

WEB BASED MACHINERY RENTAL SYSTEM FOR T.A ENTERPRISES

GEARGRID-WHERE TOOLS TRANSFORM DREAMS

By
H.T.V. Fernando
IM/2020/090

A report submitted in partial fulfillment of the requirements for the degree of Bachelor of
Science Honors in Management and Information Technology (B.Sc. MIT)

Name of the supervisor : Dr.Yehemini Jayatissa

Department of Industrial
Management Faculty of Science
University of Kelaniya
Sri Lanka
2024

Declaration

I hereby certify that this project and all the artifacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any other degree program.

Full name of the student: Hettiyakandage Theekshana Vimukthi Fernando

Student No: IM/2020/090



Signature of the student:

Date: 2024/01/01

Name of the supervisor: Dr. Yehemini Jayatissa

Signature of the supervisor:

Date: 2024/06/11

Contents

Declaration	1
ACKNOWLEDGEMENT	5
ABSTRACT	6
1.INTRODUCTION.....	7
1.1 Description of the business organization and the business area chosen.....	8
1.2 Problem Definition	8
1.3 Aims and Objectives	9
1.4 Organization of the Report	9
2.System Analysis	11
2.1 Use Case diagram for the existing system.....	12
2.2 Activity Diagrams to describe the functionality of the Use-Cases	13
2.2.1 This figure illustrate the activity diagram to describe the functionality of how the internal process works.....	13
2.2.2 This figure illustrate the activity diagram to describe the functionality of how the process going on within owner and the supplier.	14
2.3 Requirement Catalogue	15
2.3.1 Functional Requirments	15
2.3.2 Nonfunctional requirements	17
3.SYSTEM DESIGN	19
3.1 Use case diagrams for the proposed system.....	19
3.1.1 Overall Use Case Diagram for the Proposed System	20
3.1.2 High level view of the Use case diagram: functionalities of the proposed system	21
3.2.3 Use case diagram for describing functionalities: Login use case.....	21
3.1.4 Use case diagram for describing functionalities: Handling Customer details use case.....	22
3.1.5 Use case diagram for describing functionalities: Handling Machine details use case	23
3.1.6 Use case diagram for describing functionalities: Handling Invoice details use case	24
3.1.7 Use case diagram for describing functionalities: Handling Reservations use case	25
3.1.8 Use case diagram for describing functionalities: Warehouse handler actions use case.....	26
3.1.9 Use case diagram for describing functionalities : Admin panel special actions use case.....	27
3.2 User Storis and Use Case diagram for Existing System	28
3.2.1 This is a simple graphical representation to distinguish the permissions (functionalities) of the main system actors clearly.....	36
3.3 Activity diagram for the proposed system	37
3.3.1Activity diagram for describing functionalities: Login	37
3.3.2Activity diagram for describing functionalities: Register customers.....	38
3.3.3Activity diagram for describing functionalities: Manage equipment.....	39
3.3.4 Activity diagram for describing functionalities: Manage users	40
3.3.5 Activity diagram for describing functionalities: Rent equipment.....	41
3.4 Entity relationship diagrams.....	42
3.5 Class diagram for proposed system	43
3.5.1 Entity-Class diagram	43
3.5.2 Interface-Class diagram.....	44
3.5.3 Controller-Class diagram	45

3.6 Sequence diagram.....	46
3.6.1 Sequence diagram for login	46
3.6.2 Customer register in the website to reserve equipment	47
3.6.3 Manage users and customers	48
3.6.4 Manage equipment (items).....	49
3.6.5 Manage Invoices	50
3.7 INTERFACES	51
3.7.1 Invoice page	51
3.7.2 Invoice page after data retrieve	51
3.7.3 Customer page	52
3.7.4 Customer form validations.....	52
3.7.5 New Customer form	53
.....	53
3.7.6 Equipment page	53
.....	53
3.7.7 Equipment Form.....	54
.....	54
3.7.8 Equipment report 2	54
3.7.9 Equipment report 3	55
3.7.10 Equipment report 4.....	55
.....	55
3.7.11 Equipment report 5.....	56
3.7.12 Invoice report 1	56
3.7.13 Invoice report 2	57
3.7.14 Customer report 1	57
3.7.15 Customer report 2	58
3.7.16 Add user page.....	58
3.7.17 Invoice report	59
4.SYSTEM DEVELOPMENT	60
4.1 Programming Languages	60
4.1.1 Hyper Text Markup Language (HTML).....	60
4.1.2 Cascading Style Sheets (CSS)	60
4.1.3 JavaScript (JS)	61
4.1.4 Structured Query Language (SQL)	61
4.2 Development Tools and Technologies	61
4.2.1 Visual Studio Code.....	61
4.2.2 Version Control System (Git)	61
4.2.3 Database Management System (MySQL)	62
4.3 Third-Party Components/Libraries and Frameworks	62
4.3.1 Bootstrap.....	62
4.3.2 cors.....	62
4.3.3 dayjs	62
4.3.4 dotenv	62
4.3.5 express	62
4.3.6 jsonwebtoken.....	63
4.3.7 mysql2	63
4.3.8 nodemon	63
4.3.9 chart.js.....	63
4.3.10 firebase.....	63
4.3.11 html2canvas	63
4.3.12 jspdf.....	63

4.3.13 notistack	63
4.3.14 MUI (Material-UI)	64
4.3.15 Axios	64
4.4 Summary	64
5. Testing	65
5.1 White Box Testing	65
5.1.1 Unit Testing	65
5.2 Black Box Testing	66
5.2.1 Functional Testing	66
5.2.2 Non-Functional Testing	68
5.3 Summary	69
6. IMPLEMENTATION	70
6.1 Installation.....	70
6.1.1 Server Requirements.....	70
6.1.2 Software Considerations	70
6.1.3 Hardware Considerations.....	71
6.1.4 Data Migration	71
6.1.5 Installation Guide	71
6.1.6 Training	73
6.2 Backups	73
6.3 Summary	74
7. EVALUATION AND CONCLUSION	75
7.1 Degree of Objectives Met.....	75
7.2 Usability, Accessibility, Reliability and Friendliness	76
7.3 User's Response	77
7.4 Limitations and Drawbacks	77
7.5 Future Modifications.....	78
7.6 Conclusion	79
8. REFERENCES.....	81

ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my project supervisor, Dr.Yehemini Jayatissa for her immense support and guidance given to me from the beginning of the project.

I thank all the staff members, especially the panel of judges who gave valuable inputs and feedback on the video presentation of the system I was proposing to do. Those comments helped greatly to identify points to be improved and to have a clear idea of the system scope.

I would also like to convey my thanks to Mr.Cinthaka Fernando, the owner of T.A Enterprises for providing me the required information and for all the support.

My sincere thanks go to my parents and my family. They were a great strength when collecting information for this project and gave me a lot of new ideas to think about.

Finally, I would like to thank all my friends for their true friendship and for providing honest support, feedback and encouragement during the preparation of this report.

ABSTRACT

In today's technologically advanced society, we rely heavily on the internet and cloud-based solutions to meet our daily needs. This reliance extends to various industries, including machinery rental businesses, where efficient management of resources is crucial. To address the operational challenges faced by T.A. Enterprises, we present GearGrid, a comprehensive machinery rental management system.

GearGrid is designed to streamline and automate the rental operations of T.A. Enterprises, enhancing efficiency and accuracy across multiple aspects of the business. By integrating advanced functionalities such as inventory management, customer interactions, and transaction processing, GearGrid aims to reduce manual errors and provide real-time visibility into critical business operations.

The system supports robust user management, allowing different roles (admin, cashier, warehouse handler) to access tailored functionalities, ensuring secure and efficient operations. With capabilities for tracking equipment status, managing customer histories, generating detailed reports, and facilitating real-time communication between warehouses, GearGrid significantly improves operational efficiency.

Furthermore, GearGrid prioritizes customer satisfaction by automating notifications and reminders, ensuring timely communication and adherence to rental agreements. The integration of multi-language support, cross-platform compatibility, and rapid invoice generation further enhances the user experience.

Overall, GearGrid is a powerful solution that addresses the specific needs of T.A. Enterprises, promoting seamless and efficient machinery rental management.

1.INTRODUCTION

In this chapter, the focus is on introducing the project that involves the creation of a web-based application for T.A Enterprises. The chapter will examine the business's characteristics, current processes, and problems. Additionally, the chapter will assess the goals and objectives of the proposed system, its scope and limitations, and how the components are organized.

Outline of the Chapter

1.1 Description of the business organization and the business area chosen

1.2 Problem Definition

1.3 Aims and Objectives

1.4 Organization of the report

1.1 Description of the business organization and the business area chosen

The chosen business organization, a machinery rental enterprise, operates under the name T.A Enterprises functioning within the expansive domain of machinery rental services. The business is meticulously organized, with distinct roles assigned to key personnel. The administrative responsibilities fall under the purview of the Admin, who oversees overall operations and manages administrative tasks. But in the current advanced business culture still machinery equipment renting didn't get a huge revolution in their business process because a lot of rental shops follow traditional techniques. Everyone who need these machines got disappointed because of these poor business strategies.

1.2 Problem Definition

Manual Information Storage: All business data, including transactions, customer details, and inventory management, is currently stored manually through traditional bookkeeping methods.

Difficulty in Information Retrieval: Retrieving specific information promptly is a significant challenge due to the manual paperwork involved, leading to delays in accessing critical data.

Customer ID Management Issues: The storage and management of customer identification cards present potential legal concerns. The current system may lack organization and security measures, raising compliance issues with legal and privacy standards.

Communication Challenges within Warehouses: The absence of an effective communication system hampers coordination between the main shop and its various warehouses. This limitation affects tasks such as equipment handovers and efficient inventory management.

These problems collectively indicate the need for a comprehensive system upgrade to automate processes, improve information retrieval efficiency, ensure legal compliance in customer data management, and establish seamless communication channels within the business.

1.3 Aims and Objectives

- Automate Information Management:

Aim: To transition from manual bookkeeping to an automated system for comprehensive information management.

Objectives:

Implement a digital database to store and manage transactions, customer details, and inventory.

Integrate automated data entry processes to reduce manual efforts and enhance accuracy.

- Streamline Information Retrieval:

Aim: To facilitate quick and efficient retrieval of information.

Objectives:

Implement a user-friendly interface for easy access to transaction records, customer details, and inventory information.

Introduce advanced search and filter functionalities to enhance information retrieval speed.

- Design a compact POS system integrated with the Project for Billing

Aim: Develop an integrated Point of Sale (POS) system to streamline the billing process.

Objectives:

Design a user-friendly POS interface that integrates seamlessly with the GearGrid system.

Enable real-time processing and updating of transactions and inventory during billing.

Ensure compatibility with various payment methods to facilitate customer convenience.

Implement receipt generation and printing capabilities for completed transactions.

- Enhance Customer ID Management:

Aim: To address legal and compliance concerns related to customer identification.

Objectives:

Establish a secure and organized system for storing and managing customer identification cards.

Implement encryption and access controls to ensure compliance with legal and privacy standards.

1.4 Organization of the Report

Chapter 1 – Introduction

In the first chapter, the focus is on introducing the business's characteristics, current processes, challenges, scope and limitations as well as the goals and objectives of the proposed system.

Chapter 2 - System Analysis

The primary objective of this chapter is to discuss the system's requirement analysis. It provides an overview of the existing system with the help of user descriptions, use cases, functional and non-functional requirements and activity diagrams

Chapter 3 – System Design

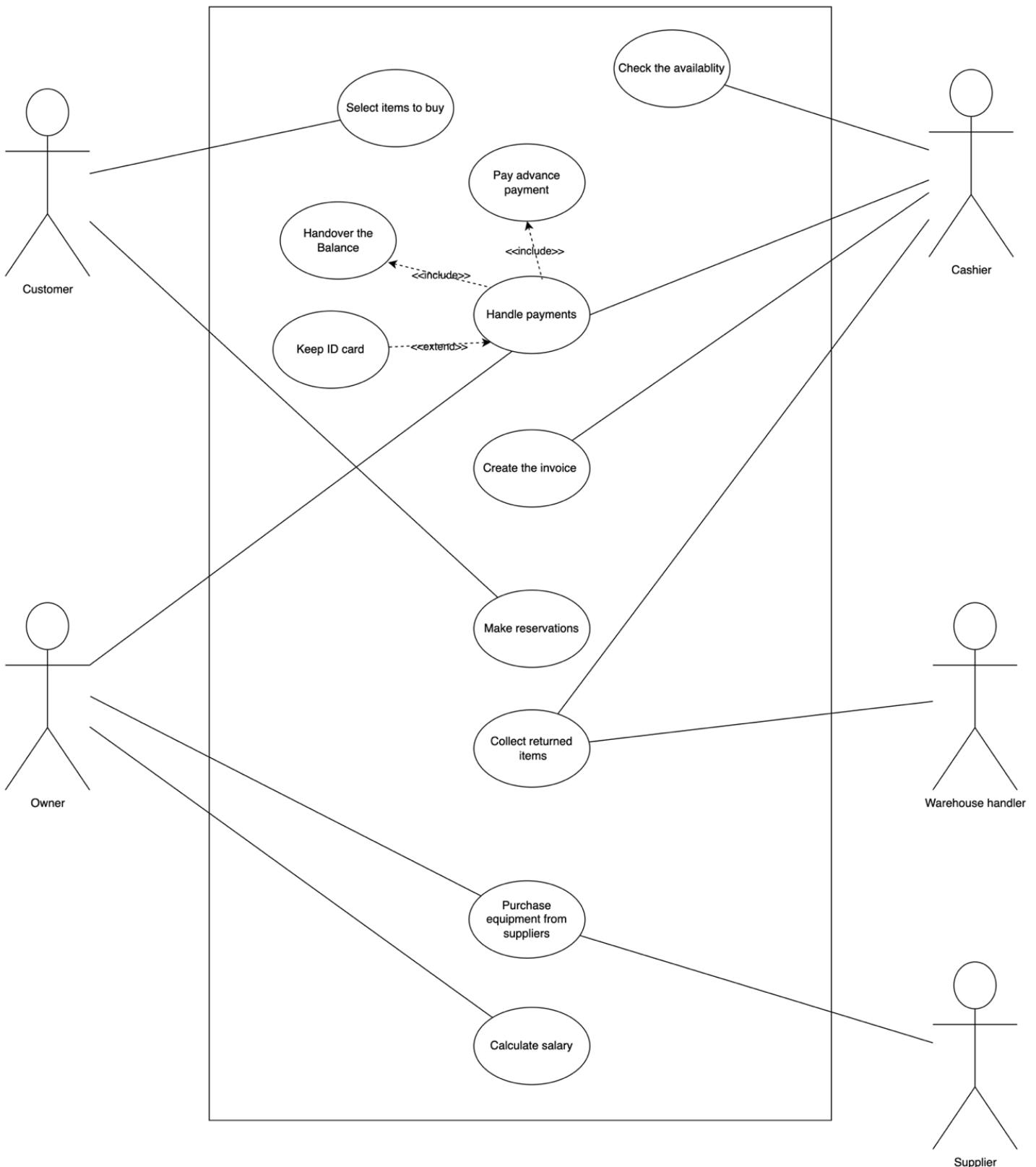
After clarifying the requirements and specifications of the system, the project will move on to the System Design stage. The purpose of this chapter is to provide a good understanding of the proposed system's behavior and relationships through diagrams. These diagrams will build

upon one another to explain the system's functionality, entities, and relationships. Towards the end of the chapter.

2. System Analysis

This chapter presents the documentation of the existing system, with the use of Object Oriented Systems Analysis and Design where the Use Case Diagram and activity diagrams are depicted. As a result of the analysis, Business System Options (BSOs) are developed. Thereafter the best BSO is selected after further analysis and justification is provided.

2.1 Use Case diagram for the existing system.



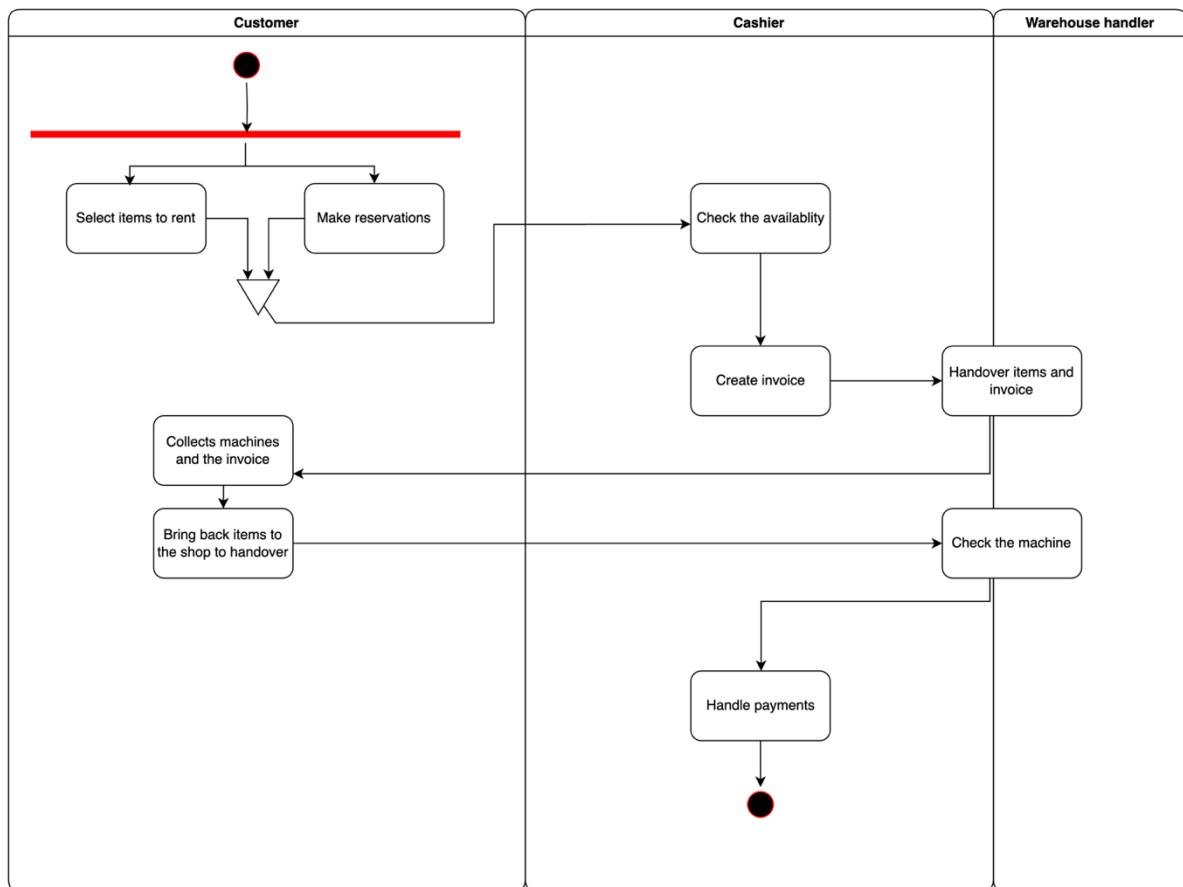
The functionalities of the existing processes are completely handled manually, so cannot identify an exact boundary for the manual process because it is open for all users involved in the process under different levels. Therefore, the consistency of the process is very poor.

