**Cleaning Service Management System**

**Project Report**

A blue and white logo

Description automatically generated with low confidence

Sri Lanka Institute of Information Technology

IT2080 Information Technology Project

**Group ITP\_WD\_B02\_04**

IT21377730 – Ameen F.A

IT21226632 – Weerasinghe W.W.A.B.M

IT21063596 – Fernando W.S.K

IT21190148 – Jayathilaka W.I.S

IT21257018 – Rathnayaka K.P.S.K

IT21027574 – Samarathunga S.P.V.S

IT21375446 – Munasinghe I.M.S.D.T

IT21316722 – Jayathilaka A.G.V.D.T

May 2023

# Declaration

This project report is our original work, and the content is not plagiarized from any other resource. References for all the content taken from external resources are correctly cited. To the best of our knowledge, this report does not contain any material published or written by third parties, except as acknowledged in the text.

**Authors:**

**Author SID Author Name Signature**

IT21377730 Ameen F.A

IT21226632 Weerasinghe W.W.A.B.M

IT21063596 Fernando W.S.K

IT21190148 Jayathilaka W.I.S

IT21257018 Rathnayaka K.P.S.K

IT21027574 Samarathunga S.P.V.

IT21375446 Munasinghe I.M.S.D.T

IT21316722 Jayathilaka A.G.V.D.T

Date:………………………………….

# Abstract

This project aimed to develop a comprehensive cleaning service management system web application using the Agile Scrum methodology and MERN stack technologies. The objective was to eliminate the need for manual cleaning service management, which was both time-consuming and inefficient. The existing manual process involved managing physical files, manually scheduling each cleaning session, and handling customer details. Our management system encompasses various essential features, including customer and feedback management, employee management, quotation management, schedule management, payment management, payroll management, inventory management, and attendance management. By automating these tasks, we aimed to streamline operations for the Yamini cleaning service company, optimizing output while minimizing input resources.

Throughout the development process, we adhered to the Agile Scrum methodology, which facilitated the creation of a user-friendly web application tailored to the company's specific needs. Leveraging the MERN stack technologies (MongoDB, Express, React, and Node.js). we built a robust and scalable application. The user interface was thoughtfully designed to prioritize simplicity, intuitiveness, and seamless navigation.

In conclusion, the web application we developed successfully overcame the challenges associated with manual cleaning service management, offering a dependable and efficient solution for business owners. The incorporation of various features and the user-friendly interface catered to both the owners and customers, ensuring a satisfying experience for all users. This project serves as a testament to the effectiveness of Agile Scrum and the MERN stack in delivering a high-quality web application that precisely meets customer requirements.

# Acknowledgement

We would like to take this opportunity to express our sincere gratitude to our lecturers, instructors, and director of the Yamini cleaning service for providing us with valuable and constructive suggestions as well as motivation during the process of planning and developing this project. We would like to express our gratitude to all the other academic lecturers who contributed their expertise and advice to the development of the web application. Our gratitude and appreciation are also extended to our coworkers in the creation of the project as well as the individuals who have voluntarily assisted us with their abilities.

# Table of contents

Contents

[Declaration 2](#_Toc135294683)

[Abstract 3](#_Toc135294684)

[Acknowledgement 4](#_Toc135294685)

[Table of contents 5](#_Toc135294686)

[List of Tables 7](#_Toc135294687)

[List of Figures 8](#_Toc135294688)

[List of abbreviations 9](#_Toc135294689)

[1. Chapter 01 – Introduction 10](#_Toc135294690)

[1.1 Background 10](#_Toc135294691)

[1.2 Problem and Motivation 11](#_Toc135294692)

[1.3 Aim and Objectives 12](#_Toc135294693)

[1.4 Literature Review 13](#_Toc135294694)

[1.5 Solution Overview 14](#_Toc135294695)

[1.6 Methodology 16](#_Toc135294696)

[1.7 Structure of the report 17](#_Toc135294697)

[1.8 GitHub Repository Link 17](#_Toc135294698)

[2. Chapter 02 – Requirements 18](#_Toc135294699)

[2.1 Stakeholder Analysis 18](#_Toc135294700)

[2.2 Requirement Analysis 20](#_Toc135294701)

[2.2.1 Functional requirements 20](#_Toc135294702)

[2.2.2 Non-Functional Requirements 21](#_Toc135294703)

[2.2.3 Technical requirements 22](#_Toc135294704)

[2.3 Requirements Modelling 23](#_Toc135294705)

[2.3.1 User stories 23](#_Toc135294706)

[2.3.2 Use Case Diagram 27](#_Toc135294707)

[3. Chapter 03 – Design and Development 28](#_Toc135294708)

[3.1 System Architecture Diagram 28](#_Toc135294709)

[3.2 DataBase Diagram 29](#_Toc135294710)

[3.3 Component Diagrams 30](#_Toc135294711)

[3.4 Workflow diagrams 31](#_Toc135294712)

[4. Chapter 04 – Testing 32](#_Toc135294713)

[4.1 Unit testing 32](#_Toc135294714)

[4.2 Integration testing 48](#_Toc135294715)

[4.3 User acceptance testing 48](#_Toc135294716)

[5. Chapter 05 - Evaluation and Conclusion 50](#_Toc135294717)

[5.1 Evaluation 50](#_Toc135294718)

[5.2 Conclusion 51](#_Toc135294719)

[References 52](#_Toc135294720)

[Individual contribution 53](#_Toc135294721)

[Appendices 54](#_Toc135294722)

# List of Tables

[Figure 2.1 Onion Diagram of the stakeholders 19](https://mysliit-my.sharepoint.com/personal/it21228612_my_sliit_lk/Documents/2nd%20yr%202nd%20sem/ITP%20stuff/phase%203/final%20report%20v3.docx#_Toc135093826)

[Figure 2.2 Use case diagram of the system. 24](https://mysliit-my.sharepoint.com/personal/it21228612_my_sliit_lk/Documents/2nd%20yr%202nd%20sem/ITP%20stuff/phase%203/final%20report%20v3.docx#_Toc135093827)

[Figure 3.1 System Architecture Diagram 25](https://mysliit-my.sharepoint.com/personal/it21228612_my_sliit_lk/Documents/2nd%20yr%202nd%20sem/ITP%20stuff/phase%203/final%20report%20v3.docx#_Toc135093828)

[Figure 3.2 DataBase Diagram of the system 26](https://mysliit-my.sharepoint.com/personal/it21228612_my_sliit_lk/Documents/2nd%20yr%202nd%20sem/ITP%20stuff/phase%203/final%20report%20v3.docx#_Toc135093829)

[Table 4.1 Test case 1 29](#_Toc135093830)

[Table 4.2 Test Case 1 results 29](#_Toc135093831)

# List of Figures

# List of abbreviations

# Chapter 01 – Introduction

## Background

Based in Kalubovila, Sri Lanka, Yamini Cleaning Service is a respectable and well-known provider of cleaning services. Since its founding in 2017, the business has provided outstanding cleaning services to its clients, building a strong reputation in the field and rapidly expanding its clientele to serve 200–300 clients.

