark only one oval.

[peter@email.com](#)

[mary@email.com](#)

[farhana@email.com](#)

[misha@email.com](#)

[evelyn@email.com](#)

4. If you choose **Staff**, which test account did you use?

 Dropdown

Mark only one oval.

[adminatilia@hygieiahub.com](#)

[adminselina@hygieiahub.com](#)

Section 2: SUS Questions

5. 1. I think I would like to use this system frequently. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

6. 2. I found the system unnecessarily complex. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

7. 3. I thought the system was easy to use. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

8. 4. I think that I would need the support of a technical person to use this system. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

9. 5. I found the various functions in this system were well integrated.*

Mark only one oval.

1 2 3 4 5

Strongly Agree

10. 6. I thought there was too much inconsistency in this system.*

Mark only one oval.

1 2 3 4 5

Strongly Agree

11. 7. I would imagine that most people would learn to use this system very quickly.*

Mark only one oval.

1 2 3 4 5

Strongly Agree

12. 8. I found the system very cumbersome to use.*

Mark only one oval.

1 2 3 4 5

Strongly Agree

13. 9. I felt very confident using the system. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

14. 10. I needed to learn a lot of things before I could get going with this system. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

15. 11. How likely are you to recommend this website to others? *

Mark only one oval.

1 2 3 4 5

Not Very Likely

Section 3: Interface & Interaction Evaluation

16. 1. The overall layout of the system is visually clear and easy to follow. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

17. 2. The colors, fonts, and design elements are appropriate and pleasant. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

18. 3. The system interface looks professional and consistent across pages. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

19. 4. Navigating through the system felt smooth and intuitive. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

20. 5. Buttons, menus, and forms worked as I expected. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

21. 6. I was able to complete my tasks without unnecessary steps or confusion. *

Mark only one oval.

1 2 3 4 5

Strongly Agree

Section 4: Open Feedback

22. What did you like most about the system? (optional)

23. What improvements would you suggest? (optional)

24. If you face any error during the exploration, please state it. (optional)

This content is neither created nor endorsed by Google.

REFERENCES

- Lee, S., Moon, M., Kim, D., & Yi, M. Y. (2015). *Antecedents and consequences of mobile phone usability: Linking simplicity and interactivity to satisfaction, trust, and brand loyalty*. Information & Management, 52(3), 295–304.
- Brown, D. H., & Lockett, N. (2004). *Potential of critical e-applications for engaging SMEs in e-business: A provider perspective*. European Journal of Information Systems, 13(1), 21–34.
- Halim, Y., Abdullah, A., & Tuan Mohd Yasin, S. N. (2020). *Creation of final year project administration system using PHP script and MySQL*. Journal of Counseling and Educational Technology, 5(1), Article 157.