Below use case diagram shows the overall scope of the existing process which completely carries out manually.

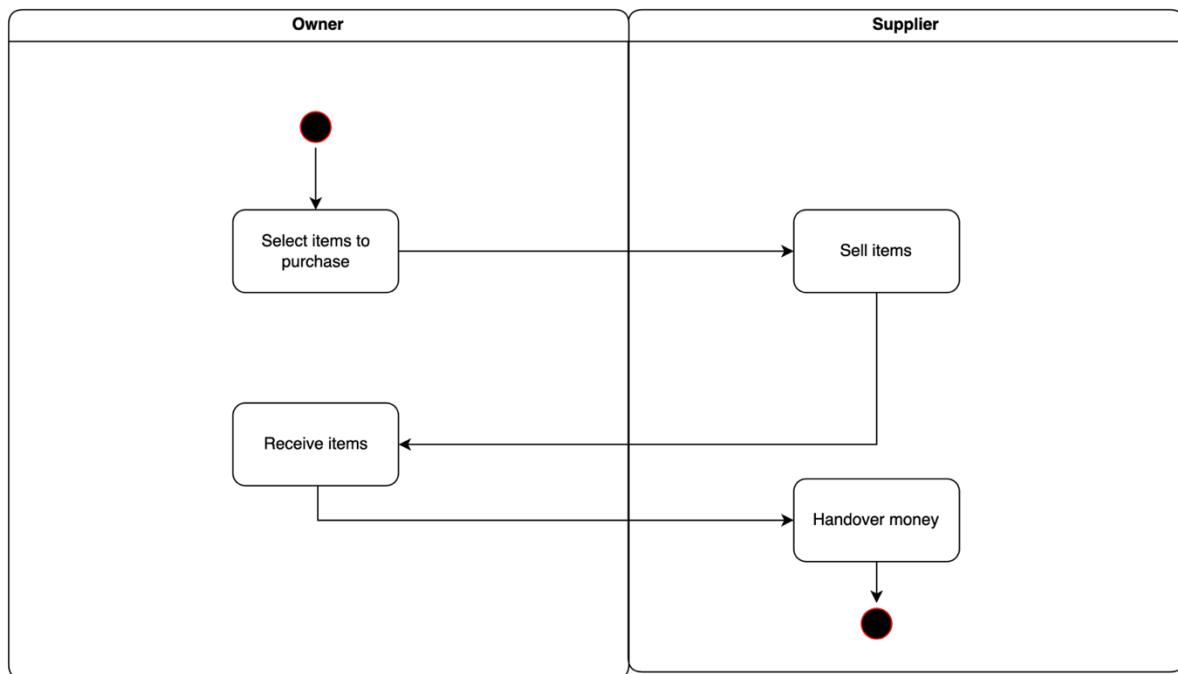
2.2 Activity Diagrams to describe the functionality of the Use-Cases

The activity diagrams are used to elaborate on the functionality of the previously described use cases. These diagrams showcase how the actors within the current business process interact with one another to accomplish business functions, and identify the decision-making points relevant to their specific roles.

2.2.1 This figure illustrates the activity diagram to describe the functionality of how the internal process works.



2.2.2 This figure illustrate the activity diagram to describe the functionality of how the process going on within owner and the supplier.



2.3 Requirement Catalogue

The SRS is an official report of what the system developers should implement (Sommerville, 2011, p. 91). This includes information such as what the system should do (functional requirements), how well the system should perform (nonfunctional requirements), and any constraints on the system's design.

This subsection delivers the functional and non-functional requirements for the proposed system collected in the requirement-gathering phase.

2.3.1 Functional Requirements

The system requirements analysis process aims to offer a thorough explanation of the problem based on the principles outlined in the particular problem. The following section will give a brief overview of the functional and non-functional requirements identified during the system analysis phase.

ID	Description	Priority	Weight
User Management			
1.1	The system shall allow administrators to create user accounts for different roles (cashier, warehouse handler).	H	H
1.2	Users shall be able to log into their accounts using a username and password	H	H
1.3	Administrators shall be able to update user information and roles.	M	H
1.4	Administrators shall be able to delete user accounts.	M	H
Customer Management			
2.1	The system shall allow the creation of customer accounts.	H	H
2.3	Cashier and administrators shall be able to update customer information.	M	M
2.4	Administrators shall be able to delete customer accounts.	M	H
2.5	The system shall allow the addition and storage of National ID card details to customer accounts.	L	M
2.6	The system shall track and display the rental history of each customer.	M	M
Equipment Management (Inventory)			
3.1	The system shall allow the creation of records for new equipment.	M	H

3.2	Administrators shall only be able to delete equipment records.	H	H
3.3	The system shall allow updating equipment details.	H	H
3.4	The system shall allow deleting equipment records.	M	H
3.5	The system shall display a categorized view of the entire inventory.	M	M

Invoice Management

4.1	The system shall allow the creation of invoices for equipment rentals.	H	H
4.2	The system shall allow searching for invoices using various criteria.	H	H
4.3	The system shall allow updating invoice details.	H	H
4.4	The system shall allow deleting invoices.	L	H
4.5	The system shall allow marking equipment as defected upon handover and receipt.	M	L
4.6	The system shall track deposit amounts and display the balance when returned.	H	H
4.7	The system shall allow entering installment payments with relevant dates.	H	H

Role based access

5.1	Shall be able to log into the system by multiple users accounts with different access levels (e.g., admin,cashier,warehouse handler).	H	H
5.2	Shall be able to define roles and permissions to control access to various system functionalities.	H	H

Report Generation

6.2	The system shall generate collection summary reports.	M	H
6.3	The system shall track and report on current, defected, and outgoing inventory.	M	H
6.4	The system shall track and report on Equipment utilization	M	H
6.5	The system shall track and report on Equipment revenue	M	H
6.6	The system shall track and report on under utilized Equipment	L	L
6.7	The system shall track and report on equipment maintainance needs	L	L
6.8	The system shall track and report on incomplete rentals by equipment	H	H
6.9	The system shall track and report on detailed invoice report to check past invoices	M	H

6.10	The system shall track and report on deleted invoices	M	H
6.11	The system shall track and report on sale against customers	M	H
6.12	The system shall track and report on invoices based on customer ratings	M	H
Warehouse handler requirements			
7.1	Warehouse handler shall be able to check the invoices placed by the cashiers realtime	H	H
Customer notifications			
8.2	The system shall send automated messages to customers for various notifications (e.g reminders,updates)	L	L

2.3.2 Nonfunctional requirements

Non-functional requirements describe the behavior of the system other than its main functionalities. The usability, reliability, performance, maintainability, and other similar aspects of the system are described by the non-functional requirements. These sets of requirements may not be directly related to the main functionality, but they are of extreme importance to the proper functioning of the system.

The following table illustrates the non-functional requirements for the proposed system including the priorities and weights.

User Interface			
1.1	The system shall provide a user-friendly graphical interface.	H	H
1.2	The system shall display invoices in both English and Sinhala.	L	M
Cross -Platform compatibility			
2.1	The system shall run on any web browser (e.g., Chrome, Firefox, Safari).	M	M
Performance			
3.1	The system shall generate invoices within 20 seconds, accounting for printer delays.	M	H

Security aspect			
4.1	Shall only let authenticated roles to access the back-end servers.	H	H
4.2	The system shall encrypt all sensitive data before transmitting it over insecure connections, such as the internet, as determined by the security policy	H	H
4.3	The system shall have a minimum uptime of 99.9% per month, measured over a period of at least one year.	H	H

3.SYSTEM DESIGN

This chapter centers on system design, leveraging the specifications detailed in Chapter 2's System Requirement Specification. The design will be articulated through a range of diagrams, encompassing Use Case Diagrams, Activity Diagrams, Entity-Relationship Diagrams, Entity Class Diagrams, Controller Class Diagrams, Interface Class Diagrams, and Sequence Diagrams. The culmination of the chapter will involve presenting a Normalized Database Design, complete with database specifications, alongside Wireframes illustrating the anticipated Graphical User Interfaces.

Outline of the Chapter

- 3.1 Use case diagrams for the proposed system
- 3.2 Activity diagrams for the proposed system
- 3.3 Sequence diagrams for the proposed system
- 3.4 Entity relationship diagram for the proposed system
- 3.5 Entity-Class diagram for the proposed system
- 3.6 Interface-Class diagram for the proposed system
- 3.7 Controller-Class diagram for the proposed system

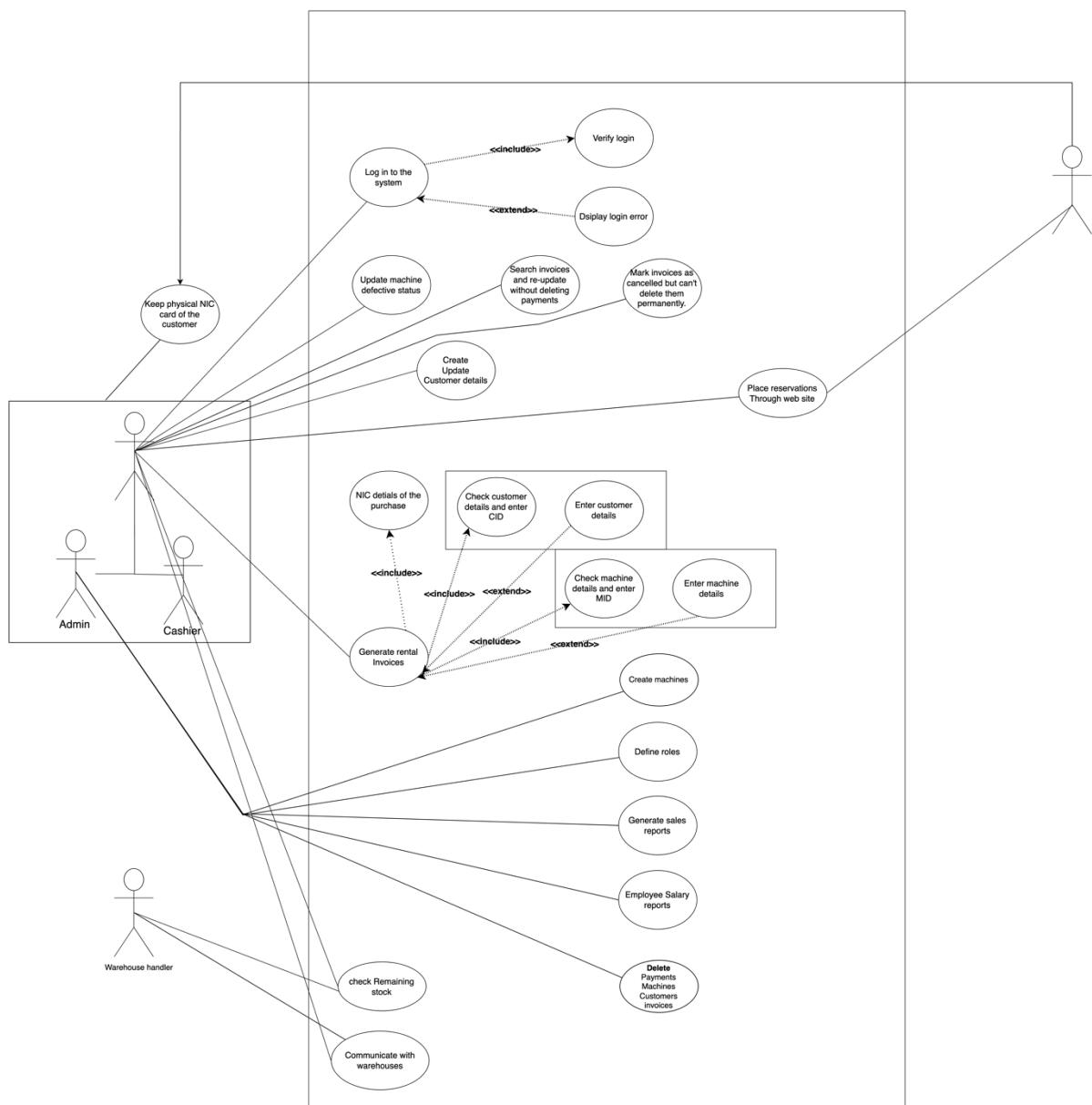
3.1 Use case diagrams for the proposed system

Use case diagrams serve as a holistic representation of a system's functional requirements, encompassing both internal and external elements pivotal for system design and implementation. These requirements play a crucial role in shaping the Software Development Life Cycle, as the development of use cases and actor identification precedes the system design phase. In the context of Use Case diagrams, multiple use cases coexist with system actors, where actors represent individuals interacting with the system. The figures showcasing these use case diagrams delineate the envisioned system functionality and users' interactions, employing actors and use cases to provide a clear overview.

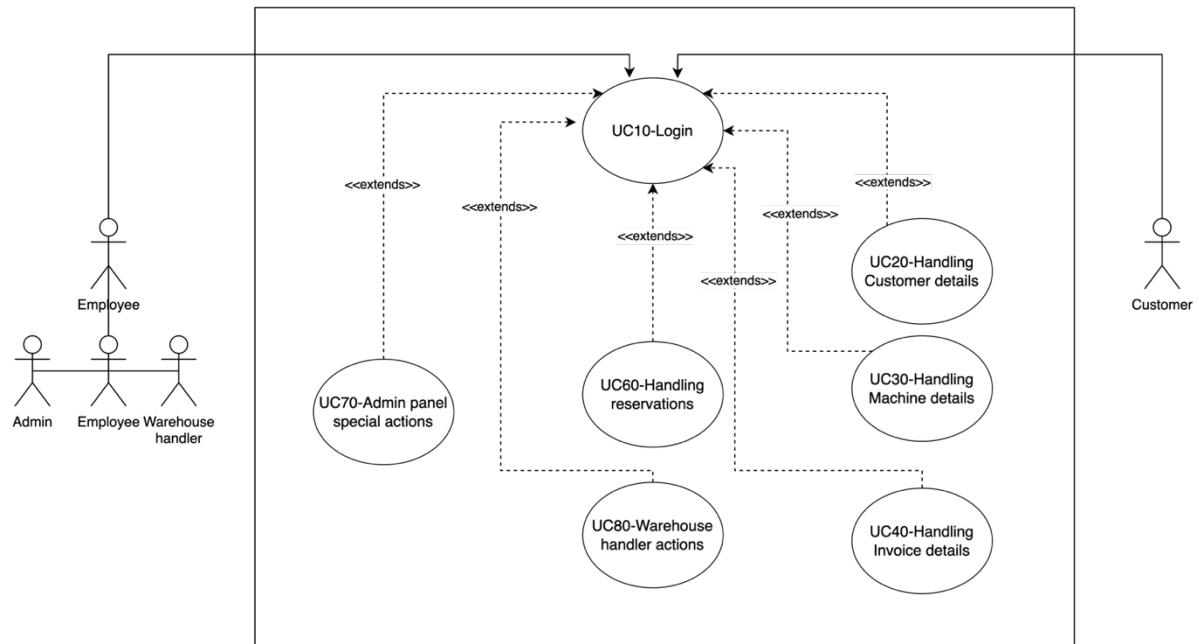
Actors of proposed system:

- Admin
- Cashier
- Warehouse handlers

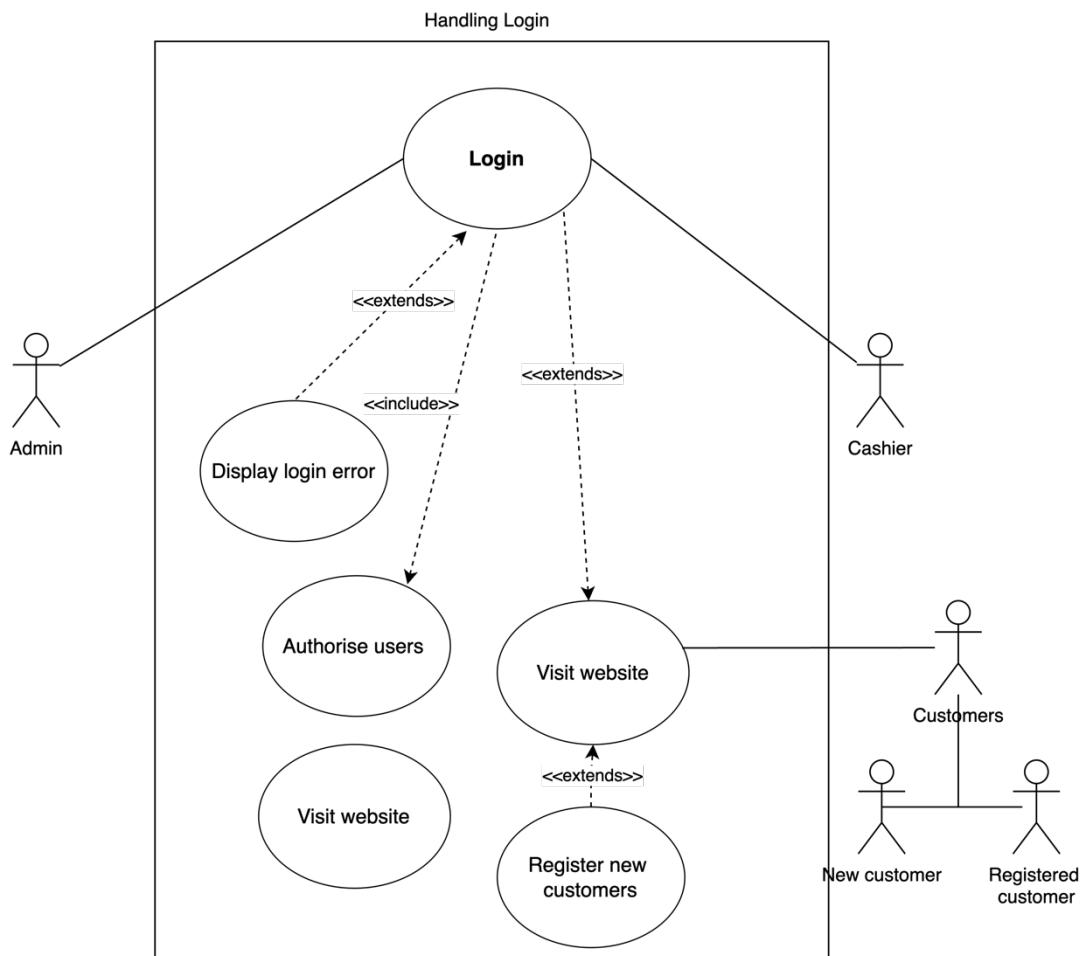
3.1.1 Overall Use Case Diagram for the Proposed System