To meet a variety of needs, the organization provides a wide range of cleaning services. Their knowledge of office cleaning ensures that firms have a tidy and hygienic working environment. Additionally, they have a focus on apartment cleaning and offer comprehensive and effective cleaning services to apartment buildings. To maintain a spotless and cozy living space for homeowners, home cleaning services are also offered. Additionally, their floor cleaning services include skilled methods to improve the hygienic conditions and aesthetics of many kinds of flooring.

The management of the Yamini cleaning service has been understanding the potential of technology to revolutionize the cleaning service industry, Yamini Cleaning Service currently searching for new IT solutions that could automate and optimize their manual processes. By leveraging the power of digital tools, they aimed to enhance operational efficiency, reduce errors, and provide a seamless experience for both their clients and employees.

.

## Problem and Motivation

Yamini Cleaning Service, a renowned cleaning service company, was grappling with several challenges in its management system, primarily due to outdated manual methods. The first obstacle was the significant time consumption involved in performing administrative tasks such as customer registration, managing their details, and generating reports manually using physical documents. This inefficiency resulted in time wastage and decreased overall productivity.

The second challenge arose from the reliance on physical storage systems, which posed a risk of data loss due to factors like misplacement, damage, or theft. This vulnerability compromised the company's ability to access crucial information, disrupted its operations, and undermined customer trust.

The third issue revolved around the cost of maintaining a physical storage system. Expenses associated with stationery, filing equipment, and manual record-keeping added up, straining the company's financial resources unnecessarily.

Another challenge was encountered during scheduling, particularly during peak hours. Manual scheduling led to errors, overlapping appointments, and delays in service delivery, causing frustration among both clients and employees.

The fifth issue stemmed from the lack of an efficient attendance tracking system. The manual process of recording and managing attendance was prone to errors, making it difficult to accurately track employee work hours and ensure fair compensation.

Lastly, communication barriers between the company and clients were prevalent due to the absence of an effective notification system. This resulted in missed messages, delayed responses, and reduced overall efficiency in addressing customer needs and concerns.

Addressing these challenges became imperative for Sparkling Clean. Implementing a systematic solution would save time, enhance data management, reduce costs, improve scheduling accuracy, ensure reliable attendance tracking, and enhance communication with clients. These improvements would ultimately boost overall efficiency, and customer satisfaction, and establish a competitive edge in the cleaning service industry.

By developing a web application specifically tailored to the needs of Yamini Cleaning Service, the company aimed to revolutionize its cleaning service management system. Through automation and digitization, they sought to streamline operations, increase productivity, and provide a seamless experience for both employees and clients.

## Aim and Objectives

Aim:

The aim of this project is to develop a web application for managing cleaning service operations and customer data, replacing the current manual system used by Yamini cleaning service. The proposed system will provide an efficient, user-friendly, and reliable solution to the problems faced by the administration, clients, and labors.

Objectives:

* To design and develop a web application using the MERN stack, implementing the Agile Scrum methodology for project management.
* To create a user management module that allows the administrators to manage customer data.
* To develop a schedule management system that handles all the schedules for clients and labors.
* To create a quotation management system to manage all the quotations in the web application and handle all the customer bookings and track the location of the client.
* To develop an inventory management system that efficiently manages all the inventory-related tasks, including tracking stock levels, and generating real time reports.
* To create an employee management system that handles all the employee-related functions, including employee profiles and reports.
* To develop a payment management system that handles all payment-related processes, including invoicing, online payment integration, generating receipts, managing payment records, and ensuring secure and streamlined transactions for improved financial management.
* To create a payroll management system that automates and streamlines the entire payroll process, including salary calculation, tax calculations, and generating accurate and timely payroll reports for efficient and error-free payroll management.
* To develop an employee attendance system based on QR reader technology, enabling employees to easily clock in and out by scanning unique QR codes assigned to everyone. The system will accurately track attendance records, provide real-time attendance data, automate attendance reports, and enhance efficiency in managing employee attendance for improved workforce management.

By achieving these objectives, we aim to develop a comprehensive gym management system web application that fulfills the requirements and needs of Dream Fitness, providing an efficient and effective solution to their current manual system.

## Literature Review

## Solution Overview

The proposed solution is a web-based cleaning management system that provides an efficient and user-friendly platform for gym owners to manage their business operations. The solution is developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, a popular and robust technology stack for building web applications.

The system comprises of several functional modules, those are,

1. **User management**:
2. **Quotation Management:**

The Quotation Manager function is an essential component of a website or system that handles quotations. It enables the addition, display, modification, and deletion of quotations, catering to the needs of both the quotation manager and the customers. With this function, the quotation manager can easily add new quotations to the site, ensuring a constantly updated collection. The system efficiently displays the available quotations to the customers, allowing them to search and view specific quotes of interest. Additionally, the quotation manager has the authority to update or delete existing quotations, ensuring accurate and relevant information. The system also generates a comprehensive quotation report, providing valuable insights and analysis. Furthermore, the function allows the admin to track the location of customers, enhancing customer service and support. Lastly, customers can conveniently obtain quotes from the site, facilitating smooth and efficient interactions.

1. **Payment Management:**

As customer side, after confirming the services, customer can complete the final step which is payment. By login into the payment portal, the customer should fill in a detail form and confirm the payment.

Then the customer directs to the page where filled data can be updated if need. Otherwise, the customer can click on “OK” button and navigate to the home page. If a customer wants to cancel the service, can click on “CANCEL” button.

As the admin side, the finance department is responsible for managing all the payments related to the company.

The finance executive can view and accept or decline the payments made by the customer and is able to view, add, update, and delete payments which are for departments.

If finance department or general manager need to get a summary of the payments, anyone who can access the admin dashboard, is able to click on “Quick Summery” button without download the monthly payment report.

At the end of every month, the finance executive downloads the monthly payment report which includes all the information during the month.

1. **Inventory Management:**

Main user to inventory management system is the stock manager. The stock manager handles all stock details. This helps to new current stock levels to make decision.

Stock manager can add a new equipment and chemical providing the Item Id, Name, Initial stock quantity etc. IF an already added the system, the stock manager can increase the quantity of the item by a given amount. IF the needs to update details of a particular item stock manager clicking the update button and change the details. (Add and delete) Every time a particular item is consumed, the current stock quantity of that item will be deducted.

When releasing item update will be to database following Item Id. The stock manager can see a notification if a particular item is running low so that they can reorder according to the need. Supplier name, Id and contact number stored in the database and handle by the stock manager. Finally, the stock manager can generate reports about the stock levels of the current time**.**

1. **Employee Management:**

An employee management system in an office is a software platform designed to streamline and optimize various aspects of managing employees within an organization. It provides a centralized hub where employee information, such as personal details, job roles, and performance data, can be stored and accessed easily by authorized personnel. This system simplifies administrative tasks related to employee management, such as attendance tracking, leave management, and scheduling. It allows supervisors to efficiently assign tasks, monitor employee performance, and generate reports for performance evaluations. Communication between management and employees is enhanced through the system, enabling seamless collaboration and feedback exchange. It also facilitates effective resource allocation by providing insights into employee availability, skill sets, and workloads. With features like leave management, the system automates the process of requesting and approving time off, reducing paperwork and ensuring accurate record-keeping. This promotes transparency and fairness in managing employee absences.

Ultimately, an employee management system improves operational efficiency, enhances communication, and enables data-driven decision-making, leading to a more organized and productive work environment. It saves time and effort for both employees and management, allowing them to focus on core business tasks.

1. **Schedule Management:**
2. **Attendance Management:**
3. **Payroll Management:**

## Methodology

Due to its capacity to promote iterative development, team cooperation, and early issue discovery, the Agile Scrum methodology was chosen for the development of the proposed Cleaning Service Management System web application. Four unique phases, namely Planning, Design, Development, and Testing, were used to divide the project into these sections.

Planning Phase:

After conducting stakeholder interviews to determine the project's scope and needs, a Product Backlog with a list of all the features and functionalities needed for the system was established. Following the ranking of the features according to significance, a Sprint Backlog was created for each of the two-week-long sprints. Each sprint had a sprint planning meeting before it began.

Design phase

Using Figma, a collaborative design tool, the proposed system's wireframes and prototypes were created during the Design phase. The project team completed the database schema during this phase, and the MERN stack (MongoDB, Express, React, and Node.js) was chosen for the system's development.

Development Phase:

The project team focused on developing the functional modules of the proposed system, which included User Management, Quotation Management, Schedule Management, Employee Management, Payment Management, Inventory Management, Payroll Management, and Attendance Management. VS-Code was used as the IDE, and GitHub was used as the code repository.

Testing Phase:

The Testing phase involved conducting unit testing, integration testing, and user acceptance testing. Unit testing and Integration testing was done manually using the black-box method. Postman was also used for testing the back-end routing.

In conclusion, the suggested Cleaning Service Management System web application was created using the Agile Scrum approach. Planning, Design, Development, Testing, and Deployment were the five stages that made up the project. Modern software and tools, like Figma, the MERN stack, GitHub, and Postman, were used in a collaborative manner. The outcome is a user-friendly, high-quality web application that satisfies the specifications.

## Structure of the report

The structure of the report is organized in a logical sequence to provide a clear flow of information. The rest of the report’s sections are as follows:

* **Chapter 02 -** Requirements: This chapter focuses on the requirements analysis, including stakeholder analysis, detailed functional and non-functional requirements, and technical requirements. It also includes requirements modeling techniques used.
* **Chapter 03 -** Design and Development: Chapter 03 covers the design and development aspects of the project, including system architecture diagrams, database diagrams, component diagrams, and workflow diagrams. It provides insights into how the system is designed and developed.
* **Chapter 04 -** Testing: This chapter discusses the testing phase of the project, including unit testing, integration testing, and user acceptance testing. It includes test cases, results, and an evaluation of the system's performance.
* **Chapter 05 -** Evaluation and Conclusion: This chapter presents the evaluation of the project, which will be based on test results, user feedback, and expert opinions. It concludes with a summary of the project's achievements, addressing how the objectives were met and the aim is achieved.

The structure of the report is designed to provide a comprehensive understanding of the project, starting from the introduction, moving through the requirements, design, and testing phases, and concluding with an evaluation and summary of the project's outcomes.

## GitHub Repository Link

# Chapter 02 – Requirements

## Stakeholder Analysis

To identify and comprehend the important people and organizations with a stake in the Cleaning Service Management System project, a stakeholder analysis was carried out. Their roles, interests, and possible influence on the project's success are identified thanks to this investigation. The following parties involved have been recognized,

**Clients:** Clients are individuals or organizations who seek the services of cleaning service management. They are the primary stakeholders as they engage the cleaning service to fulfill their cleaning needs. Clients may include residential homeowners, commercial businesses, educational institutions, healthcare facilities, and more. Their expectations and satisfaction are crucial for the success and growth of the cleaning service.

**Manager:** Managers are essential to the administration of cleaning services. They oversee organizing and supervising the daily activities of the cleaning service. They deal with a range of duties, including interacting with clients, planning cleaning appointments, supervising personnel, guaranteeing quality control, and resolving any problems or difficulties that may come up.

**Instructors:** Instructors are stakeholders who provide training, and guidance to the cleaning service staff, and manage all the client’s bookings. They are essential in ensuring that the person has the abilities and information needed to carry out their cleaning responsibilities successfully.

**Labors:** Labor refers to the workforce employed by the cleaning service management to perform the actual cleaning tasks. They could be the janitors, cleaners, housekeepers, or another group of people in charge of carrying out the cleaning services. The labor stakeholders are providing the cleaning services and interacting with clients right in front of them. Customer satisfaction and the cleaning service's reputation are significantly impacted by their performance, professionalism, and adherence to the cleaning standards.

**Banks:** Banks help and act as stakeholders in your cleaning business by offering financial services. They are essential in conducting financial transactions, monitoring accounts, and giving your cleaning service management credit or loans.

By analyzing the stakeholders and understanding their needs, expectations, and potential impact on the project, the development team can make informed decisions, prioritize requirements, and ensure that the system meets the diverse needs of all stakeholders.

|  |  |
| --- | --- |
| Internal | External |
| Managers | Clients |
| labors | Banks |
| Instructors |  |
|  |  |

## Requirement Analysis

### Functional requirements

Each management system has different functional requirements from different users. Functional Requirements of the system are analyzed below.

**User management:**

**Quotation Management:**

1. Quotation Manager needs to add new quotations to the site.
2. The system must display the available quotations to the customers.
3. System must allow quotation manager to update, delete available quotations.
4. Users must be allowed to search and view quotations.
5. The system must generate quotation report.
6. Admin must able to track the location of the customer
7. Customer must able to get quote from the site.
8. Manager must be allowed to update, delete quotes

**Payment Management:**

1. Customer add card details
2. Customer update card details
3. Customer delete card details
4. Add payment to department details
5. Update payment from department details
6. Delete payment from department details
7. Accept payment in customer payment details
8. Decline payment in customer payment details
9. Search payment details
10. Generate monthly payment report

**Employee Management:**

1. The Employee Manager needs to add new employees to the system.
2. The system must display the available employees' profiles and information to authorized personnel.
3. The system must allow the Employee Manager to update and delete employee records as necessary.
4. Users, such as supervisors or managers, must be allowed to search and view employee profiles.
5. The system must generate employee-related reports, such as performance evaluations, attendance records, and training history.
6. Admin must be able to track the location of employees, especially for field-based or remote workers.
7. Employees must be able to access the system to update their personal information, view company policies, and submit requests.
8. Managers must be allowed to update and delete employee records, such as job roles, performance evaluations, and leave requests**.**

**Schedule Management:**

**Inventory Management:**

**1.** **Maintain Stock- add**

**delete**

**retrieve**

**update**

**2.** **Search for stock manager in a tabular way.**

**3.** **Generate Notification-When a good near to out of stock, system will generate a warning notification from the system to the stock manager.**

**4.** **Generate purchase Order-When the stock level reaches the given minimum level the purchase order will be auto generated and store the tabular way.**

**5.** **Send purchase order to supplier-Generated purchase order containing list of the items require in the inventory. The prepared purchase order is sent the supplier.**

**6.** **Generate goods received note and stock report-When invoice is received into the system the good receive note (GRN) is generate by stock manager.**

**7.** **The stock manager can have a generate inventory report given the time period for the levels to make decision.**

**8. The reports can check payment manager and senior manager.**

**Attendance Management:**

**Payroll Management:**

.