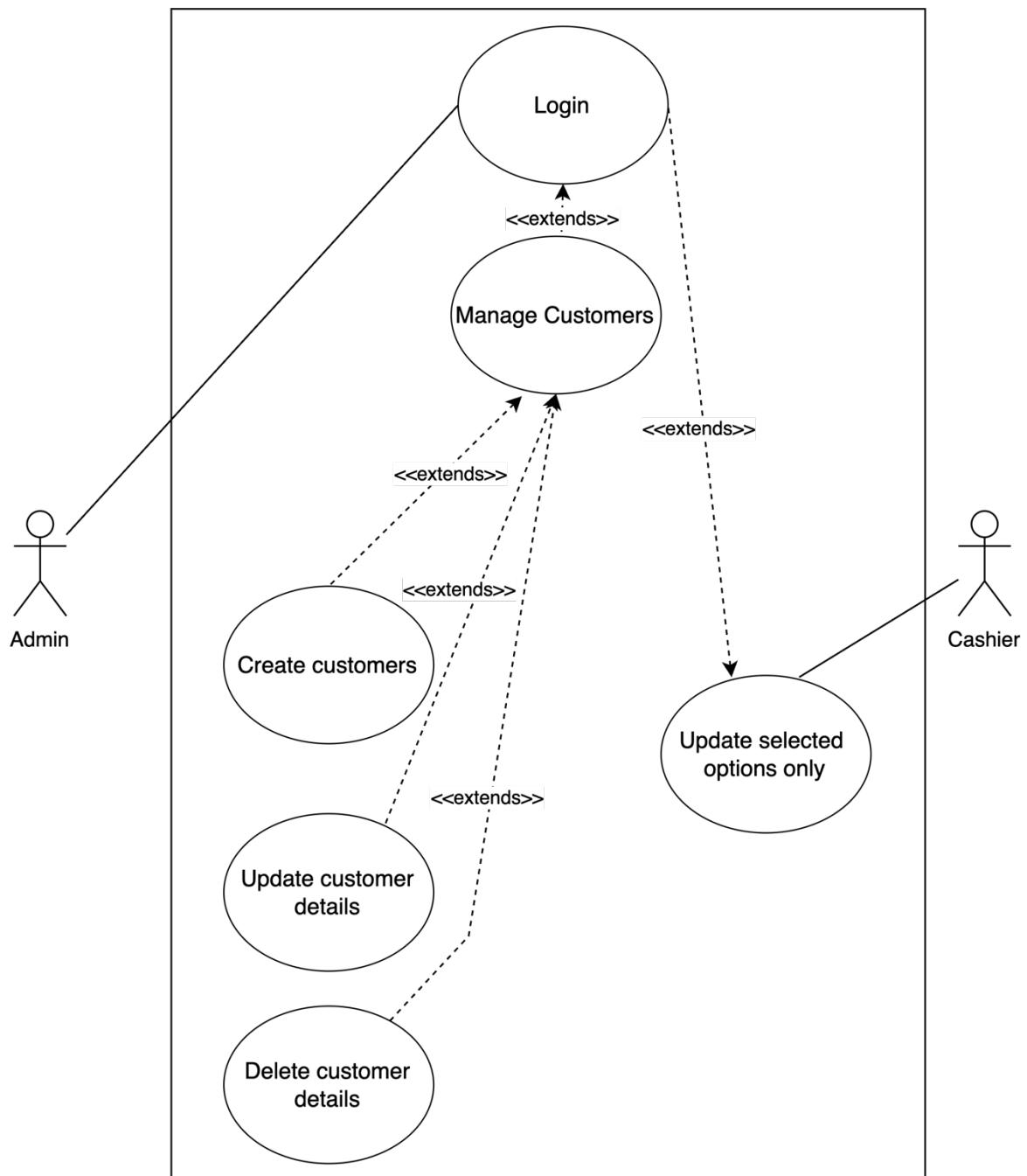
3.1.2 High level view of the Use case diagram: functionalities of the proposed system



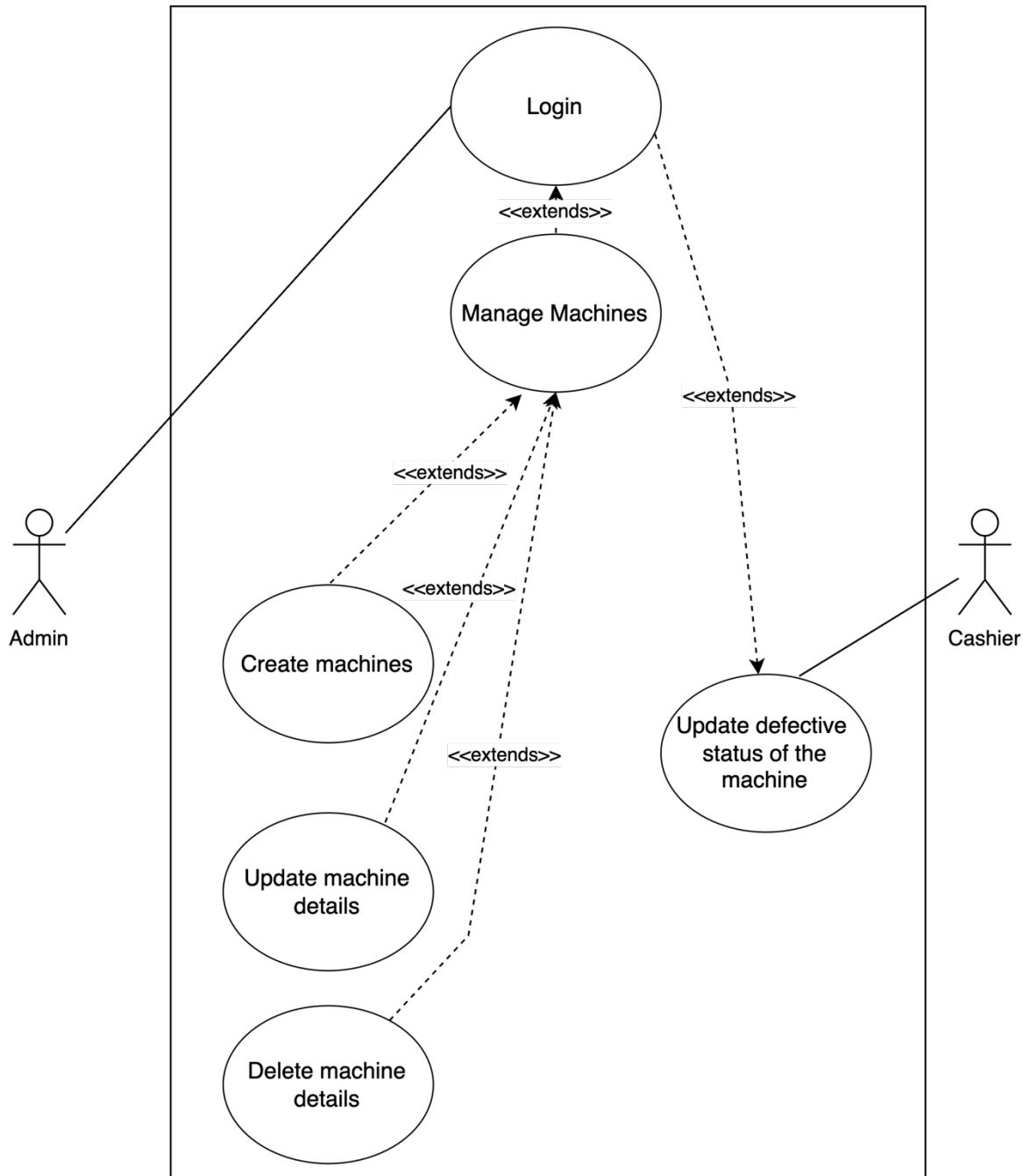
3.2.3 Use case diagram for describing functionalities: Login use case



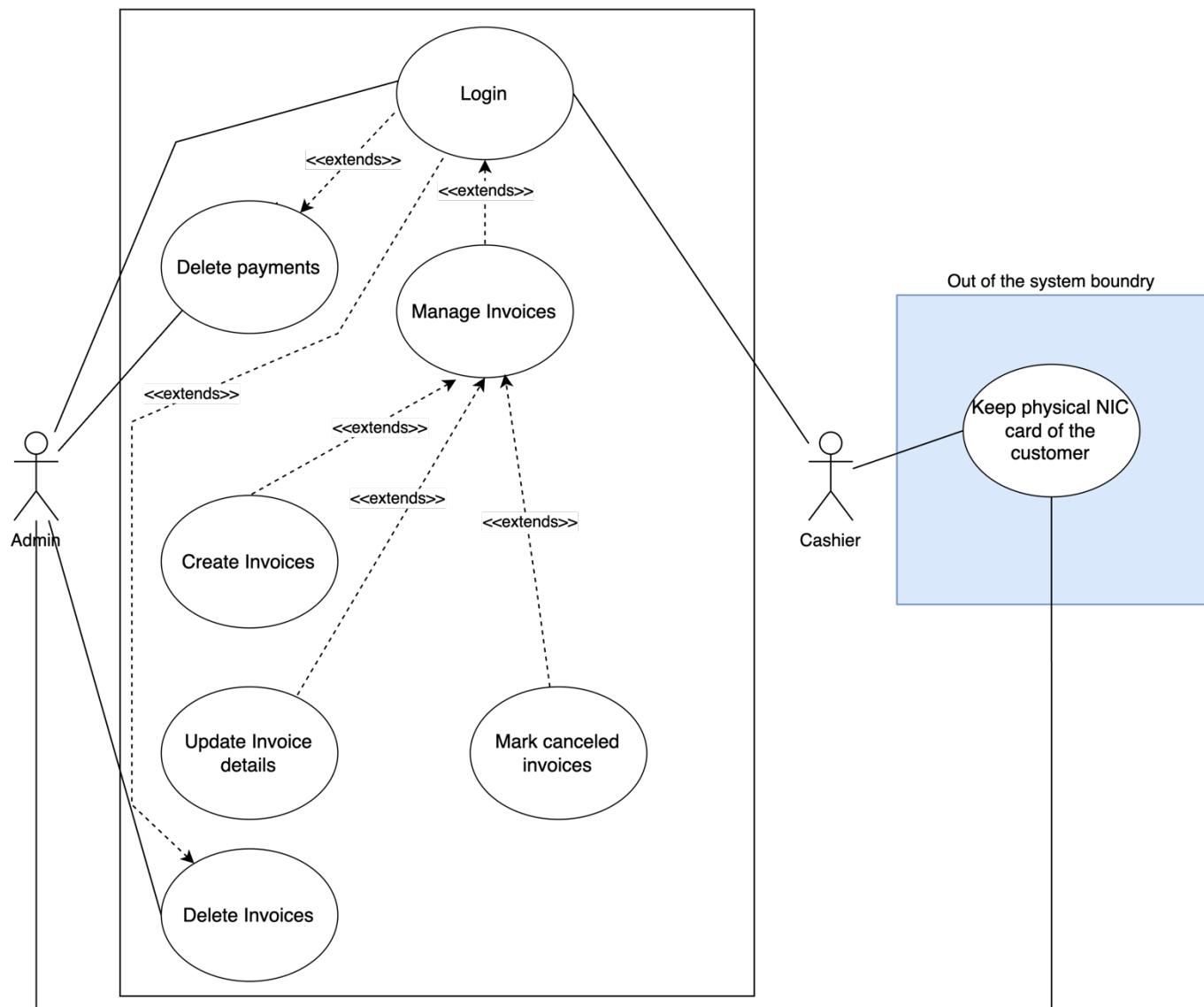
3.1.4 Use case diagram for describing functionalities: Handling Customer details use case



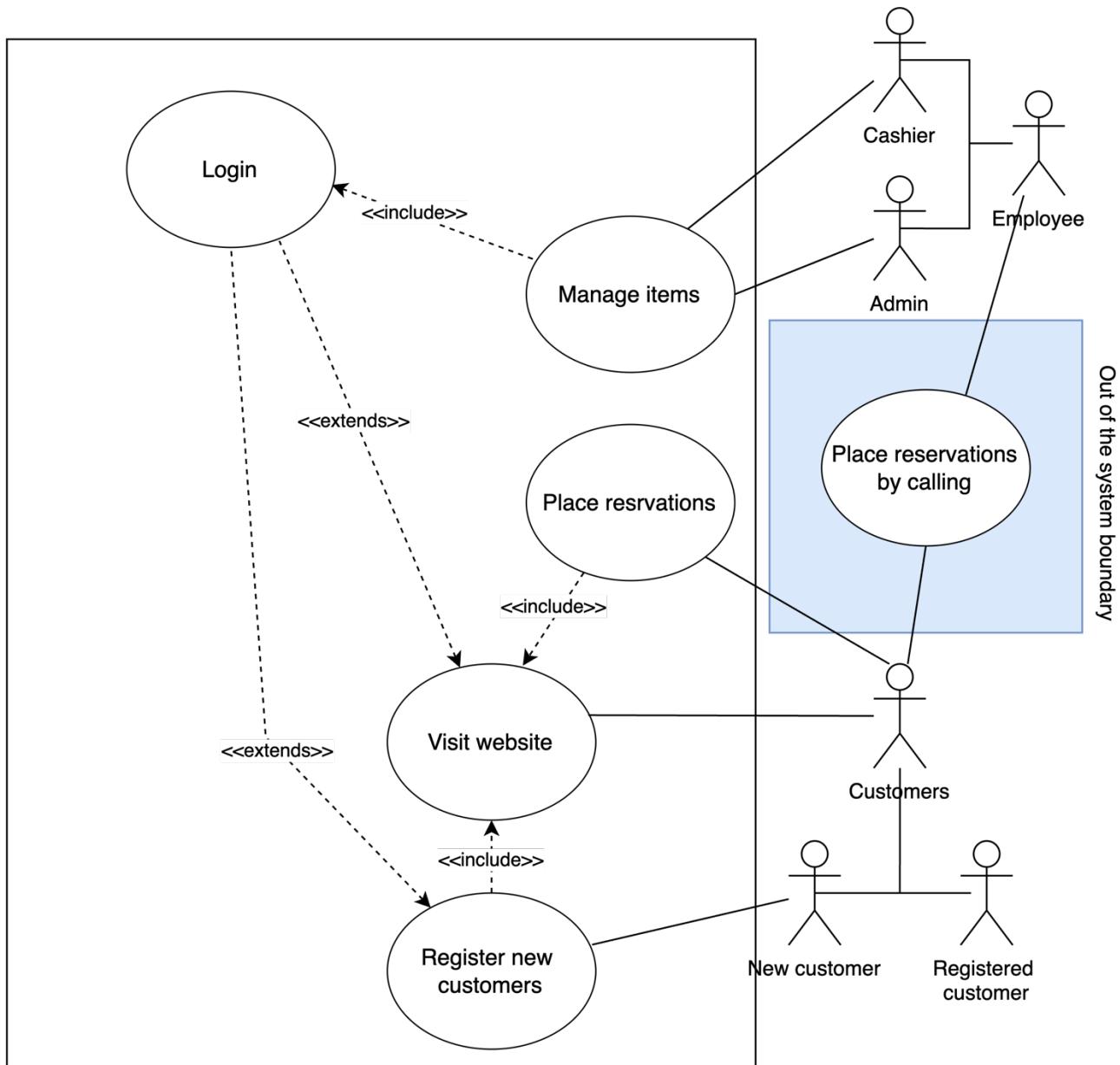
3.1.5 Use case diagram for describing functionalities: Handling Machine details use case



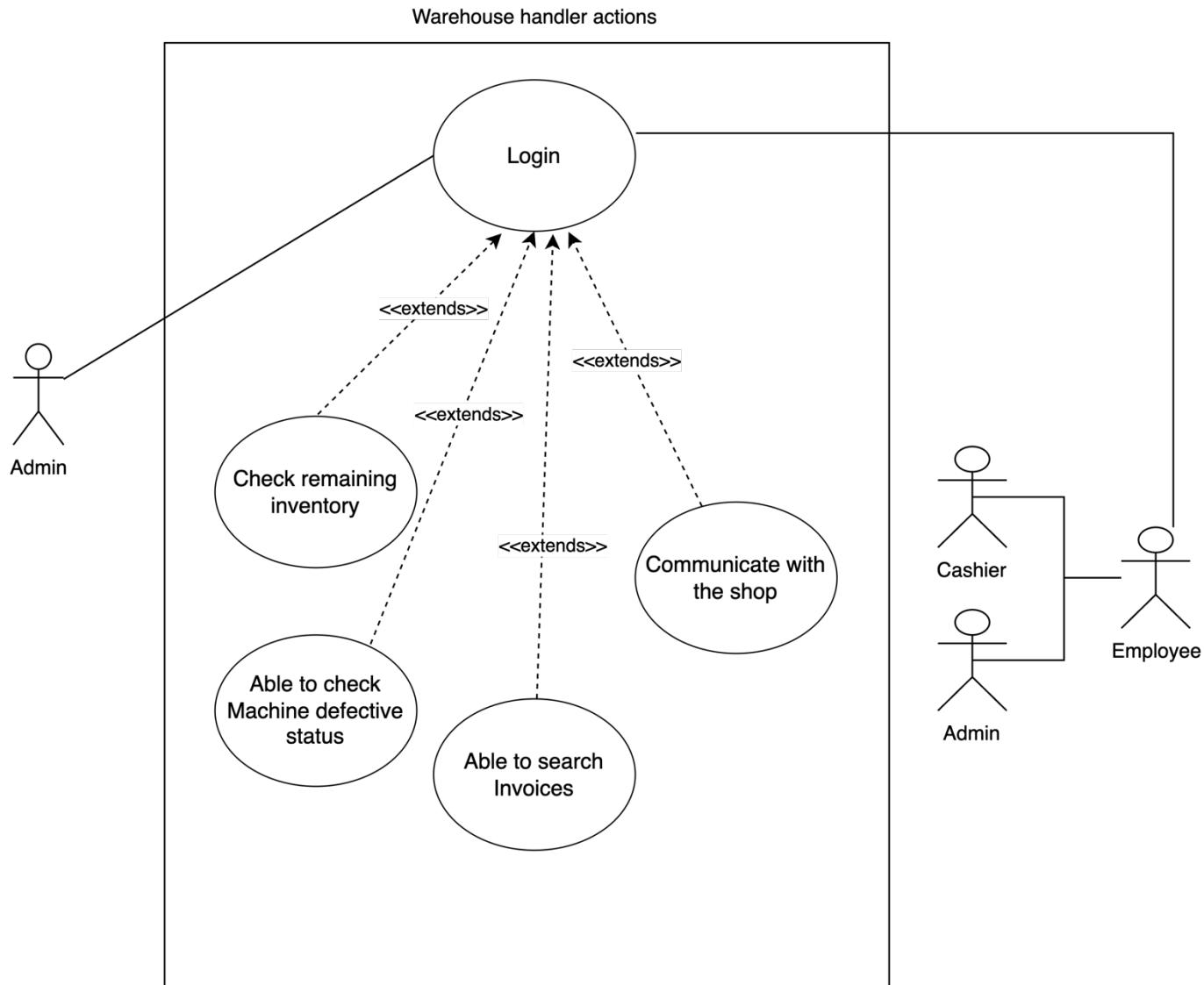
3.1.6 Use case diagram for describing functionalities: Handling Invoice details use case



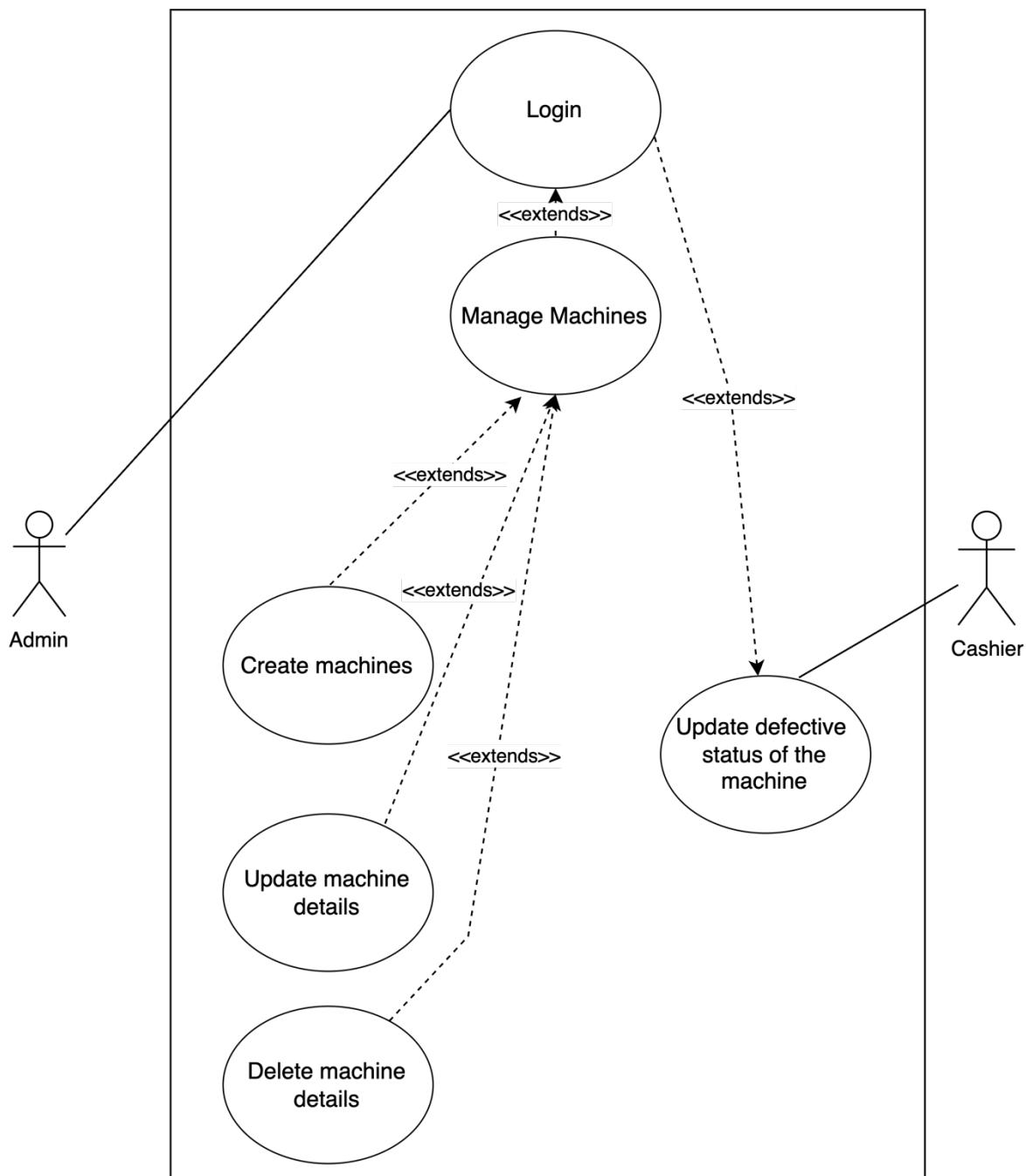
3.1.7 Use case diagram for describing functionalities: Handling Reservations use case



3.1.8 Use case diagram for describing functionalities: Warehouse handler actions use case



3.1.9 Use case diagram for describing functionalities : Admin panel special actions use case



3.2 User Stories and Use Case diagram for Existing System

The identification of user stories offers an end-user perspective of the system, defining software functions. These stories typically include the type of user, their desired outcome, and the reason for their request. The chapter presents a table with several user stories, outlining the functional and non-functional features specified by different user types.

Use Case ID	UC 10
Use Case Name	The process of login to the system.
Participating Actors	Business owner/ Admin, Cashier, Warehouse handler, Customers
Use Case Description	This use case describes the process of log in to the system to access the functionality of the system.
Entry condition	For the customer - should have to be registered in the system For employee - should have a valid account in the system
Basic Flow of events	1. User visits the Website and request for login 2. Enter Username and Password 3. The system verifies the Username & Password 4. The system identifies the user type 5. User logged on to the system 6. Display available options in the platform
Alternate Flow/Exceptions	2.a. New User, Direct user to the registration 2.b. Username/Password forgot, direct to Recover Password 3.a. Invalid Username or Password System displays an error message 3.b. All fields are not filled System displays an error message
Exit condition	User logs into the system

3.2.1 Use case description for manage customers by staff

Use Case ID	UC 20
Use Case Name	The process of handling customer details
Participating Actors	Business owner/ Admin, Cashier
Use Case Description	This use case describes the process of manage customer details by the staff in the shop.
Entry condition	The system is running and accessible. The user (Admin or Cashier) has a valid username and password. The user has the necessary permissions to access the system.
Basic Flow of events	Admin and Cashier login to the system Execute method 1 or 2 Method 1- Go to the customer section and add new customer, update and delete with their respective permissions. Method 2- In the invoice section, when filling out an invoice, the system will automatically register the customer if they are new to the system.
Alternate Flow/Exceptions	3.1.a. Invalid email, System displays an error message and allows the user to re-enter the email address 3.2.a. Invalid telephone number, System displays an error message and allows the user to re-enter the telephone number 3.3.a. Created password does not match with the re-entered password, the system displays an error message and allows the user to re-enter the password 5.a. Details incomplete, System displays an error message and allows the user to complete the form
Exit condition	New customer has been registered

Use Case ID	UC 30
Use Case Name	The process of handling machine details
Participating Actors	Business owner/ Admin, Cashier
Use Case Description	This use case describes the process of manage machine details by the staff in the shop.
Entry condition	Admin or Cashier should be logged in to the system
Basic Flow of events	<p>Admin,</p> <p>1. Go to inventory panel</p> <p>2.1. Add new equipment.</p> <p>2.1.1. Add equipment details.</p> <p>2.1.2. Select equipment category.</p> <p>2.1.3. Upload photograph</p> <p>2.1.4. Submit details.</p> <p>2.1.5. Confirmation message</p> <p>3.2 Update equipment details</p> <p>3.2.1. Select equipment.</p> <p>3.2.2. Edit equipment details.</p> <p>3.2.3. Save equipment details.</p> <p>3.2.4. Confirmation message</p> <p>3.3 Remove equipment.</p> <p>3.3.1. Select equipment.</p> <p>3.3.2. Click Remove button.</p> <p>3.3.3. Confirmation message</p> <p>Cashier, Go to the inventory panel. Update the defective status. Save changes and exit.</p>
Alternate Flow/Exceptions	<p>3.1.a. Invalid email, System displays an error message and allows the user to re-enter the email address</p> <p>3.2.a. Invalid telephone number, System displays an error message and allows the user to re-enter the telephone number</p> <p>3.3.a. Created password does not match with the re-entered password, the system displays an error message and allows the user to re-enter the password</p> <p>5.a. Details incomplete, System displays an error message and allows the user to complete the form</p>
Exit condition	1.The machine details have been

	<p>successfully updated in the system.</p> <p>2.The user is redirected to the machine details page or another relevant interface.</p> <p>3.A confirmation message is displayed to inform the user that the update was successful.</p>
--	---

Use Case ID	UC 40
Use Case Name	The process of handling invoice details
Participating Actors	Business owner/ Admin, Cashier
Use Case Description	This use case describes the process of manage invoice details by the staff in the shop.
Entry condition	Admin or Cashier should be logged in to the system
Basic Flow of events	<p>Go to invoice panel</p> <p>Add a new invoice.</p> <p>Search customers by phone number or NIC number.</p> <p>Fill customer details if he not an existing customer</p> <p>Search machine by machine ID.</p> <p>Enter advanced amount.</p> <p>Enter ID card handover details for the bill.</p> <p>Update invoices.</p> <p>Search invoice by Invoice ID or the Bill ID.</p> <p>Update relevant fields.</p> <p>Delete invoices.</p> <p>4.1. As an admin, he should be able to delete invoices.</p> <p>4.2. As a cashier, he should be able to mark invoices as cancelled.</p>
Alternate Flow/Exceptions	<p>Pending invoice option</p> <p>The Cashier decides to mark an invoice as pending instead of canceled.</p> <p>The Cashier selects the "Mark as Pending" option.</p>

	<p>The system updates the invoice status to "Pending," records the action, and ends the use case.</p> <p>Error in Adding New Invoice While adding a new invoice, the Admin encounters an error entering the advanced amount.</p> <p>The system displays an error message.</p> <p>The Admin corrects the amount and continues with the invoice creation.</p> <p>The use case continues with the corrected information.</p>
Exit condition	New customer has been registered

Use Case ID	UC 60
Use Case Name	The process of handling reservations
Participating Actors	Business owner/ Admin, Cashier, Customers (New customers, Registered customers)
Use Case Description	This use case describes the procedure and how users manage the reservation process.
Entry condition	<p>Admin or Cashier should be able to log in to the system</p> <p>Customer should be able to log in to the website to place online reservations</p>
Basic Flow of events	<p>Customers log in to the website and choose an item to reserve.</p> <p>The website asks from customer to login.</p> <p>Customer login to the website</p> <p>Customer is registered in the system and reservation placed.</p> <p>Message box popup and ask customer to contact the shop to verify the reservation.</p> <p>Admin or cashier login to the system</p> <p>Enter to the reservation panel.</p> <p>Check incoming reservations.</p> <p>Check availability for the date.</p> <p>Accept the reservation.</p> <p>Call customer to verification</p>

Alternate Flow/Exceptions	<p>Customer Not Registered The website identifies that the customer is not registered in the system.</p> <p>The system prompts the customer to register.</p> <p>The customer completes the registration process.</p> <p>The system continues with the reservation process.</p>
Exit condition	New customer has been registered

Use Case ID	UC 70
Use Case Name	The process of handling admin panel special actions
Participating Actors	Business owner/ Admin
Use Case Description	This use case describes what are the options that admin has
Entry condition	Admin should be able to log in to the system
Basic Flow of events	<p>Admin login to the system</p> <p>Enter to the Reports section.</p> <p>Select report type.</p> <p>Filter options.</p> <p>Generate reports.</p>
Alternate Flow/Exceptions	<p>Invalid report type Step 3.1.1: The admin selects an invalid or non-existent report type.</p> <p>Step 3.1.2: The system displays an error message indicating an invalid report type.</p> <p>Step 3.1.3: The admin is prompted to select a valid report type.</p> <p>Step 3.1.4: The admin selects a valid report type and continues with the report generation process.</p>

	<p>Error in report generation</p> <p>Step 3.1.1: The admin selects an invalid or non-existent report type.</p> <p>Step 3.1.2: The system displays an error message indicating an invalid report type.</p> <p>Step 3.1.3: The admin is prompted to select a valid report type.</p> <p>Step 3.1.4: The admin selects a valid report type and continues with the report generation process.</p>
Exit condition	<ul style="list-style-type: none"> a. The Admin selects a valid report type. b. The Admin applies filter options as needed. c. The Admin triggers the report generation process. d. The system successfully generates the requested report.

Use Case ID	UC 80
Use Case Name	Warehouse handler actions
Participating Actors	Admin, Cashier, Warehouse handler
Use Case Description	This use case describes what are the options that the warehouse handler has
Entry condition	Warehouse handler needs to be log in to the system
Basic Flow of events	<p>Search Invoices:</p> <p>The Warehouse Handler selects the "Search Invoices" option.</p> <p>The system provides a search interface.</p> <p>The Warehouse Handler enters relevant criteria (e.g., date range, customer name) to search invoices.</p> <p>The system displays matching invoices.</p> <p>Communicate with the Shop (Cashier and Admin):</p> <p>The Warehouse Handler selects the "Communicate" option.</p> <p>The system provides a messaging interface.</p> <p>The Warehouse Handler can send messages</p>

	<p>to the Cashier and Admin regarding inventory, shipments, or other relevant matters.</p> <p>See Machine's Defective Status:</p> <p>The Warehouse Handler selects the "View Machine Defective Status" option.</p> <p>The system displays a list of machines with their respective defective statuses.</p> <p>Check Remaining Inventory:</p> <p>The Warehouse Handler selects the "Check Remaining Inventory" option.</p> <p>The system provides information on the current inventory levels for various equipment.</p>
Alternate Flow/Exceptions	<p>Communication Error</p> <p>While trying to communicate with the shop, the system encounters an error.</p> <p>The system displays an error message indicating a communication issue.</p> <p>The Warehouse Handler may retry the communication task or contact technical support.</p>
Exit condition	Press handover button on the invoice that sent by the shop as a message

3.2.1 This is a simple graphical representation to distinguish the permissions (functionalities) of the main system actors clearly.

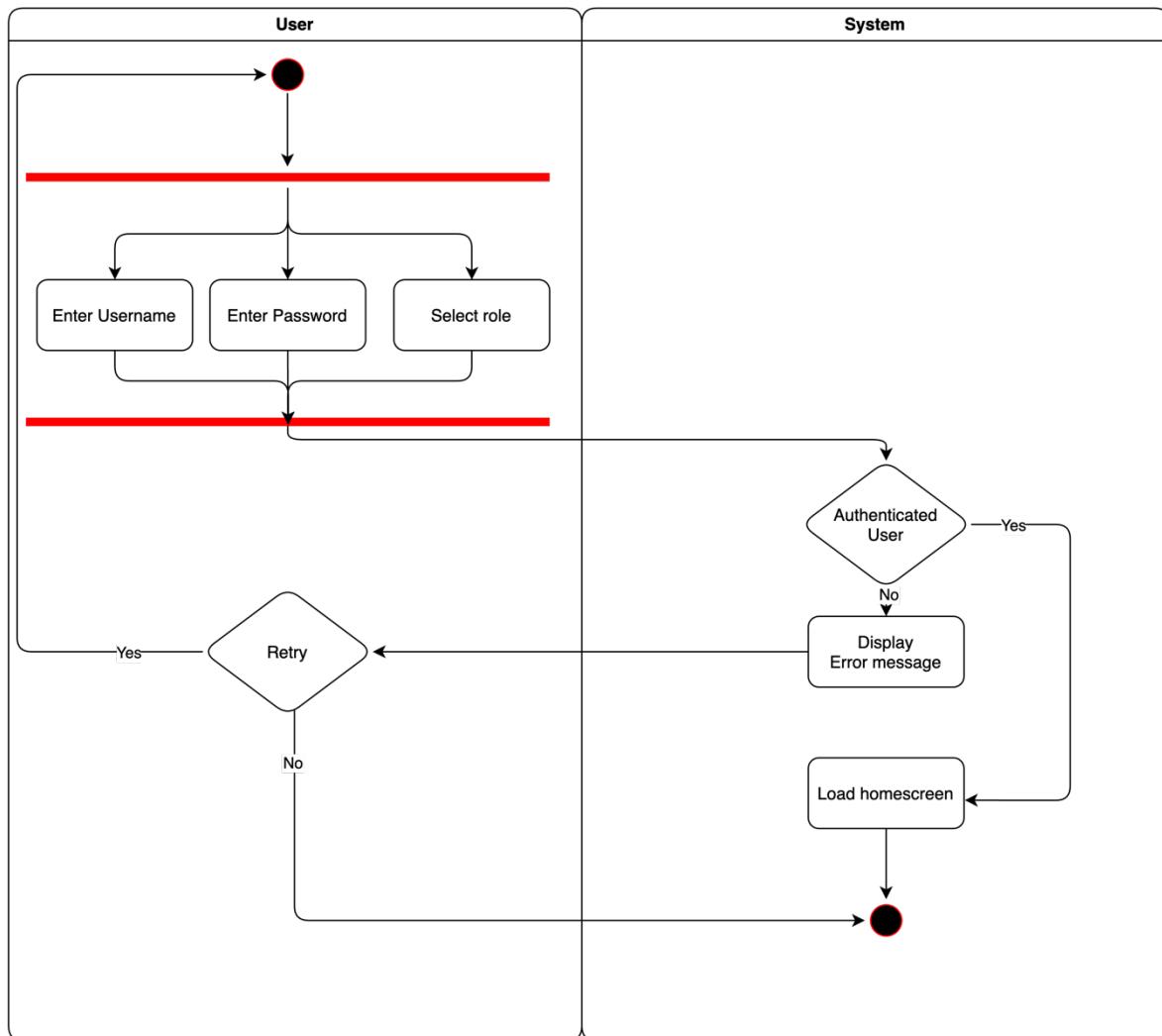
Admin			
	Machines	Create	Update
		✓	✓
Machines	Customers	✓	✓
Invoices	Payments	✓	✓
Payments		✓	✓

Cashier			
	Machines	Create	Update
		✗	✓ ✗ Only the defective status
Machines	Customers	✓	✓ ✗ Certain details only
Invoices	Payments	✓	✗ Can mark as cancelled
Payments		✓	✗

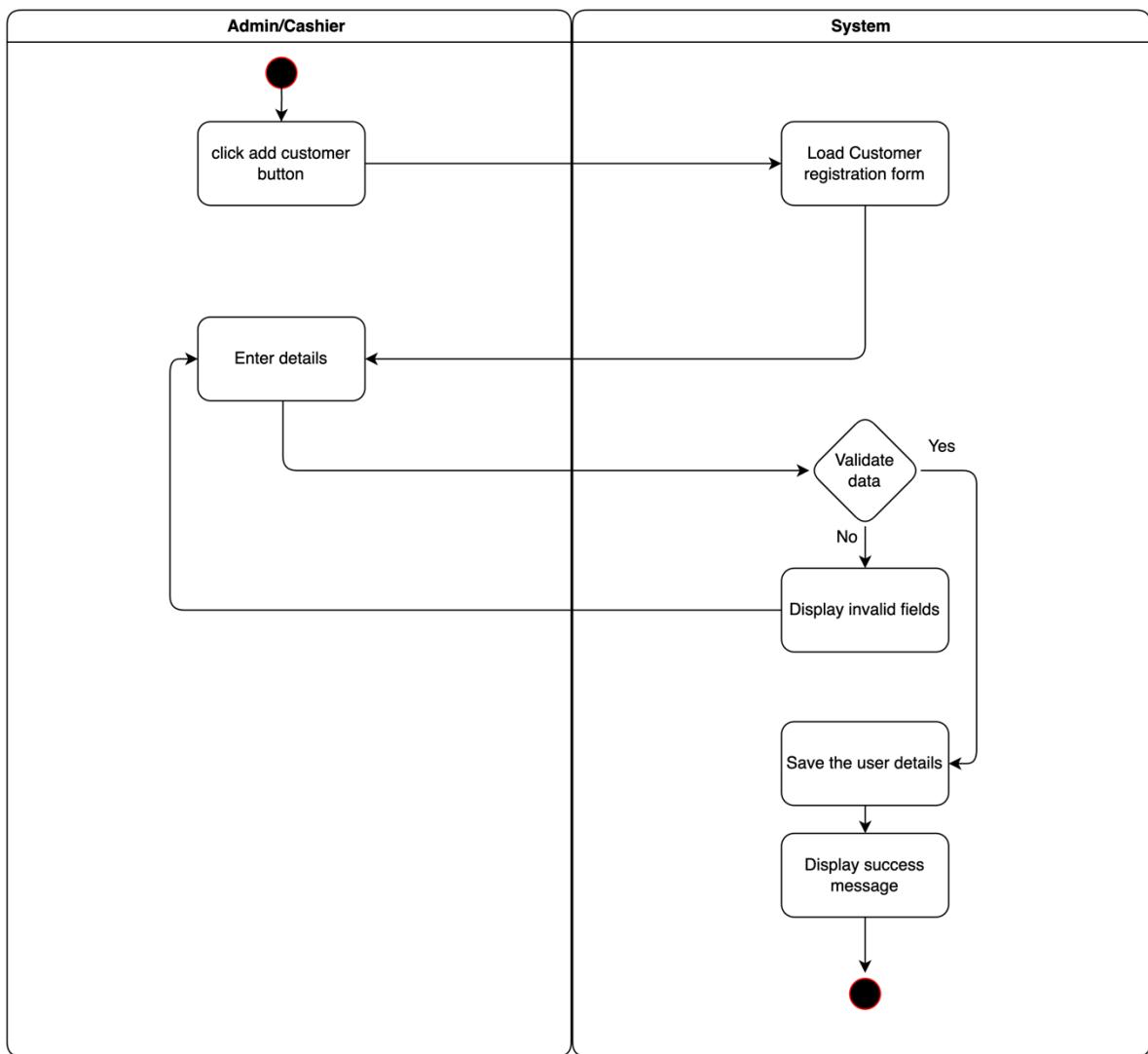
3.3 Activity diagram for the proposed system

Activity diagrams are crafted to visually illustrate the logical flow of tasks outlined in a use case, facilitating the design of the system's functions and decision-making processes. This section showcases the tasks associated with key functions and the involved users. Furthermore, it provides insights into concurrent activities, along with the pre- and post-conditions associated with each activity.