### Non-Functional Requirements

**Performance:**

1. Email/phone number validation OTPs must be sent within 1 minute.
2. Email notifications must be sent under 2 minutes.
3. All the pages in the system should be loaded within 2 seconds with all images and videos of the page.
4. The system must be capable of getting any video, image formats and gifs as inputs.
5. Any changes in the system database must be updated in the whole system in under 2 seconds.

**Scalability:**

1. The system should be able to handle at least 1000 registered users at a time.
2. The system should be able to handle 100,00 user visits at the same time.

**Responsive Design:**

1. Ensure that the web application is mobile-friendly and responsive across various devices and screen sizes.

**Availability:**

1. The system should be available 24x7.

**Security:**

1. Admin UI must only be accessible to site admin, manager, and instructors.
2. Sensitive user data must be encrypted before storing them in the database.

**Maintenance and Support:**

1. Plan for regular maintenance and updates to address bug fixes, security patches, and feature enhancements.

**Usability:**

1. The system must be user friendly.
2. All the main functions should be able to be accessed from the home page.

### Technical requirements

1. Users will need a stable internet connection.
2. For optimal performance, the system should be accessed through up-to-date versions of web browsers such as Google Chrome, Mozilla Firefox, or Safari.
3. The system can be accessed on desktop or laptop computers running the latest operating systems.
4. Mobile devices running Android 5.0 or later and iOS 11 or later are supported for accessing the system on the go.
5. During the registration process, users will be required to provide and verify their email address. They will also receive important system notifications via email.

## Requirements Modelling

### User stories

User Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| User Authentication | 111 | As a user, I want to be able to enter my login credentials and be authenticated so that I can access the system.  As an administrator, I want to be able to configure authentication settings for the system so that I can ensure that only authorized users can access the system. |
|  |  |  |
|  |  |  |
| User Authorization | 112 | As a user, I want to be able to access certain resources and perform specific actions within the system based on my role or permissions.  As an administrator, I want to be able to manage user roles and permissions so that I can control who has access to what resources and actions. |
| User Provisioning | 113 | As an administrator, I want to be able to create new user accounts and assign appropriate roles and permissions. |
| User Monitoring | 114 | As an administrator, I want to be able to monitor user activity in the system so that I can detect and respond to potential security threats or compliance violations.  As an administrator, I want to be able to implement automated monitoring tools that can detect and alert me to potential security threats or compliance violations. |
| User Deprovisioning | 115 | As an administrator, I want to be able to deactivate or delete user accounts when users no longer need access to the system.  As an administrator, I want to be able to automate the user deprovisioning process so that deactivated or deleted accounts are handled consistently and accurately. |
| Search Customers Details | 116 | As a customer manager, I want to be able to search for customers using their CID so that I can view their details and history. |

Quotation Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Log in to the system | 511 | “As the quotation manager, I need to log into the system to complete my service” |
| 512 | “As the inspector, I want to log  into the system, so that I can  Complete my services.” |
| 513 | “As the customer, I want to log  into the system, so that I can  get my services.” |
| Create new quotations | 514 | “As a quotation manager, I need to create new quotations for the system” |
| Update/delete available quotations | 515 | “As a quotation manager, I need to update available quotations from the website” |
|  | 516 | “As a quotation manager, I need to delete available quotations from the website” |
| Search and view quotations | 517 | “As the quotation manager, I need to search and view available quotations from the website.” |
|  | 518 | “As the Customer, I want to search and view all the available quotations on the website.” |
| Get a quote | 519 | “As the Customer, I want to get a quote by filling in the requirement details.” |
| 520 | “As the Customer, I want to get a quote by purchasing fix quota package .” |
| Track the customer location and get the requirements | 521 | “As the Inspector I need to track the location of the customer if the customer hires me to fill out the requirement form.” |
|  |  |  |
| Update/delete requirement details. | 522 | “As the Inspector, I need to update requirement details when customers request to change the details.” |
|  | 523 | “As the Inspector, I need to delete requirement details when customers request to delete the details.” |
| Generatequotationreport. | 524 | “As the quotation manager, I need to generate reports according to the service type and  date, so that I can manage all the quotation details properly.” |

Schedule Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Log in to the system as a schedule manager | 411 | “As a schedule manager, I want to login to the system so I can schedule daily orders.” |
| Schedule daily orders | 412 | “As a schedule manager, I want to schedule daily orders so I can manage them easily.” |
| View scheduled orders as a schedule manager | 413 | “As a schedule manager, I want to see daily orders so I can handle them daily.” |
| 414 | “As a schedule manager, I want to search order details so I can find particular order easily.” |
| Update scheduled orders | 415 | “As a schedule manager, I want to I want to update scheduled orders so I can manage them correctly.” |
| Delete scheduled orders | 416 | “As a schedule manager, I want to delete schedule orders so I can find completed orders.” |
| Generate monthly order report | 417 | “As a schedule manager, I want to generate monthly order report so I can find progress of the company.” |

Employee Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Login to the system as the admin | 711 | “As the Employee Manager, I want to login the system to maintain Employee details. So that I can add new employee details.” |
| Insert, delete, update of employee details. | 712 | “As the system Admin, I want to add, update or delete employee details to the system, so that I will be able to see all details.” |
| Prepare employee salary details | 713 | “As the employee Manager, I want to prepare salary details of employees confidentially, so that I can calculate salary expenses.” |
|  |  |  |
| View attendance details of all employees | 714 | “As the employee Manager, I want to view attendance details of all employees, so that I can track the attendance” |
| Employee leave details | 715 | “As the employee Manager, I want to accept or ignore leave requested by employee, so that I can manage leave requests” |
| Employee login their own profile | 716 | “As the employee can be able to see their own profile, so that they can manage their details properly.” |
| Employee requested leave | 717 | “As the employee I can request a leave, so that I can get a leave easily” |
|  |  |  |
| Daily attendance of employee | 718 | “As an employee insert the date, time and their EMP ID number and submit. So that manager can view those.” |

Payment management

|  |  |  |
| --- | --- | --- |
| Feature | User Story ID | User Story |
| Monthly profit and loss calculation | 611 | “As a Finance executive, I want to get income and expenditure, so that I can calculate profit and loss of the month” |
| Generate Reports | 612 | “As a Finance executive, I want to generate reports by considering the profits and losses of the company during a month so that I can analysis monthly payments easily” |
| View History | 613 | “As a Finance manager, I want to view the history of works done in payment management, so that I can search for payments of the company” |
| Accept/ Decline | 614 | “As a Finance executive, I want to accept / decline the payments done by the customer, so that I can complete the transaction” |
| Complete payment | 615 | “As a customer, I want to complete the payment, so that I can get the service” |
| Add, delete, update of payment details. | 616 | “As the finance executive, I want to add, update or delete payment details to the system which are done to the departments, so that I will be able to see all details. |

Inventory Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Login to the website as a Stock manager. | 811 | As a Stock manage staff, I want a login system, so that I can restrict the access for unauthorized users.” |
|  |

|  |  |  |
| --- | --- | --- |
| Manage Inventory | 812 | “As a Stock manage staff, I want to enter details to the database add, update and delete items and view, so that I can maintain the stock.” |
|  |  |  |
|  |  |  |
| Manage Stock Level | 813 | “As a Stock manage staff, I want to track the availability of stocks, so that I can request deficiency items from suppliers at right time.” |
|  |  |  |
| Generate Reports | 814 | “As finance executive I want to view reports on different functions, so that I can to make decision about whether the company’s transactions are going on properly.” |
|  |  |  |
|  |
|  |  | “As a Senior manager, I want to generate and view reports on different functions, so that I can to make decisions for the company. |
| “As a Stock manage staff, I want to generate and view reports on different functions, so that I can so that I can get a better understanding of the store items. |
| Search and filter | 815 | “As a Stock manage staff, I  want to search and filter the  items in the stock, so that I  can improve the efficiency of  my work. |
|  |  | “As a Senior manager, I  want to search and filter the  items in the stock, so that I  can improve the efficiency of  my work.” |

Payroll Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Registering to the system as an employee. | 911 | “As a payroll department staff, I want to add employees so that I can manage their salary calculations for each employee.” |
| 912 | “As a payroll department staff, I want to update employees details so that I can update changes if any.” |
|  |  |  |
| 913 | “As a payroll department staff, I want to delete employees so that I can remove them from the list when they leave.” |
| Calculating basic salary. | 914 | “As a payroll department staff, I want to enter the basic salary of each employee so that I can calculate their net salary.” |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Calculating Taxes. | 917 | “As a payroll department staff, I want to enter the current tax rate of the country so that I can calculate their tax expenditure.” |
|  |  |  |
|  |  |  |
| Generating Pay Slips | 918 | “As a manager, I want to view pay slips of any employee when I enter their employee ID so that I can make decisions.”  “As a payroll department staff, I want to generate the pay slips of each employee so that I can print them at the month end.” |
|  |  |  |
| Generating Pay Sheet. | 919 | “As a manager, I want to view pay sheet of the employees so that I can make decisions.”  “As a payroll department staff, I want to generate the pay sheet of the employees so that I can list them all at the month end.” |
|  |  |  |
|  |  |  |
| Generate Salary expense report. | 920 | “As a manager, I want to view Salary Expense Report of the month so that I can make decisions.”  “As a payroll department staff, I want to generate the Salary expense report for each month so that I can send it to the manager.” |
| Generate PAYE Tax report. | 921 | “As a manager, I want to view PAYE Tax report of each month so that I can make decisions.” |

Attendance Management

|  |  |  |
| --- | --- | --- |
| **Feature** | **User Story ID** | **User Story** |
| Register to the system as a employee | 311 | “As an employee, He should register to the system, Then  He can enter the leave |
| Login to the website as an admin | 312 | “As an admin, I want to login to the system for visit the employee details” |
| Searching for an employee list read update and delete | 313 | “As an admin, I want to update the database, delete employee details and add employee details.” |
|  |  |  |
| Generate a report | 314 | “As a manager, I want a daily attendance report and monthly report” |
| Calculate working hours | 315 | “As a payroll manager, I want employees working hours for making salary sheet. |
| Employee enters the company | 316 | “As an employee when he enters to the company he should show his face” |
| Employee requested leave | 317 | “As the employee I can request a leave, so that I can get a leave easily.” |
|  |  |  |

### Use Case Diagram

Figure 2.1 Use case diagram of the system.

# Chapter 03 – Design and Development

## System Architecture Diagram

A diagram of a financial management

Description automatically generated with medium confidence

Figure 3.1System\_Architecture\_diagrm

## DataBase Diagram

## Component Diagrams

## Workflow diagrams

# Chapter 04 – Testing

## Unit testing

In this project, unit testing was conducted as a black-box testing. The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. In this, tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

A total of 16-unit tests were conducted, with 2 tests performed per functional module. Test cases and their results are listed below.

Table 4.1 Test case 1

|  |  |
| --- | --- |
| **Test case ID:** 001 | **Test Designed and executed by:**  Samarathunga.H.P.V.S |
| **Test title:** Add customer |
| **Test priority:** high | **Date:** 2023 – 05 - 17 |
| **Module name:** Customer Management |
| **Description:** This test case verifies the functionality of adding a new customer in the cleaning service application. | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to add customers. | |
| **Test steps:**   1. Open the cleaning service application. 2. Navigate to the "Customers" section. 3. Click on the "Add Customer" button. 4. Enter the customer's name, address, contact information, and any other required fields. 5. Optionally, fill in any additional fields or information. 6. Click on the "Save" or "Add" button to submit the customer details. 7. Verify that the customer is successfully added and a confirmation message is displayed. 8. Go back to the "Customers" section and search for the newly added customer using their name or other identifying details. 9. Confirm that the customer is displayed in the search results with the correct information | |

Table 4.2 Test Case 1 results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 001 | Customer Name:  Vinodi Saumya  Address:  saumya,kiriwawla,thorayaya.  Contact Information:  [vinodisaumya@gmail.com](mailto:vinodisaumya@gmail.com) | * Confirmation message: "Customer successfully added." * Customer displayed in search results with the entered details | * Confirmation message: "Customer successfully added." * Customer displayed in search results with the entered details | pass | The customer was successfully added and the confirmation message was displayed, indicating that the test case was successful. The search feature correctly retrieved the client with the saved customer details that were accurate. The "Add Customer" feature was successfully confirmed by the test scenario. |

|  |  |
| --- | --- |
| **Test case ID:** 002 | **Test Designed and executed by :**  Samarathunga.H.P.V.S |
| **Test title:** Delete customer |
| **Test priority:** high | **Date :** 2023 – 05 - 17 |
| **Module name:** Customer Management |
| **Description:** This test case verifies the functionality of deleting a customer in the cleaning service application | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to delete customers. There is an existing customer in the system. | |
| **Test steps:**   1. Open the cleaning service application. 2. Navigate to the "Customers" section. 3. Search for the customer to be deleted using their name or other identifying details. 4. Select the customer from the search results. 5. Click on the "Delete" button or option. 6. Confirm the deletion in the prompt or dialog box. 7. Verify that the customer is successfully deleted and a confirmation message is displayed. 8. Search for the deleted customer again. 9. Confirm that the customer is no longer displayed in the search results. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 002 | Customer Name: Vinodi Saumya | * Confirmation message: "Customer successfully deleted." * Customer no longer displayed in the search results. | * Confirmation message: "Customer successfully deleted." * Customer no longer displayed in the search results. | pass | The client was successfully erased without any problems, and the confirmation message was displayed, indicating that the test case was successful. The customer is no longer visible in the search results, and the search functionality accurately reflected the deletion. The "Delete Customer" feature was successfully confirmed by the test scenario. |