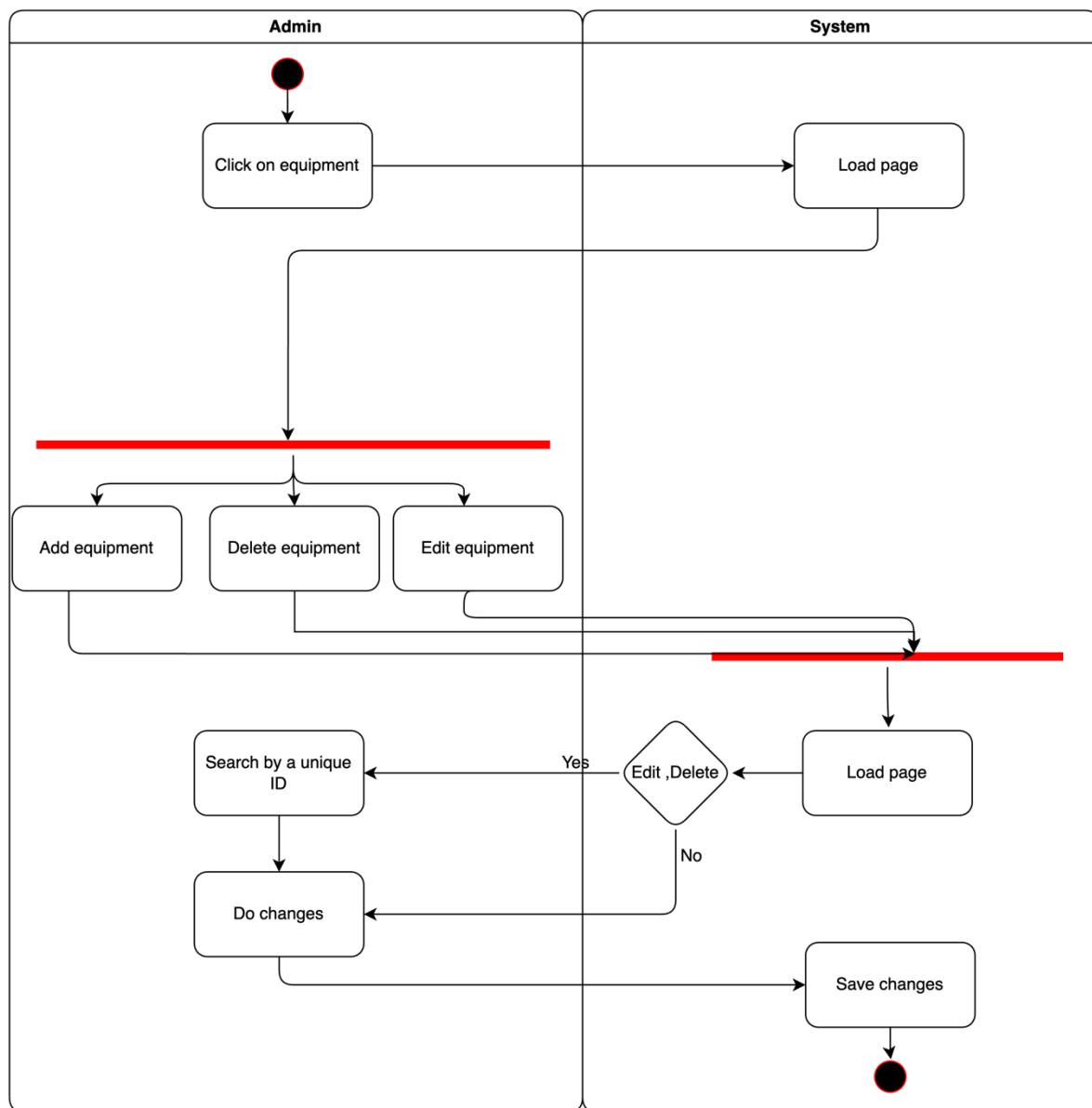
3.3.1 Activity diagram for describing functionalities: Login



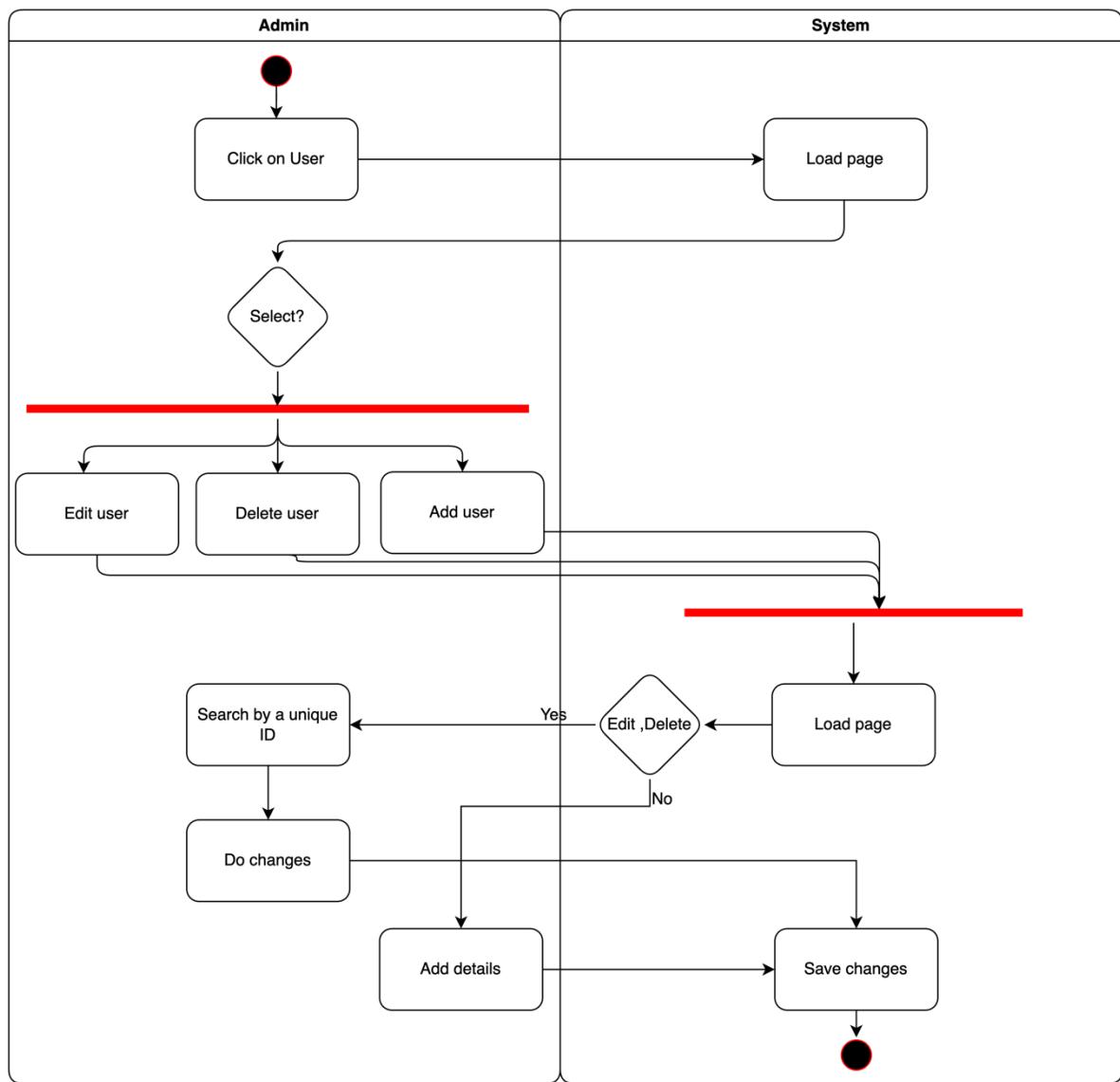
3.3.2Activity diagram for describing functionalities: Register customers



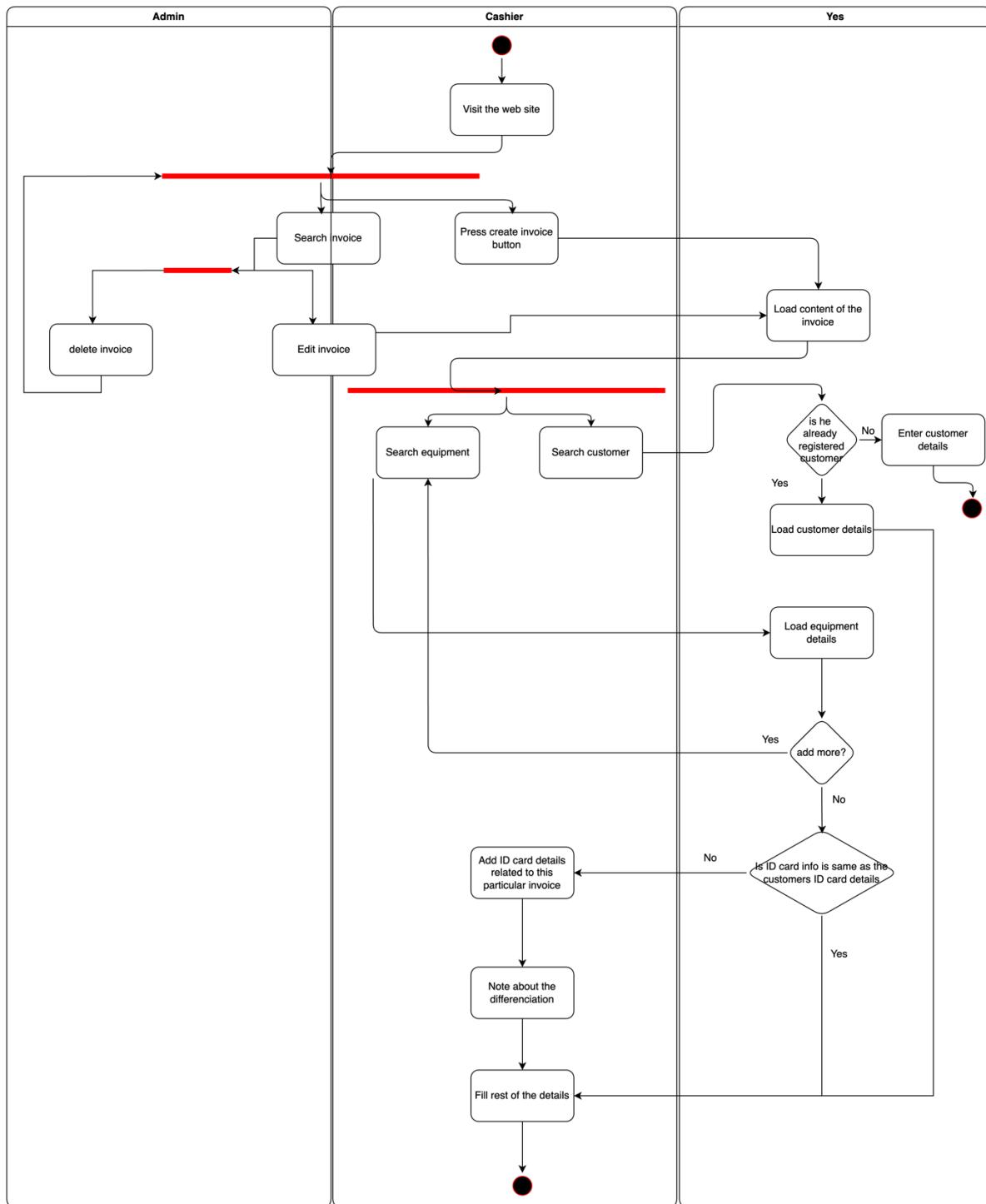
3.3.3Activity diagram for describing functionalities: Manage equipment.



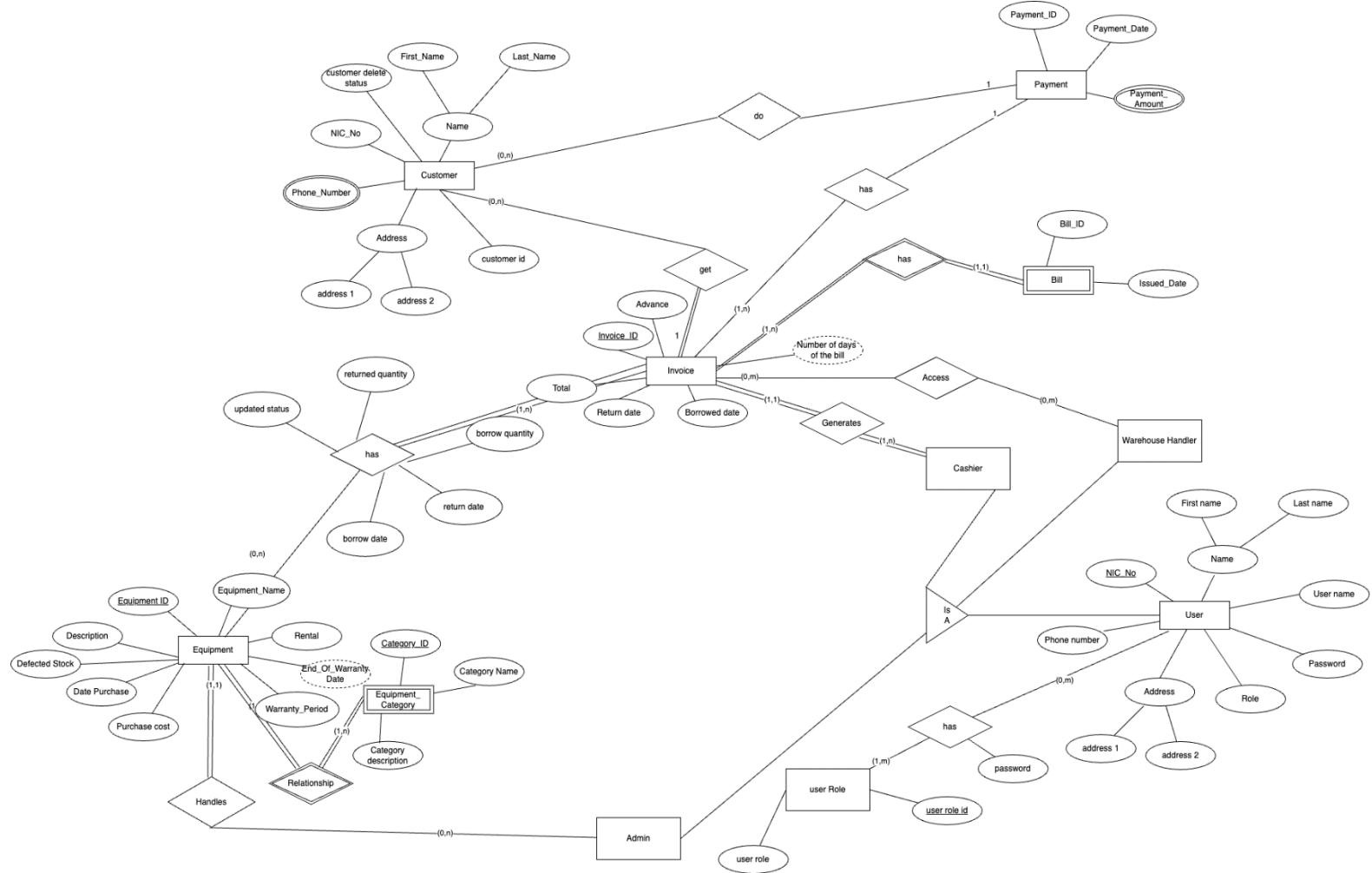
3.3.4 Activity diagram for describing functionalities: Manage users



3.3.5 Activity diagram for describing functionalities: Rent equipment



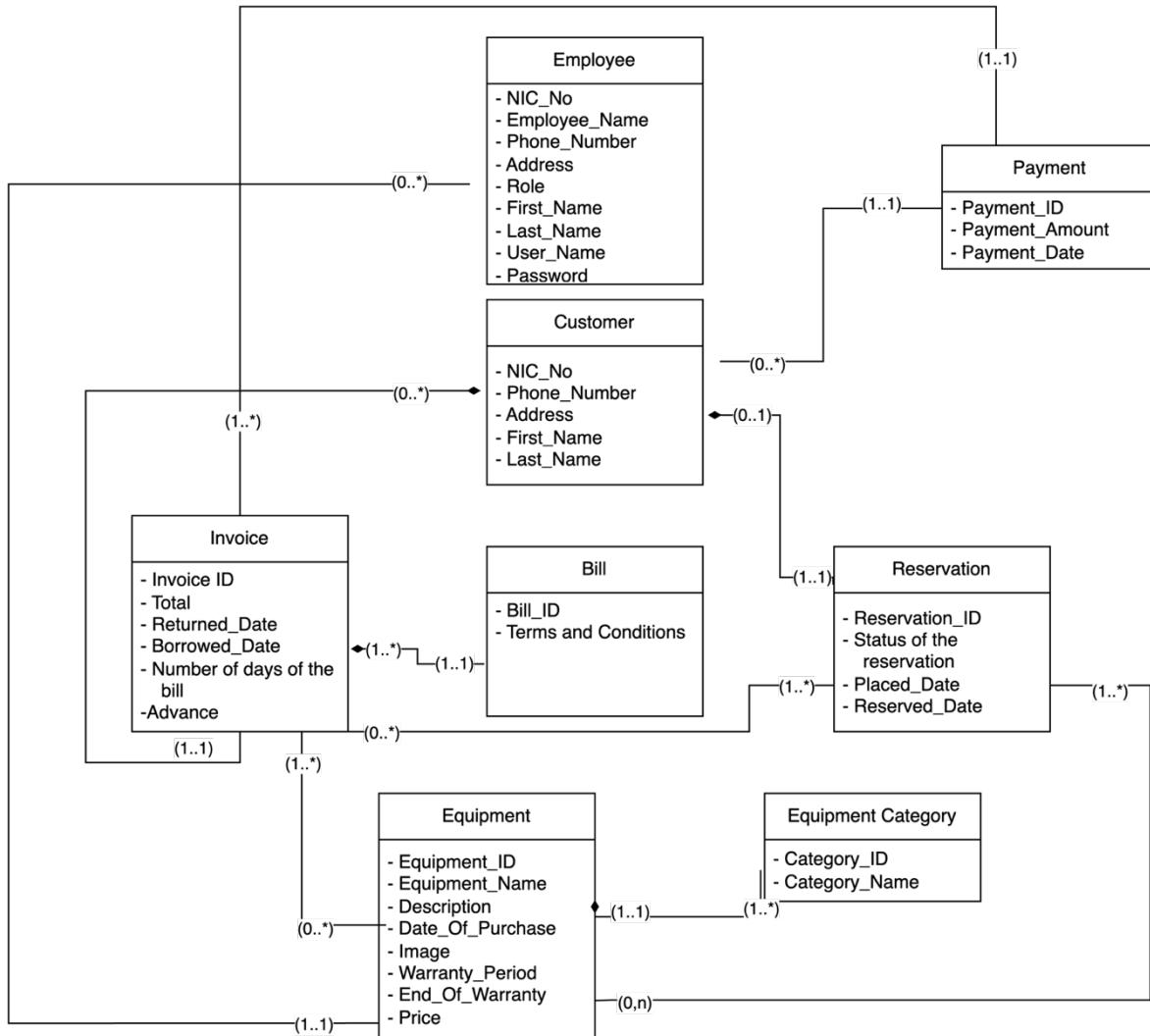
3.4 Entity relationship diagrams



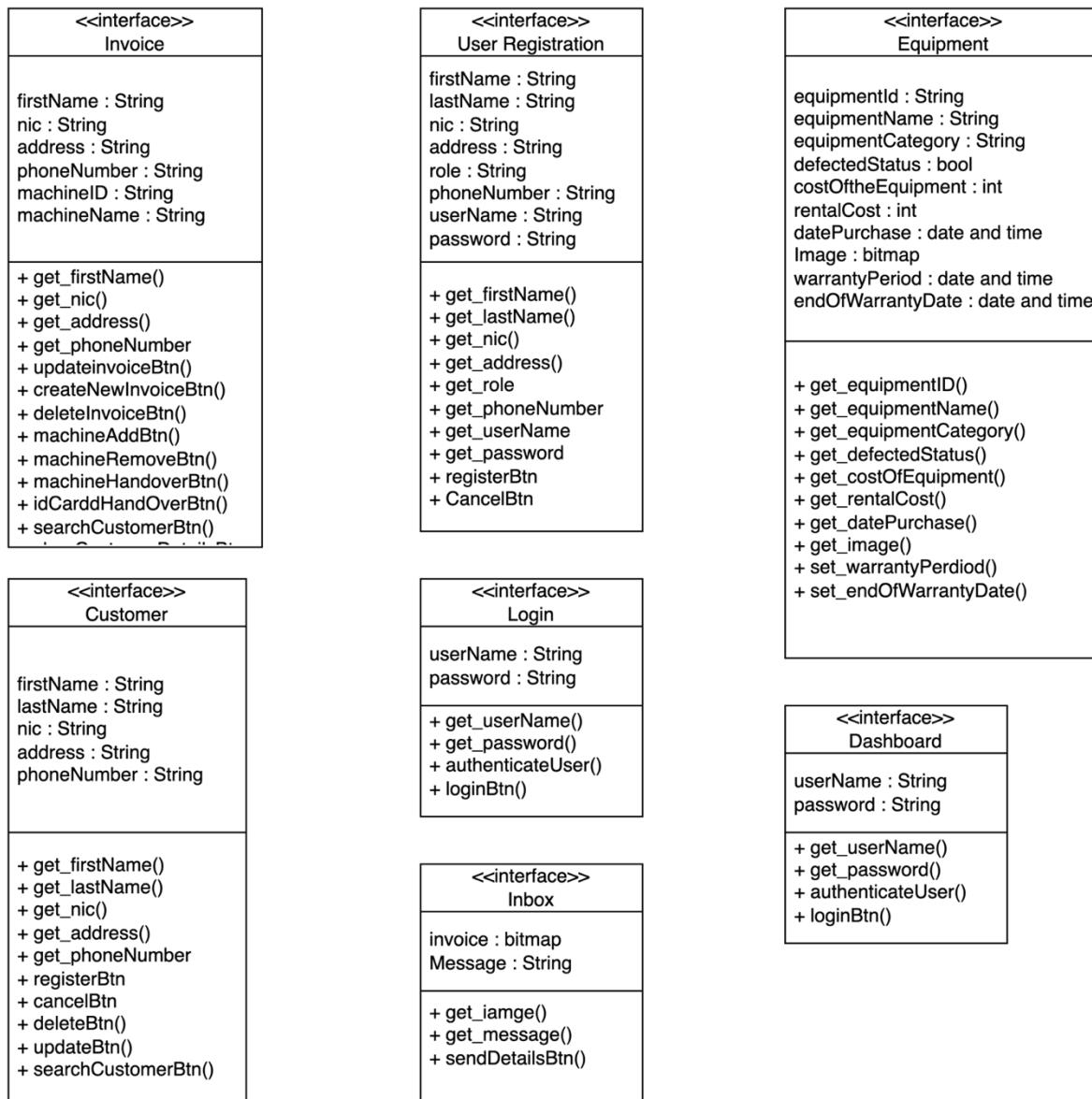
3.5 Class diagram for proposed system

Class diagram below describes the structure of proposed system by showing the system's classes, their attributes, and relationships among the classes.

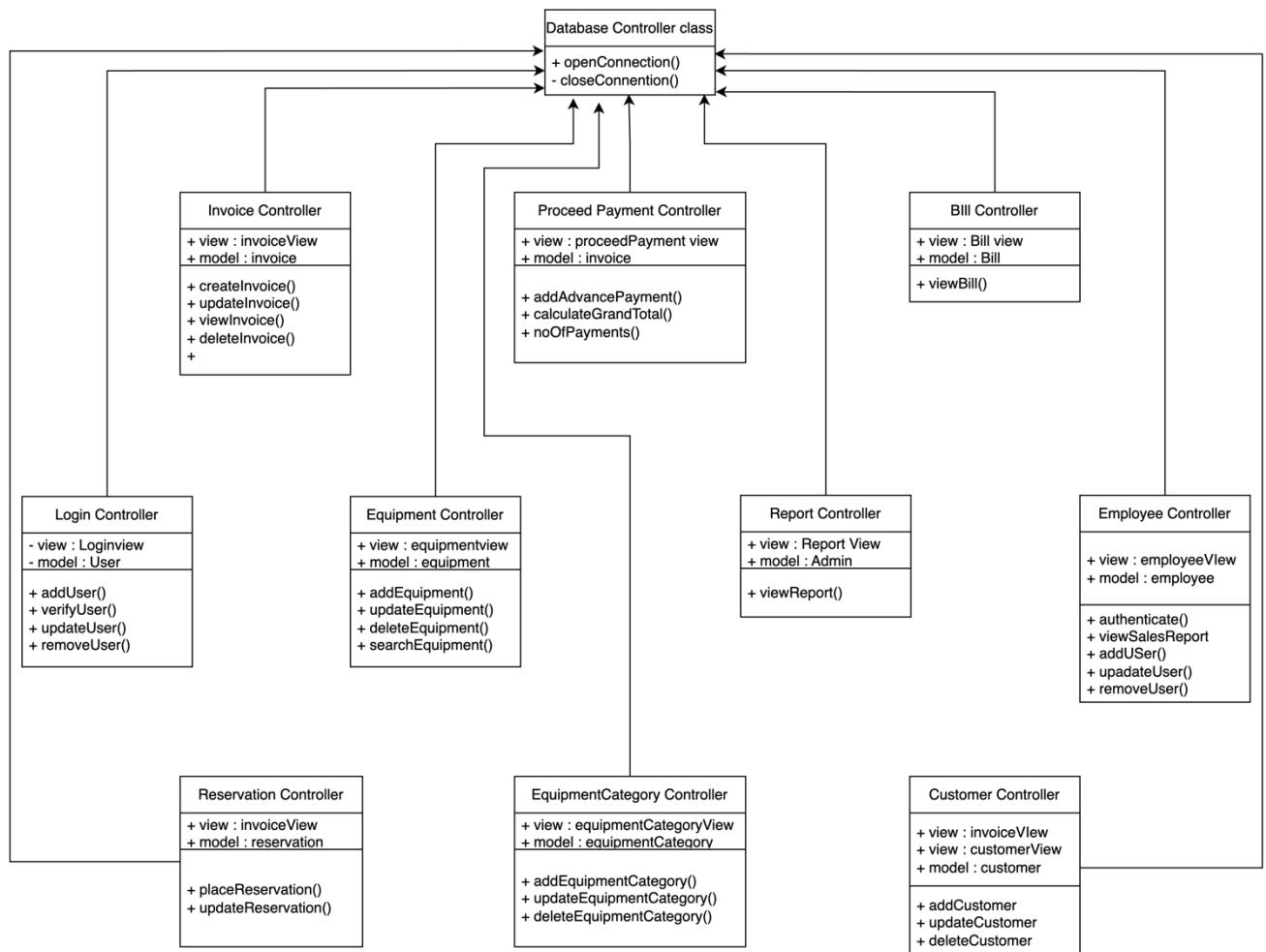
3.5.1 Entity-Class diagram



3.5.2 Interface-Class diagram



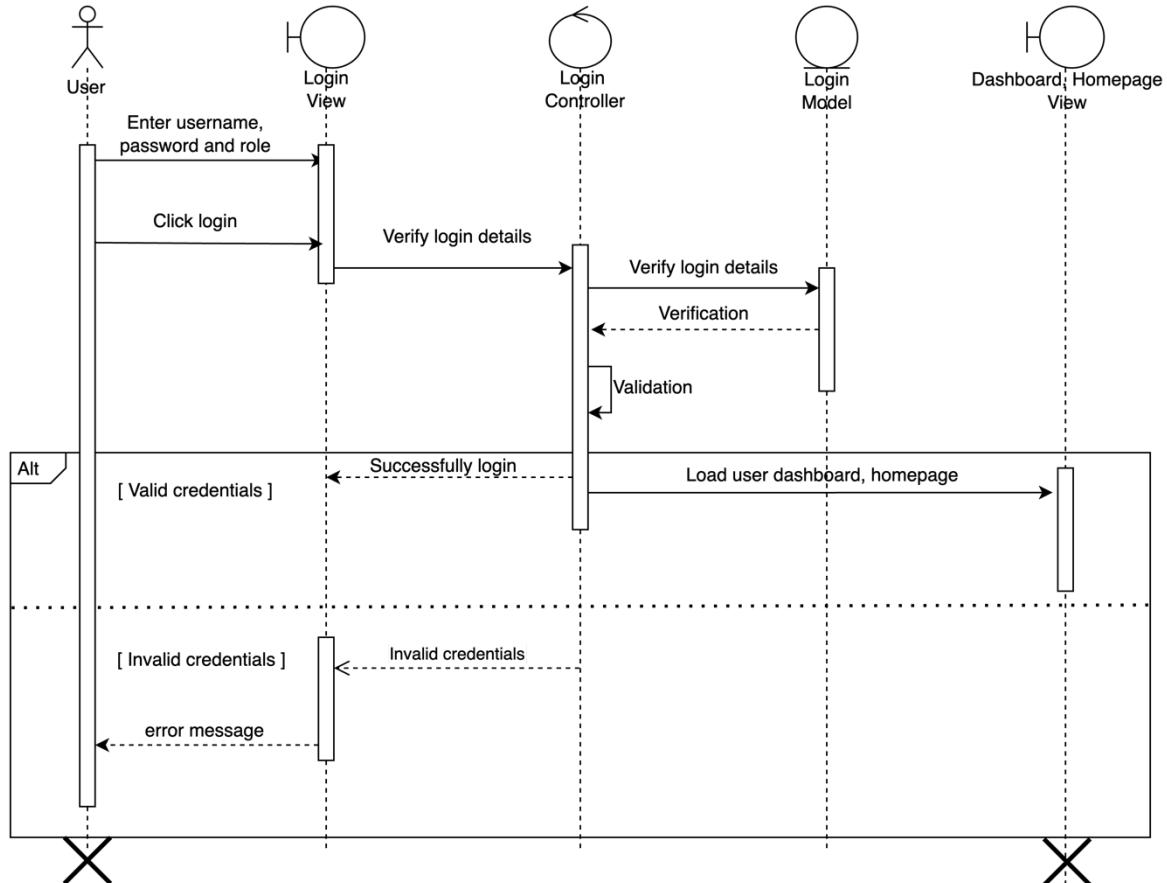
3.5.3 Controller-Class diagram



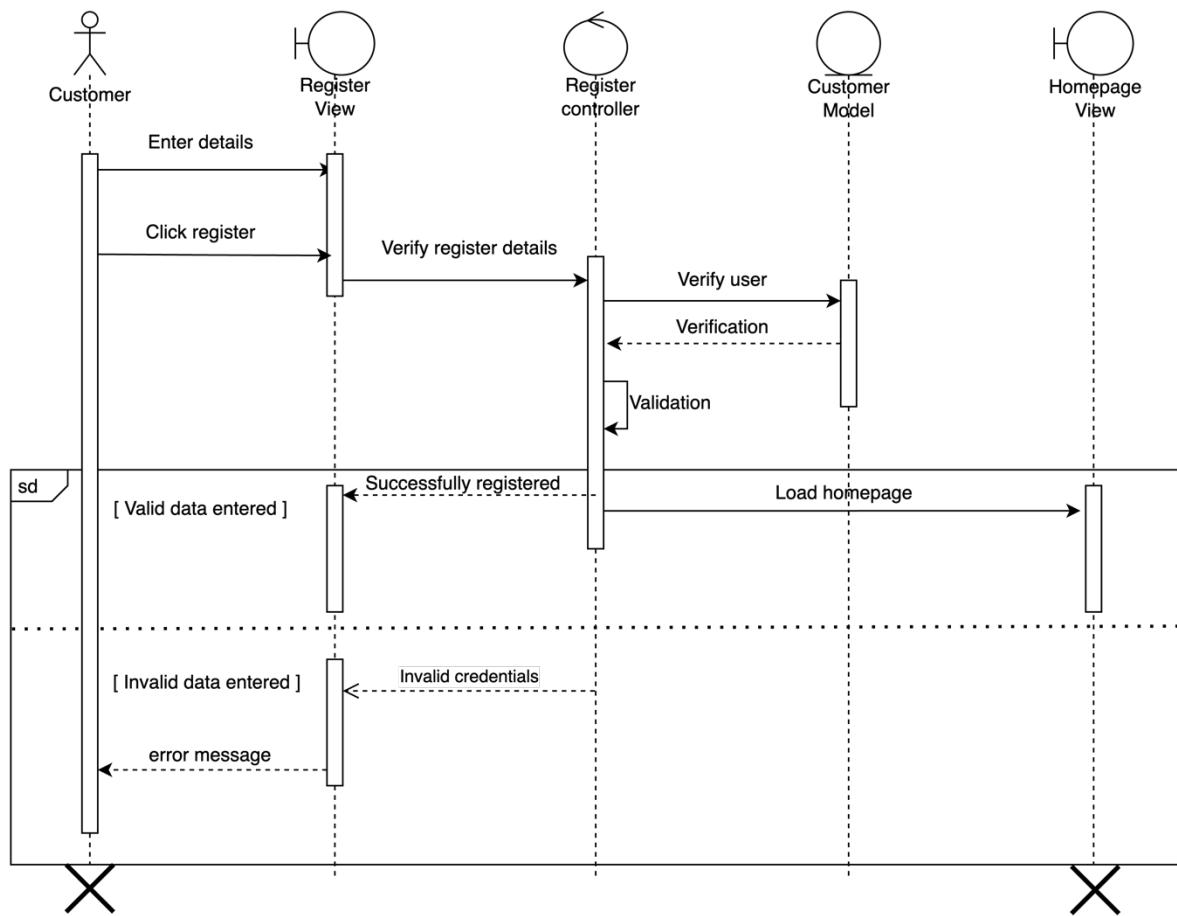
3.6 Sequence diagram

Within this segment, the sequence diagrams elucidate the interactions between the proposed system and both actors and objects, delineating the chronological sequence of the system's component engagements. Actors and objects are represented by horizontal lifelines, and the flow of data transactions is visually conveyed through these elements. The principal focus of these diagrams is to highlight the primary use cases of the system.