|  |  |
| --- | --- |
| **Test case ID:** 003 | **Test Designed and executed by:**  Weerasinghe W.W.A.B.M- IT21226632 |
| **Test title:** Add new quotation to the web site |
| **Test priority:** high | **Date:** 2023 – 05 - 09 |
| **Module name:** Quotation Management |
| **Description:** testing the create quotation. | |
| **Preconditions:** admin has logged into the system using admin credentials. | |
| **Test steps:**   1. Navigate to Quotation management dashboard. 2. Click the “Quotation” button. 3. Select “Add” button. 4. Input all the details about the quotation and click “Submit” button. 5. Check whether a new quotation add to the site | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 003 | Quotation Name: Quote1  Service Type:  Apartment Cleaning  Package Type: Premium  Advanced Options: Vacuum cleaning | New Quotation is added to the web site and displayed | New Quotation is added to the web site and displayed | pass | The test case successfully validates all the provided inputs. No issues or errors were encountered during the test. |

|  |  |
| --- | --- |
| **Test case ID:** 004 | **Test Designed and executed by:**  Weerasinghe W.W.A.B.M- IT21226632 |
| **Test title:** Get a Quote from the site |
| **Test priority:** high | **Date:** 2023 – 05 - 09 |
| **Module name:** Quotation Management |
| **Description:** testing the get a quote function. | |
| **Preconditions:** user has logged into the system and filled account details in user’s account | |
| **Test steps:**   1. Navigate to Home page. 2. Click the “Get Quote” button. 3. Fill in the required input values. 4. Select “Get” button. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 004 | Name: Saman Kumara  Email Address:  [saman@gmail.com](mailto:saman@gmail.com)  Service Type:  Home cleaning  Location:  Anuradhapura  Phone Number:  0439084342 | * Get a Quote successfully and received the notification. | * Get a Quote successfully and received the notification. | pass | The test case successfully validates all the provided inputs. No issues or errors were encountered during the test. |

|  |  |
| --- | --- |
| **Test case ID:** 005 | **Test Designed and executed by :**  Rathnayaka K.P.S.K |
| **Test title:** Add Schedule |
| **Test priority:** high | **Date :** 2023 – 05 - 17 |
| **Module name:** Schedule management |
| **Description:** This test case verifies the functionality of adding a new schedule in the cleaning service application. | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to add schedule. | |
| **Test steps:**   1. Open the application and navigate to the schedule management section. 2. click on the "Add Schedule" button. 3. Fill in the required fields, customer name, address, order type and phone number. 4. Provide valid and relevant data for any optional fields. 5. Click on the "Add" button to submit the schedule. 6. Verify that the system validates the input fields appropriately. 7. Confirm that the schedule is successfully saved to the database. 8. Check for any notifications or success messages indicating a successful schedule addition. 9. Ensure that the added schedule is displayed correctly in the schedule management section. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 005 | Customer Name: Sahan Kalhara  Customer Address: Kurunegala  Order Type: House  Customer Phone: 0714678234 | * Schedule successfully added to the system. * System displays a success message/notification. * . The added schedule appears correctly in the schedule management section. | * Schedule successfully added to the system. * System displays a success message/notification. * . The added schedule appears correctly in the schedule management section. | pass | The test case successfully validates the addition of a schedule with all the provided inputs. No issues or errors were encountered during the test execution. The added schedule appears as expected in the schedule management section. |

|  |  |
| --- | --- |
| **Test case ID:** 006 | **Test Designed and executed by:**  Rathnayaka K.P.S.K |
| **Test title:** Update Schedule |
| **Test priority:** high | **Date:** 2023 – 05 - 17 |
| **Module name:** Schedule management |
| **Description:** This test case verifies the functionality of update a schedule detail in the cleaning service application. | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to update schedule. | |
| **Test steps:**   1. Open the application and navigate to the schedule management section. 2. Select the schedule that want to update. 3. Click on the "Update" button to update the schedule details. 4. Verify that the schedule details are pre-filled with the existing data. 5. Make the necessary changes to the fields that want to update. 6. Click on the "Update" button to submit the changes. 7. Verify that the system provides appropriate validation for the updated fields. 8. Confirm that the schedule details are successfully updated in the database and reflected in the schedule management section. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 006 | Customer Name: Vihan Maneth  Customer Address: Kurunegala  Order Type: Office  Customer Phone: 0714678234 | * Schedule details are successfully updated in the system. * System displays a success message/notification. * The updated schedule appears correctly in the schedule management section. | * Schedule details are updated in the system. * A success message/notification is displayed. * The updated schedule is visible in the schedule management section. | pass | The test case successfully validates the update of a schedule with the provided inputs. No issues or errors were encountered during the test execution. The updated schedule details appear as expected in the schedule management section. |

|  |  |
| --- | --- |
| **Test case ID: 007** | **Test Designed and executed by :**  Munasinghe I.M.S.D.T |
| **Test title:** Delete Employee |
| **Test priority:** High | **Date:** 2023 – 05 - 17 |
| **Module name:** Employee Management |
| **Description:** This test case verifies the functionality of deleting a employee in the cleaning service application | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to delete employees. There is an existing employee in the system. | |
| **Test steps :**   1. Open the cleaning service application. 2. Navigate to the "Customers" section. 3. Search for the customer to be deleted using their name or other identifying details. 4. Select the customer from the search results. 5. Click on the "Delete" button or option. 6. Confirm the deletion in the prompt or dialog box. 7. Verify that the customer is successfully deleted, and a confirmation message is displayed. 8. Search for the deleted customer again. 9. Confirm that the customer is no longer displayed in the search results. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 007 | Employee Name:  Kamal | * Confirmation message: "Employee successfully deleted." * Employee no longer displayed in the search results. | * Confirmation message: "Employee successfully deleted." * Employee no longer displayed in the search results. | pass | The client was successfully erased without any problems, and the confirmation message was displayed, indicating that the test case was successful. The employee is no longer visible in the search results, and the search functionality accurately reflected the deletion. The "Delete Employee" feature was successfully confirmed by the test scenario |

|  |  |
| --- | --- |
| **Test case ID:** 008 | **Test Designed and executed by:**  Munasinghe I.M.S.D.T |
| **Test title:** Add Employee |
| **Test priority:** High | **Date:** 2023 – 05 - 17 |
| **Module name:** Employee Management |
| **Description:** This test case verifies the functionality of adding a new employee in the cleaning service application | |
| **Preconditions:** The admin is logged into the cleaning service application and has the necessary permissions to add employees. | |
| **Test steps:**   1. Open the cleaning service application. 2. Navigate to the “Employees" section. 3. Click on the "Add Employee" button. 4. Enter the employee's name, address, contact information, and any other required fields. 5. Optionally, fill in any additional fields or information. 6. Click on the "Save" or "Add" button to submit the employee details. 7. Verify that the employee is successfully added, and a confirmation message is displayed. 8. Go back to the "Employees" section and search for the newly added employee using their name or other identifying details. 9. Confirm that the employee is displayed in the search results with the correct information. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 008 | Name: Kamal  Age: 23  Address:2/23A, new Kandy Road, Malabe.  Department: Management  Telephone: 0716709878  Email: [qwe@gmail.com](mailto:qwe@gmail.com)  Job position: Cleaner | * Confirmation message: "Employee successfully added." * Employee displayed in search results with the entered details | * Confirmation message: "Employee * successfully added." * Employee displayed in search results with the entered details | pass | The employee was successfully added, and the confirmation message was displayed, indicating that the test case was successful. The search feature correctly retrieved the client with the saved employee details that were accurate. The "Add Employee" feature was successfully confirmed by the test scenario |