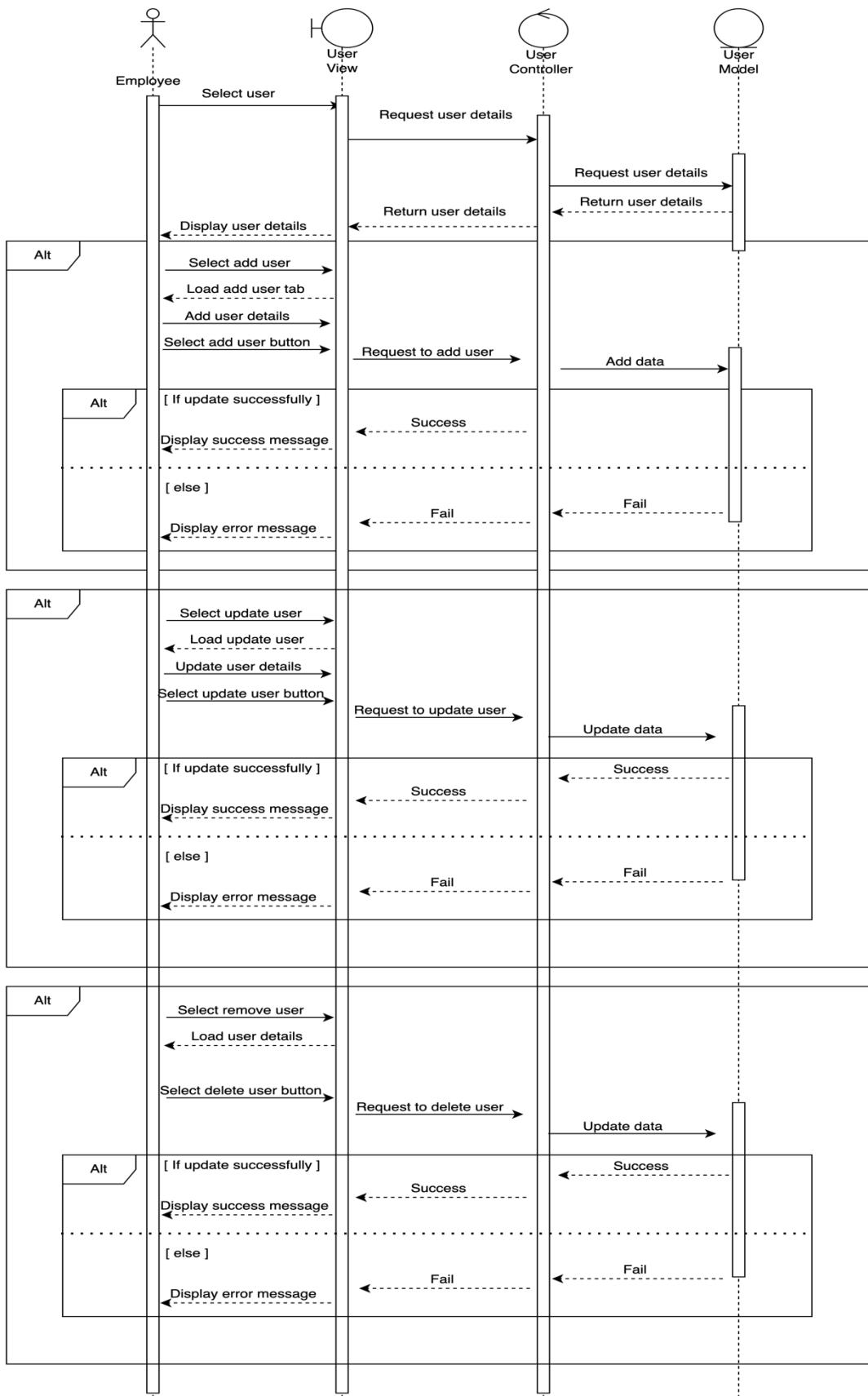
3.6.1 Sequence diagram for login



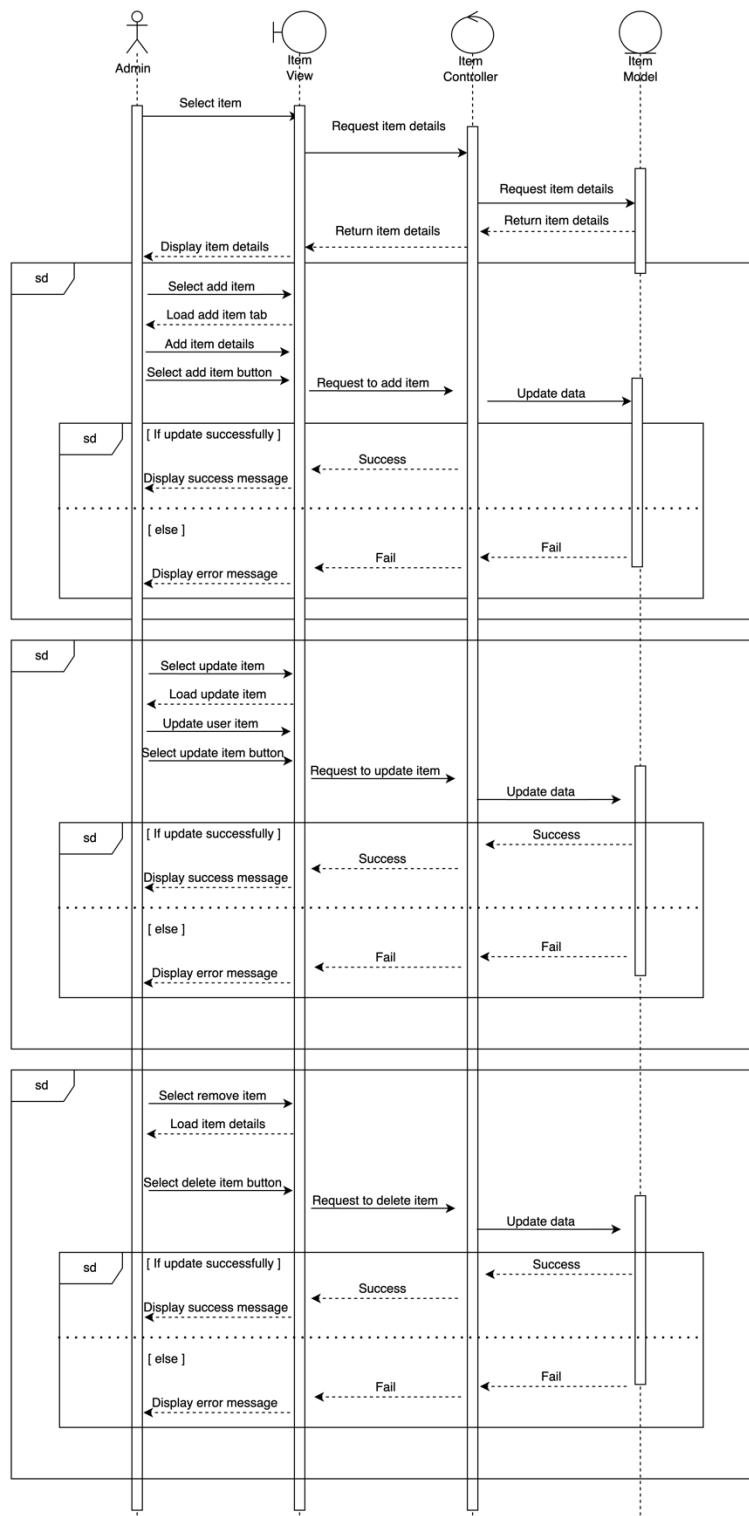
3.6.2 Customer register in the website to reserve equipment



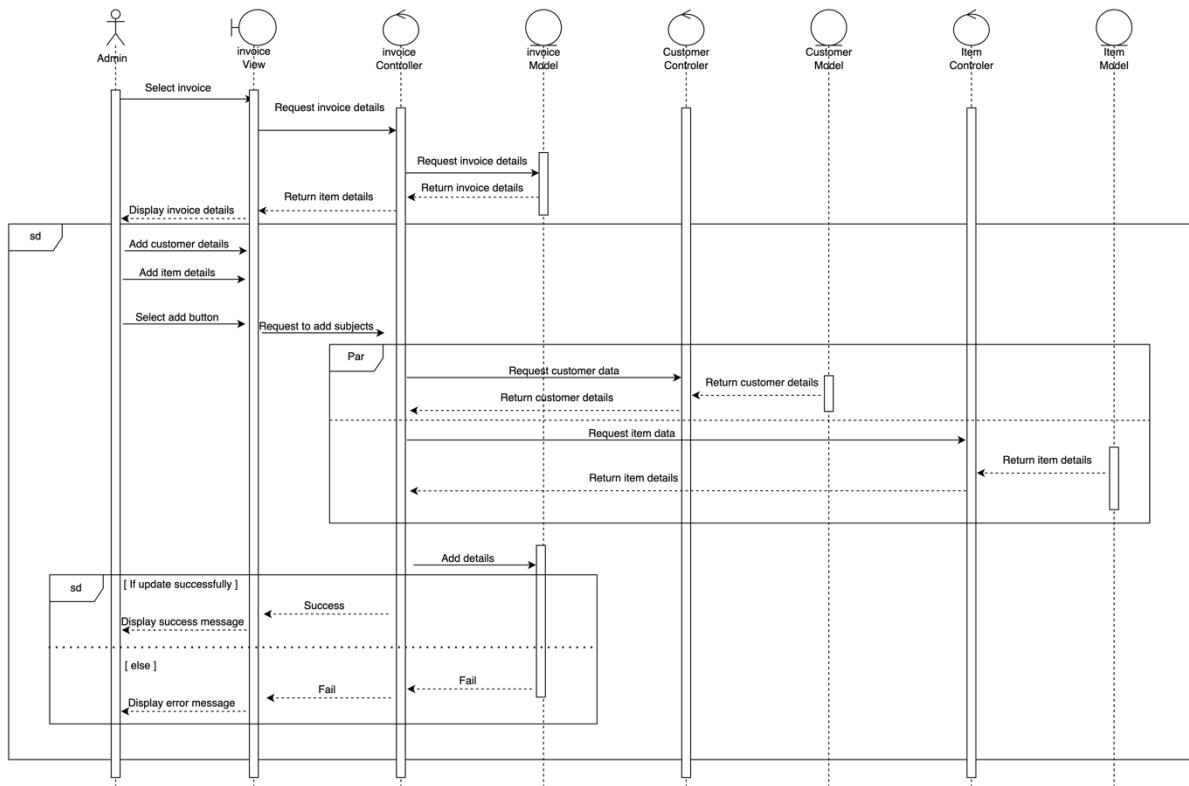
3.6.3 Manage users and customers



3.6.4 Manage equipment (items)



3.6.5 Manage Invoices



3.7 INTERFACES

3.7.1 Invoice page

Invoice ID: 41
11/06/2024, 22:34:18

Equipment Form

ID	Search with invoice Id	SEARCH	CREATE NEW
Name	Search with phone number or NIC	Q ADVANCE SEARCH CLEAR	
Quantity	remaining : No equipment found		
Remaining Stock: 0			
ADD		HANOVER	CLEAR

Payments

Payment Id	Payment	Switch

Advance Payments

Row number	equipment id	Machine name	Rental per day(LKR)	Handover Time period	Borrow Qty	Duration	Return Qty	Rental for the equipment

CREATE INVOICE

3.7.2 Invoice page after data retrieve

Invoice ID: 37
09/06/2024, 12:06:59

Equipment Form

ID	Search with invoice Id	SEARCH	CREATE NEW
Name	Search with phone number or NIC	Q ADVANCE SEARCH CLEAR	
Quantity	remaining : No equipment found		
Rem Borrowed Stock: 0			
HANOVER		CLEAR	

Payments

Payment Id	Payment	Switch
Advance payment	20000	
09/06/2024, 00:00:00	2000	

Advance Payments

Row number	equipment id	Machine name	Rental per day(LKR)	Handover Time period	Borrow Qty	Duration	Return Qty	Rental for the equipment
1#	44	Equipment 40	123.00	12/06/2024, 02:58:57	5	2	5	1230 LKR
2#	44	Equipment 40	123.00	12/06/2024, 02:58:57	5	2	5	1230 LKR

UPDATE INVOICE

3.7.3 Customer page

The screenshot shows a user interface for managing customers. On the left is a sidebar with navigation links: DASHBOARD, INVOICES, CUSTOMERS (selected), EQUIPMENT, REPORTS, and USER. At the top right is a search bar with placeholder text "Search by anything" and a dropdown menu with options: First Name, National ID card, Phone number, and Address Line 1. Below the search bar is a "SAVE" button and a "CLEAR" button. A cartoon illustration of a person using a smartphone is positioned between the search bar and the buttons. The main area displays a table of customer records:

Customer Id	Customer Name	Customer Phone number	Customer NIC	Customer Address line 1	Customer Address line 2
16	Mia Taylor	0770000016	852741963V	159 Maple Ave	Suite 1501
18	Harper White	0770000018	654258369V	741 Pine Ave	Apt 1701
22	Avery Robinson	0770000022	456123789V	258 Cedar St	Floor 20
23	Liam Wood	0770000023	654987321V	369 Elm Ave	Apt 2101
30	Luna King	0770000030	789654321V	852 Maple Ave	Unit G
37	Oliver Adams	0770000037	654258369V	963 Spruce Ave	Floor 32
41	Aiden Cooper	0770000041	456123789V	369 Elm Ave	Suite 3501
43	Avery Hunt	0770000043	789456123V	852 Maple Blvd	Apt 3701
48	Ethan Martin	0770000048	147258369V	369 Pine Ave	Apt 4101
..

3.7.4 Customer form validations

The screenshot shows a customer form with validation errors. The sidebar and search bar are identical to the previous screenshot. The validation message "⚠️ This phone number already exist" is displayed next to the phone number field. The table of customer records is also present.

Customer Id	Customer Name	Customer Phone number	Customer NIC	Customer Address line 1	Customer Address line 2
14	Isabella Anderson	0770000014	987123456V	456 Spruce Dr	Room 1401
15	Lucas Thomas	0770000015	321654987V	789 Oakwood Blvd	Unit D
16	Mia Taylor	0770000016	852741963V	159 Maple Ave	Suite 1501
17	Mason Jackson	0770000017	369147258V	369 Walnut St	Floor 16
18	Harper White	0770000018	654258369V	741 Pine Ave	Apt 1701
19	Ethan Harris	0770000019	789123456V	852 Cedar Blvd	Room 1801
20	Evelyn Martin	0770000020	123456789V	963 Spruce Dr	Unit E
21	Logan Thompson	0770000021	987654321V	147 Oakwood Blvd	Suite 1901
22	Avery Robinson	0770000022	456123789V	258 Cedar St	Floor 20
..

3.7.5 New Customer form

 admin
theekshana

first name
Theekshana

last name
Fernando

address line1
16/52 P.M Fernando 5th Lane

national ID card number
071-897-6568

phone number
071-897-6568

address line2
Moratuwa

First Name
 National ID card
 Phone number
 Address Line 1
⚠ This phone number already exist


SAVE
CLEAR

DASHBOARD
INVOICES
CUSTOMERS
EQUIPMENT
REPORTS
USER

Customer Id	Customer Name	Customer Phone number	Customer NIC	Customer Address line 1	Customer Address line 2
^ 14	Isabella Anderson	0770000014	987123456V	456 Spruce Dr	Room 1401
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">first name Isabella</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">last name Anderson</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">address line1 456 Spruce Dr</div> </div>					
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">national ID card number 987123456V</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">phone number 0770000014</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">address line2 Room 1401</div> </div>					
DELETE SAVE					

LOGOUT	^ 15	Lucas Thomas	0770000015	321654987V	789 Oakwood Blvd	Unit D
	^ 16	Mia Taylor	0770000016	852741963V	159 Maple Ave	Suite 1501
	^ 17	Mason Jackson	0770000017	369147258V	369 Walnut St	Floor 16
	^ 18	Harper White	0770000018	654258369V	741 Pine Ave	Apt 1701
	^ 19	Ethan Harris	0770000019	789123456V	852 Cedar Blvd	Room 1801
	^ 20	Evelyn Martin	0770000020	123456789V	963 Spruce Dr	Unit E

3.7.6 Equipment page

 admin
theekshana

Machine Name

Date of Purchase

UPLOAD AN IMAGE

Warranty Expiration Date

Machine Cost

Rental

Description

Category

Complete Stock

SUBMIT
CLEAR

DASHBOARD
INVOICES
CUSTOMERS
EQUIPMENT
REPORTS
USER

Id	Machine Name	Category Name	Rental	DOP	Warranty Due	Machine Cost	Machine Image	Description	Defective status	Stock remaining
^ 14	Equipment 14	Power tools	283.94	2024-04-14	2024-04-14	1400.00	VIEW	0	1	2
^ 16	Equipment 16	Power tools	1309.74	2024-05-09	2024-04-14	1600.00	VIEW	0	1	1
^ 18	Equipment 18	Power tools	729.67	2024-04-08	2024-04-08	1800.00	VIEW	0	0	12
^ 19	Equipment 19	Power tools	1613.01	2024-04-11	2024-04-11	1900.00	VIEW	0	0	2
^ 20	Equipment 20	Power tools	276.05	2024-04-08	2024-04-08	2000.00	VIEW	0	0	10

3.7.7 Equipment Form



admin
theekshana

- [DASHBOARD](#)
- [INVOICES](#)
- [CUSTOMERS](#)
- [EQUIPMENT](#)
- [REPORTS](#)
- [USER](#)

Id	Machine Name	Category Name	Rental	DOP	Warranty Due	Machine Cost	Machine image	Description	Defective status	Stock remaining
▼ 14	Equipment 14	Power tools	283.94	2024-04-14	2024-04-14	1400.00	VIEW	0	1	2
▲ 16	Equipment 16	Power tools	1309.74	2024-05-09	2024-04-14	1600.00	VIEW	0	1	1

Machine Name:

Date of Purchase:

Warranty Expiration Date:

Machine Cost:

Rental:

Description:

Category:

Complete Stock:

[SUBMIT](#)
[CLEAR](#)
[DELETE](#)

Id	Machine Name	Category Name	Rental	DOP	Warranty Due	Machine Cost	Machine image	Description	Defective status	Stock remaining
▼ 18	Equipment 18	Power tools	729.67	2024-04-08	2024-04-08	1800.00	VIEW	0	0	12
▼ 19	Equipment 19	Power tools	1613.01	2024-04-11	2024-04-11	1900.00	VIEW	0	0	2
▼ 20	Equipment 20	Power tools	276.05	2024-04-08	2024-04-08	2000.00	VIEW	0	0	10
▼ 24	Drill	Power tools	250.00	2024-05-14	2024-05-04	2000.00	VIEW	good drill	0	6
▼ 25	Drill	Power tools	250.00	2024-05-14	2024-05-04	2000.00	VIEW	good drill	0	1
▼ 26	scaffoldins	Power tools	150.00	2024-04-20	2024-05-06	2500.00	VIEW	abc	5	132
▼ 40	scaffoldins	Power tools	150.00	2024-04-30	2024-05-16	2500.00	VIEW	f	1	150
▼ 41	Drill	Power tools	250.00	2024-05-12	2024-05-02	2000.00	VIEW	good drill	1	1

3.7.8 Equipment report 2



admin
theekshana

- [DASHBOARD](#)
- [INVOICES](#)
- [CUSTOMERS](#)
- [EQUIPMENT](#)
- [REPORTS](#)
- [USER](#)

Equipment
Invoices
Cusotmers

Equipment Reports

- [EQUIPMENT UTILIZATION REPORT](#)
- [EQUIPMENT REVENUE REPORT](#)
- [UNDER UTILIZED EQUIPMENT REPORT](#)
- [EQUIPMENT MAINTENANCE NEEDS REPORT](#)
- [INCOMPLETE RENTALS BY EQUIPMENT](#)

Start Date	<input type="button" value=""/>	End Date	<input type="button" value=""/>
Equipment ID		Equipment Name	Total Revenue
26		scaffoldins	724800.00
44		Equipment 40	71586.00
24		Drill	16000.00
25		Drill	2250.00

3.7.9 Equipment report 3

The screenshot shows a software application interface for managing equipment rentals. At the top, there is a header bar with three tabs: 'Equipment' (highlighted in purple), 'Invoices', and 'Customers'. On the left, a sidebar menu includes 'DASHBOARD', 'INVOICES', 'CUSTOMERS', 'EQUIPMENT', 'REPORTS' (which is selected and highlighted in purple), and 'USER'. Below the sidebar, a 'LOGOUT' button is visible. The main content area displays a table titled 'Equipment Reports' under the 'UNDER UTILIZED EQUIPMENT REPORT' section. The table has columns for 'Equipment ID', 'Equipment Name', and 'Total Rentals'. The data shows two entries: one for Equipment ID 25 (Drill) with 1 rental, and another for Equipment ID 24 (Drill) with 2 rentals.