|  |  |
| --- | --- |
| **Test case ID:** 009 | **Test Designed and executed by:**  IT21063596 – Fernando W.S.K |
| **Test title:** Customer add card details to confirm the service |
| **Test priority:** High | **Date:** 2023 – 05 - 17 |
| **Module name:** Add card details |
| **Description:**  Enter required details such as card type, card number, card holder, CVV, and expiration | |
| **Preconditions:** Customerhas logged into the system | |
| **Test steps:**   1. Fill in the card details. 2. Click the Pay Now button. 3. Pop up “Payment is Processing” alert | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 009 | Card Type: Visa  Card Number: 12364598786012364  Card Holder: S.Fernando  CVV: 456  Expiration: 01-25 | Add card details to system. After that alert display “Successful added” | Add card details. to system. After that alert display “Successful added” | pass | Customer logged in to account and view Payment Portal. Fill the form and click Pay No button. Then details store in the database successfully |

|  |  |
| --- | --- |
| **Test case ID:** 010 | **Test Designed and executed by:**  IT21063596 – Fernando W.S.K |
| **Test title:** Finance Executive update the payment details done to the departments |
| **Test priority:** High | **Date:** 2023 – 05 - 17 |
| **Module name:** Update the payment details |
| **Description:**  Update required details such as department ID, department name, and amount | |
| **Preconditions:** Finance Executivehas logged into the system | |
| **Test steps:**   1. Click the Update button. 2. Update the relevant details. 3. Pop up “Department Updated” alert. 4. Click the OK button. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
| 010 | Department ID: pay001  Department Name: Payroll  Amount: 65000 | Update the payment details done to the departments.  After that alert display “Successful updated” | Update the payment details done to the departments.  After that alert display “Successful updated” | pass | Customer logged in to account and view Payment Portal. Fill the form and click Pay No button. Then details store in the database successfully |

**Inventory Management**

|  |  |
| --- | --- |
| **Test Case ID: 011** | **Test designed by:** Jayathilake W.I.S |
| **Test Title:** Test validations for item code | **Test designed day:** 05/05/2023 |
| **Test priority (High/Medium/Low):** Medium | **Test executed by:** Jayathilake W.I.S |
| **Module name:** Add new inventory item | **Test executed day:** 07/05/2023 |
| **Description:** Check whether the Item code is in correct number and letter quantity or not and check whether the Item code field is empty or not | |
| **Preconditions:** All the other fields should be filled correctly | |
| **Test steps:**   * Navigate to add new item page. * Enter the item code in the item code field. * Fill all other fields correctly. * Click Submit button. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result (Pass/Fail) |
| 011 | Item code: BR0021 | Display “item successfully added” message. | Display “item successfully added” message. | Pass |
| 011 | Item code: DP12 | Display “item code too short” message. | Display “item code too short”. message | Pass |
| 011 | Item code: | Display “Please fill out this field” message. | Display “Please fill out this field” message. | Pass |

|  |  |
| --- | --- |
| **Test Case ID :012** | **Test designed by:** Jayathilake W.I.S |
| **Test Title:** Test validations for Quantity | **Test designed day:** 05/05/2023 |
| **Test priority (High/Medium/Low):** Medium | **Test executed by:** Jayathilake W.I.S |
| **Module name:** Add item in item page, Update item page. | **Test executed day:** 06/05/2023 |
| **Description:** Check whether the Quantity is in correct format or not and check whether the Quantity field is empty or not | |
| **Preconditions:** All the other fields should be filled correctly | |
| **Test steps:**   * Navigate to Add New item page or Update item page. * Enter the Quantity in the Quantity field. * Fill all other fields correctly. * Click Submit button. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result (Pass/Fail) |
| **01**2 | Quantity: 34 | Display “Item successfully added” message. | Display “Item successfully added” message. | Pass |
| 012 | Quantity: Two | Display “Quantity should be a number” message. | Display “Quantity should be a number” message. | Pass |
| 012 | Quantity: | Display “Please fill out this field” message. | Display “Please fill out this field” message. | Pass |

|  |  |
| --- | --- |
| **Test Case ID:** 013 | **Test designed by:** Jayathilake W.I.S |
| **Test Title:** Test validations for Supplier Phone Number | **Test designed day:** 04/05/2023 |
| **Test priority (High/Medium/Low):** Medium | **Test executed by:** Jayathilake W.I.S |
| **Module name:** Add Supplier page, Update Supplier page | **Test executed day:** 05/05/2023 |
| **Description:** Check whether the Phone is in correct format or not and check whether the phone number field is empty or not | |
| **Preconditions:** All the other fields should be filled correctly | |
| **Test steps:**   * Navigate to Add New supplier details or Update supplier details page. * Enter the Phone number in the phone field. * Fill all other fields correctly. * Click Submit button. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result (Pass/Fail) |
| **01**3 | Phone Number: 0758967178 | Display “Supplier successfully added” message. | Display “Supplier successfully added” message. | Pass |
| **01**3 | Phone Number: 07289214704564 | Display “Supplier unsuccessfully added” message. | Display “Supplier unsuccessfully added” message. | Pass |
| **01**3 | Phone Number: Zero712345 | Display “Phone Number should be a number ” message. | Display “Phone Number should be a number” message. | Pass |
| **01**3 | Phone Number: | Display “Please fill out this field” message. | Display “Please fill out this field” message. | Pass |

|  |  |
| --- | --- |
| **Test Case ID:014** | **Test designed by:**  Jayathilake W.I.S |
| **Test Title:** Test validations for Email | **Test designed day:** 01/05/2023 |
| **Test priority (High/Medium/Low):** Medium | **Test executed by:** Jayathilake W.I.S |
| **Module name:** Add Supplier detail page, Update Supplier details page. | **Test executed day:** 05/05/2023 |
| **Description:** Check whether the Email is in correct format or not and check whether the Email field is empty or not | |
| **Preconditions:** All the other fields should be filled correctly | |
| **Test steps:**   * Navigate to Add New Supplier page or Update Supplier Details page. * Enter the Email in the Email field. * Fill all other fields correctly. * Click Submit button. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Inputs | Expected Output | Actual Output | Result (Pass/Fail) |
| 014 | Email: sewwandi@gmail.com | Display “Supplier successfully added” message. | Display “Supplier successfully added” message. | Pass |
| 014 | Email: sewwandi.com | Display “Invalid Email” message. | Display “Invalid Email” message. | Pass |
| 014 | Email: seww | Display “Invalid Email” message. | Display “Invalid Email” message. | Pass |
| 014 | Email: | Display “Please fill out this field” message. | Display “Please fill out this field” message. | Pass |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** | Date: |
| **Module name:** |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** | Date: |
| **Module name:** |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** | **Date:** |
| **Module name:** |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** medium | Date : |
| **Module name:** FAQ |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** | **Date:** |
| **Module name:** |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Test Designed and executed by:** |
| **Test title:** |
| **Test priority:** | **Date:** |
| **Module name:** |
| **Description:** | |
| **Preconditions:** | |
| **Test steps:** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Inputs** | **Expected Output** | **Actual Output** | **Result (pass/fail)** | **Comments** |
|  |  |  |  |  |  |

## Integration testing

The system underwent comprehensive integration testing to ensure seamless collaboration among its components. Following a top-down approach, testing initiated with higher-level modules before integrating and testing lower-level ones. The test cases were meticulously designed to verify error-free and bug-free interplay between system elements, aligning with the documented functional requirements. A test environment mirroring the production setup facilitated accurate evaluation. The integration testing phase comprised 16 test cases spanning all functional modules. Results indicated a commendable 98% success rate, with minor glitches promptly identified and resolved. In conclusion, the integration testing phase triumphed, affirming that system components function as intended, and all specified functional requirements have been met.

Overall, the integration testing phase was successful, and it was confirmed that the system components work together as expected and meet the functional requirements of the system.

## User acceptance testing

The User Acceptance Testing (UAT) phase of the Cleaning Service Management web application involved a group of selected participants, including cleaning service administrators, cleaners, and clients. The objective of UAT was to assess the system's usability and ensure it meets the requirements and expectations of the end-users.

During UAT, participants were assigned specific test scenarios and tasks to perform. They evaluated the system's functionality, user interface, and overall user experience. Feedback from the participants was gathered and analyzed to identify any issues or areas for improvement.

The UAT participants found the Cleaning Service Management system to be intuitive and user-friendly. They were able to navigate through the application effortlessly, performing tasks such as job scheduling, client management, and reporting without encountering significant issues.

The system's user interface received positive feedback for its clarity and ease of use. Participants appreciated the clear instructions and intuitive design, which facilitated their interactions with the system. The process of creating and managing cleaning schedules and assigning tasks was deemed efficient and straightforward. User management functionalities, such as adding and managing cleaners, were also well-received by the UAT participants.

Based on the UAT feedback, minor enhancements were identified, including improving the notification system and providing additional customization options. These suggestions will be taken into consideration for future updates to further enhance the user experience of the Cleaning Service Management system.

# Chapter 05 - Evaluation and Conclusion

## Evaluation

The Office and Domestic Cleaning Web Application underwent a comprehensive evaluation process, combining test results and user feedback to assess its performance and effectiveness. The evaluation involved various stages, including unit testing, integration testing, and User Acceptance Testing (UAT).

A series of unit tests were conducted to evaluate the individual functional modules of the system, namely user management, cleaning schedule management, service catalog, employee management, customer feedback, billing, and reporting. A total of 16-unit tests were performed, with each functional module subjected to two tests. The unit tests yielded a commendable success rate of 95%, reflecting the robustness and reliability of the implemented functionalities.

In addition to that, integration testing was carried out to validate the seamless integration of different system components. A total of 15 integration tests were executed, covering various aspects of the system's functionality. The integration tests demonstrated an impressive success rate of 98%, indicating the smooth coordination and compatibility of the system components.

Furthermore, during the UAT phase, a group of participants consisting of office administrators, cleaners, and customers were actively engaged in evaluating the system's usability and its alignment with their requirements. The participants were assigned specific test scenarios and tasks, allowing them to explore different functionalities and provide valuable feedback on the system's functionality, user interface, and overall user experience. The UAT feedback indicated a positive reception of the Office and Domestic Cleaning Web Application. Participants found the system intuitive and easy to navigate, enabling them to carry out tasks such as scheduling cleanings, managing employee assignments, providing feedback, and reviewing billing details without encountering significant issues. The user-friendly interface received praise for its clear instructions and intuitive design, facilitating smooth interactions and enhancing the overall user experience.

Based on the evaluation results, minor enhancements were identified to further improve the system. These enhancements primarily focused on refining the notification system, providing additional customization options, and addressing specific user feedback for optimizing the user experience.

In summary, the evaluation process, encompassing UAT, unit testing, and integration testing, confirmed the usability, functionality, and overall effectiveness of the Office and Domestic Cleaning Web Application. The positive feedback received from the UAT participants, along with the high success rates in both unit and integration testing, reinforced the successful implementation of the system. The evaluation outcomes serve as a testament to the system's capability to meet user requirements and provide a satisfactory experience.

## Conclusion

By utilizing the MERN stack and Agile Scrum methodology, we have designed and implemented a user-friendly web application for office and domestic cleaning management. This application addresses the specific needs and challenges faced by cleaning service providers in managing their operations and customer data.

The web application comprises various modules tailored to the cleaning industry, including user management, cleaning schedule management, service catalog, employee management, customer feedback, billing, and reporting. These modules provide a comprehensive solution for streamlining cleaning operations, optimizing resource allocation, and enhancing customer satisfaction.

The user management module allows administrators to efficiently handle customer data, track service requests, and assign cleaners to specific tasks. This centralized system ensures accurate and up-to-date customer records, facilitating smooth communication and task allocation.

The cleaning schedule management module offers a comprehensive view of scheduled cleanings, enabling administrators to assign tasks based on cleaner availability and location. By optimizing resource allocation, this module helps improve operational efficiency and ensures timely service delivery.

The service catalog module categorizes cleaning services based on type and scope, providing customers with a range of options to suit their specific needs. This feature simplifies the service selection process and allows customers to customize their cleaning requests according to their requirements.

The employee management module streamlines the management of cleaner profiles, availability, and skills. Administrators can efficiently assign tasks based on cleaner qualifications and availability, ensuring the right cleaner is assigned to the appropriate job and optimizing the utilization of available resources.

The customer feedback module enables customers to provide valuable input, complaints, and suggestions regarding the cleaning service. This feedback mechanism promotes continuous improvement, allowing service providers to address any concerns promptly and enhance customer satisfaction.

The billing module generates accurate invoices based on service utilization, simplifying the billing process for both administrators and customers. This ensures transparent and efficient payment procedures, contributing to a seamless financial workflow.

The reporting module provides valuable insights and analytics related to cleaning operations, enabling administrators to monitor key performance indicators, track service quality, and identify areas for improvement. These data-driven insights facilitate informed decision-making and help optimize overall cleaning operations.

In conclusion, the Office and Domestic Cleaning Web Application, developed using the MERN stack and Agile Scrum methodology, provides an efficient, user-friendly, and reliable solution for managing office and domestic cleaning operations. By streamlining processes, optimizing resource allocation, and enhancing customer satisfaction, this web application contributes to the overall improvement of cleaning services, promoting efficiency, and delivering a high-quality cleaning experience for customers.

# References

# Individual contribution

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **IT No** | **Individual contribution** |
|  |  |  |  |
|  |  |  |  |
|  | Fernando W.S.K. | IT21063596 | * Payment Management |
|  | Jayathilake W.I.S | IT21190148 | * Inventory Management |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Appendices