Equipment ID	Equipment Name	Total Rentals
25	Drill	1
24	Drill	2

3.7.10 Equipment report 4

The screenshot shows a similar software application interface for managing equipment rentals, specifically report 4. The layout is identical to report 3, with a purple-highlighted 'Equipment' tab in the header and a purple-highlighted 'REPORTS' button in the sidebar. The main content area displays a table titled 'Equipment Reports' under the 'EQUIPMENT MAINTENANCE NEEDS REPORT' section. This table includes additional columns: 'Start Date' and 'End Date' for filtering, and 'Total Rentals' and 'Total Rental Days' for each equipment entry. The data shows four entries: one for Equipment ID 26 (scaffolding) with 19 rentals over 186 days, one for Equipment ID 24 (Drill) with 2 rentals over 16 days, one for Equipment ID 44 (Equipment 40) with 5 rentals over 12 days, and one for Equipment ID 25 (Drill) with 1 rental over 9 days.

Equipment ID	Equipment Name	Total Rentals	Total Rental Days
26	scaffolding	19	186
24	Drill	2	16
44	Equipment 40	5	12
25	Drill	1	9

3.7.11 Equipment report 5

The screenshot shows a software application window titled 'admin theekshana'. On the left is a sidebar with icons for Dashboard, Invoices, Customers, Equipment, Reports (which is selected and highlighted in purple), and User. Below the sidebar are Logout and Help buttons. The main area has three tabs at the top: Equipment, Invoices, and Customers. The Equipment tab is active and displays a table titled 'Equipment Reports' with the following columns: Equipment ID, Equipment Name, Invoice ID, Invoice Created Date, Duration in Days, and Not Completed. The data in the table is as follows:

Equipment ID	Equipment Name	Invoice ID	Invoice Created Date	Duration in Days	Not Completed
44	Equipment 40	36	06/06/2024 13:39	5	0
44	Equipment 40	37	09/06/2024 12:06	2	0
26	scaffoldins	38	10/06/2024 12:29	1	0
26	scaffoldins	39	10/06/2024 14:49	1	0
26	scaffoldins	40	10/06/2024 15:35	1	0

3.7.12 Invoice report 1

The screenshot shows a software application window titled 'admin theekshana'. On the left is a sidebar with icons for Dashboard, Invoices, Customers, Equipment, Reports (selected and highlighted in purple), and User. Below the sidebar are Logout and Help buttons. The main area has three tabs at the top: Equipment, Invoices, and Customers. The Invoices tab is active and displays a table titled 'Invoice Reports' with the following columns: First Name, Last Name, Invoice ID, Created Date, and Updated Status. The data in the table is as follows:

First Name	Last Name	Invoice ID	Created Date	Updated Status
Michael	Wright	1	2024-05-29 15:55:34	1

3.7.13 Invoice report 2

The screenshot shows a dashboard interface with a sidebar on the left containing user information ('admin theekshana') and navigation links for Dashboard, Invoices, Customers, Equipment, Reports, and User, along with a Logout button. The main content area has three tabs at the top: Equipment, Invoices (selected), and Customers. The Invoices tab displays a table of invoices with columns: Month, Invoice Count, Total Revenue, Customer Name, Invoice ID, Created Date, Updated Status, and Invoice Total Amount. The table contains 10 rows of data. At the bottom right of the table, there are buttons for 'Rows per page' (set to 10), '1-10 of 10', and navigation arrows.

Month	Invoice Count	Total Revenue	Customer Name	Invoice ID	Created Date	Updated Status	Invoice Total Amount
2024-06	1	37500	Theekshana Fernando	26	2024-06-02 13:37:43	1	141000.00
2024-06	1		Theekshana Fernando	28	2024-06-02 14:12:25	1	177450.00
2024-06	1	28000	theekshana fernandop	32	2024-06-04 21:14:13	1	648000.00
2024-06	1	20000	theekshana fernandop	33	2024-06-05 10:53:08	1	266200.00
2024-06	1		Mia Taylor	34	2024-06-06 11:46:35	1	10332.00
2024-06	1	5000	Harper White	36	2024-06-06 13:39:04	1	61500.00
2024-06	1	3000	Theekshana Fernando	35	2024-06-06 15:37:21	1	40500.00
2024-06	1	2000	Charlotte Lewis	37	2024-06-09 12:08:59	1	1845.00
2024-06	1		Amelia Walker	38	2024-06-10 12:29:59	1	1000.00
2024-06	1	10100	Charlotte Lewis	39	2024-06-10 14:49:11	1	0.00

3.7.14 Customer report 1

The screenshot shows a dashboard interface with a sidebar on the left containing user information ('admin theekshana') and navigation links for Dashboard, Invoices, Customers, Equipment, Reports (selected), and User, along with a Logout button. The main content area has three tabs at the top: Equipment, Invoices, and Customers. The Invoices tab displays a table of customer sales with columns: Customer Name, Customer Phone Number, Average Rating, Number of Invoices, and Total Sales. The table contains 6 rows of data. The last row, 'theekshana fernandop', is highlighted in red. At the bottom right of the table, there are buttons for 'Rows per page' (set to 10), '1-10 of 10', and navigation arrows.

Customer Name	Customer Phone Number	Average Rating	Number of Invoices	Total Sales ▲
Mia Taylor	0770000016	N/A	1	738.00
Harper White	0770000018	N/A	1	246.00
Charlotte Lewis	0770000024	N/A	2	246.00
Amelia Walker	0770000026	☆☆☆☆☆	1	250.00
Theekshana Fernando	0782367100	★★★★★	11	17850.00
theekshana fernandop	0718976568	N/A	6	15600.00

3.7.15 Customer report 2

The screenshot shows a dashboard interface with a sidebar on the left containing user profile, navigation links (Dashboard, Invoices, Customers, Equipment, Reports, User), and a Logout button. The main area features three tabs: Equipment, Invoices, and Customers. The Invoices tab is active, displaying a table titled 'Customer Reports' with columns: Customer Name, Customer Phone Number, Invoice Id, Invoice Rating, Special Message, Invoice Created Date, and Total amount. The table lists several entries, with the last two rows highlighted in yellow.

Customer Name	Customer Phone Number	Invoice Id	Invoice Rating	Special Message	Invoice Created Date	Total amount
Theekshana Fernando	0782367100	26	★★★★★		2024-06-02 13:37:43	7350.00
Theekshana Fernando	0782367100	28	★★★★★		2024-06-02 14:12:25	6450.00
theekshana fernandop	0718976568	32	N/A		2024-06-04 21:14:13	4350.00
theekshana fernandop	0718976568	33	N/A		2024-06-05 10:53:08	11250.00
Mia Taylor	0770000016	34	N/A		2024-06-06 11:46:35	738.00
Theekshana Fernando	0782367100	35	★★★★★	The machine got defected after	2024-06-06 15:37:21	4050.00
Harper White	0770000018	36	N/A		2024-06-06 13:39:04	369.00
Charlotte Lewis	0770000024	37	N/A		2024-06-09 12:06:59	369.00

3.7.16 Add user page

The screenshot shows a dashboard interface with a sidebar on the left containing user profile, navigation links (Dashboard, Invoices, Customers, Equipment, Reports, User), and a Logout button. The main area features a form titled 'Add New User' with fields for First name, Last name, User name, Password, Confirm password, User role, User NIC number, User phone number, User address Line 1, and User address Line 2. Below the form is a table listing existing users with columns: Full name, Username, Phone number, Address, Roles, and Grant permissions. At the bottom, there is a grid of five user profiles with their names below them.

Full name	Username	Phone number	Address	Roles	Grant permissions
Theekshana fernando	theekshana	0718976568	52/16 p.m Fernando, 5th lane moratuwa	Admin <input checked="" type="checkbox"/> Cashier <input checked="" type="checkbox"/>	
Achila dilshan	achila	0342425132	23-A kiribathkumbura kandy	Warehouse Handler <input checked="" type="checkbox"/> Admin <input checked="" type="checkbox"/>	
Shehan charudith	shehan	0784524542	23h-2 rathnapura	Admin <input checked="" type="checkbox"/>	
Umesh Silva	umesha	0784322546	68-A M.J.C Fernando Lane Idama,Moratuwa	Admin <input checked="" type="checkbox"/>	
Zafry Mubarak	zafry	0718976568	16/52 P.M Fernando 5th Lane Moratuwa	<input checked="" type="checkbox"/>	

Theekshana fernando

Achila dilshan

Shehan charudith

Umesh Silva

Zafry Mubarak

3.7.17 Printable Invoice pdf

INVOICE

Invoice id : 37
Date : 09/06/2024
Time : 12:06:59

Invoice to :
Charlotte Lewis

📞 0770000024
📍 741 Maple Blvd
Room 2201

Equipment ID	Equipment Name	Rental	Return Date	Borrow Qty	Duration (days)	Return Qty	Total (LKR)
1 44	Equipment 40	123.00	out	10		0	0 LKR
2 44	Equipment 40	123.00	12/06/2024, 02:58:57	5	2	5	1230 LKR

SubTotal 246 LKR

Advance 20000 LKR

Payments 2000 LKR

PDF

4.SYSTEM DEVELOPMENT

The chapter gives a brief overview of the programming language features and technologies used to implement the Tennis Club Management System. The third-party components, APIs, and libraries used are also briefly described in this chapter.

Outline of the Chapter

- 4.1 Programming Languages
- 4.2 Development Tools and Technologies
- 4.3 Third-Party Components/Libraries & Frameworks
- 4.4 Summary

4.1 Programming Languages

Building a dynamic website is a complex task that requires significant time and effort to code each component. The web-based machinery rental management system, GearGrid, requires simultaneous access and real-time data tracking to facilitate data transfer through various access levels. The accessibility of the software and the effective use of current hardware/software were key factors in choosing the development technologies and programming languages. A programming language is a formal language that consists of a set of instructions that can generate different types of output. Implementing computer systems and algorithms requires the use of programming languages. Computer instructions make up the majority of programming languages.

4.1.1 Hyper Text Markup Language (HTML)

Hyper Text Markup Language (HTML) is a markup language used for creating and formatting content on the World Wide Web. It is the standard language for building web pages and is used in conjunction with Cascading Style Sheets (CSS) and JavaScript to create modern, interactive web applications and sites.

4.1.2 Cascading Style Sheets (CSS)

CSS stands for Cascading Style Sheets, which is a style sheet language used to define the presentation and layout of web pages. CSS works in conjunction with HTML and JavaScript to create modern, interactive web applications and sites. CSS allows web designers and developers to define the visual style and layout of a web page separately from its content. This makes it easier to create and manage the look and feel of a site, as changes to the CSS file will automatically be applied to all pages that reference it.

4.1.3 JavaScript (JS)

JavaScript is a high-level programming language used to create dynamic and interactive web applications. It can be used to manipulate web page content, handle user input, perform calculations, and communicate with servers. Two popular libraries for making HTTP requests in JavaScript are Axios and jQuery. Axios is a lightweight HTTP client that is widely used in modern web applications. It allows developers to easily make HTTP requests and handle responses.

4.1.4 Structured Query Language (SQL)

Structured Query Language (SQL) is a standard language for managing and manipulating relational databases. SQL is used to perform various operations on the data stored in a database, such as querying, updating, inserting, and deleting data. It also allows for the creation and modification of database structures like tables, indexes, and views. SQL is essential for managing the data of the GearGrid system, ensuring efficient and secure storage, retrieval, and manipulation of transactional and customer data.

4.2 Development Tools and Technologies

Web development tools (programmers) allow web developers to test and debug their code and allow developers to work with multiple web technologies.

4.2.1 Visual Studio Code

Visual Studio Code is a free, open-source code editor developed by Microsoft. It is a crossplatform editor that runs on Windows, macOS, and Linux, and it is used by developers for a wide range of programming languages and frameworks. Visual Studio Code is a popular choice among developers for its lightweight and flexible nature, as well as its powerful features and extensibility.

4.2.2 Version Control System (Git)

Git is a distributed version control system that allows developers to track changes to code and collaborate with other developers on a project. GitHub, on the other hand, is a web-based platform that provides hosting for Git repositories, as well as collaboration tools such as pull requests, issues, and project boards. In summary, Git is a powerful tool for version control and collaboration, while GitHub is a popular platform for hosting and collaborating on Git repositories.

4.2.3 Database Management System (MySQL)

MySQL is a popular open-source relational database management system (RDBMS) that is widely used for web applications. It is designed to be fast, reliable, and scalable, and it supports a wide range of platforms and programming languages. MySQL is widely used in web applications and is supported by many hosting providers and content management systems.

4.3 Third-Party Components/Libraries and Frameworks

4.3.1 Bootstrap

Bootstrap is a popular open-source front-end web development framework that provides a set of pre-designed HTML, CSS, and JavaScript components and utilities. Some key features of Bootstrap include, responsiveness, pre-designed components, highly customizable, cross browser compatibility, accessibility, JavaScript plugins. Bootstrap is widely used by web developers for its ease of use, flexibility, and wide range of features. It has become a standard for building responsive, mobile-first websites and web applications.

4.3.2 cors

cors is a Node.js package for providing a Connect/Express middleware that can be used to enable Cross-Origin Resource Sharing (CORS) with various options. It helps in making cross-domain requests and handling security issues related to it.

4.3.3 dayjs

dayjs is a minimalist JavaScript library that parses, validates, manipulates, and displays dates and times. It is a lightweight alternative to Moment.js with a similar API, enabling developers to handle date and time operations with ease.

4.3.4 dotenv

dotenv is a zero-dependency module that loads environment variables from a .env file into process.env. It is useful for managing configuration settings for different environments (development, testing, production).

4.3.5 express

express is a fast, unopinionated, and minimalist web framework for Node.js. It provides a robust set of features for web and mobile applications, enabling developers to build single-page, multi-page, and hybrid web applications efficiently.

4.3.6 jsonwebtoken

jsonwebtoken is a library used to create and verify JSON Web Tokens (JWT). JWTs are used for securely transmitting information between parties as a JSON object and are commonly used for authentication and authorization in web applications.

4.3.7 mysql2

mysql2 is a modern and fast MySQL client for Node.js. It supports both callback and promise APIs and is built on top of the MySQL C API, providing a robust solution for interacting with MySQL databases.

4.3.8 nodemon

nodemon is a tool that helps develop Node.js-based applications by automatically restarting the application when file changes in the directory are detected. It increases development productivity by removing the need to manually restart the server.

4.3.9 chart.js

chart.js is a simple yet flexible JavaScript library for creating engaging and responsive charts. It supports various chart types, including line, bar, radar, doughnut, pie, polar area, and bubble charts, and offers customization options for detailed visualizations.

4.3.10 firebase

firebase is a comprehensive app development platform provided by Google. It includes tools and infrastructure to build high-quality apps, including authentication, real-time databases, cloud storage, and hosting, making it easier to develop scalable web and mobile applications.

4.3.11 html2canvas

html2canvas is a library that allows you to take "screenshots" of web pages or parts of them directly from the browser. It renders the web page as a canvas image, enabling the capture of the current state of a DOM element and its children.

4.3.12 jspdf

jspdf is a library for generating PDF documents in JavaScript. It allows developers to create and customize PDFs directly from the browser or server-side environments, offering features like text, shapes, images, and even HTML rendering.

4.3.13 notistack

notistack is a highly customizable notification library for React applications. It allows developers to stack notifications (toasts) on top of each other and provides a rich API for managing notification behavior, appearance, and interactions.

4.3.14 MUI (Material-UI)

MUI (Material-UI) is a popular React UI framework that implements Google's Material Design principles. It provides a comprehensive suite of pre-built components and themes that help developers build consistent, aesthetically pleasing, and responsive user interfaces. Key features include:

- **Customizable Themes:** Easily adapt the look and feel of components using customizable themes.
- **Responsive Design:** Components that are optimized for all screen sizes.
- **Comprehensive Component Library:** Access to a wide range of components such as buttons, forms, tables, dialogs, and more.
- **Accessibility:** Built-in accessibility features to ensure components are usable by all users.

4.3.15 Axios

is a popular JavaScript library used for making HTTP requests from browsers or Node.js. It provides a simple and intuitive API for sending asynchronous HTTP requests and handling responses. With features like automatic JSON data parsing and support for interceptors, Axios simplifies the process of working with APIs in web development.

4.4 Summary

In this Chapter, used technologies such as programming languages, frameworks, and databases are discussed. Moreover cost, performance, functionality, and ease of use have been taken into account when choosing the correct technologies. Constraints may include hardware limitations, compatibility issues, and legal or regulatory requirements. Third-party support also included such as using external APIs or libraries.

5. Testing

This chapter is dedicated to the critical process of verifying and validating the system developed in the preceding chapters. Testing is an integral part of any software development project. It is through testing that the quality of the software is assured, potential errors are identified, and system requirements are verified. The importance of testing cannot be overstated as it directly impacts the success of the software in fulfilling its intended purpose.

In this project, testing was not an afterthought or a final step in the development process. Instead, it was incorporated into every stage of the project, from analysis and design to development. This approach, often referred to as ‘continuous testing,’ allows for early detection of issues and facilitates their timely resolution. This chapter includes details on the types of tests performed, such as unit tests, integration tests, and system tests, among others.

5.1 White Box Testing

White box testing was used to examine the internal logic and structure of the code. Also known as clear box testing or glass box testing, this method allowed for testing individual parts of the code, such as functions, loops, and conditional statements. White box testing was employed to verify the correctness of the code and to uncover any hidden errors that might not have been detected during black box testing. By examining the internal workings of the software, issues related to logic, syntax, and implementation were identified and fixed.

5.1.1 Unit Testing

Unit testing is a fundamental practice in white box testing. It involves examining individual components of the software in isolation to ensure they function correctly. This includes testing individual functions, methods, procedures, or objects. This is crucial as it helps to detect and prevent bugs in the early stages, making the debugging process easier. It also helps in maintaining and changing the code, enhancing the overall quality of the code.

For instance:

1. **Account Creation Function:** A unit test was conducted to ensure that the function responsible for creating user accounts worked correctly. This involved providing valid user information to the function and verifying that an account was successfully created.
2. **Equipment Inventory Display Function:** A unit test was conducted to ensure that the function responsible for displaying equipment details worked correctly. This involved calling the function and verifying that it returned the correct equipment details according to equipment selection.

These are just a few examples of unit tests that were performed as part of white box testing. By examining the internal workings of these functions, it was ensured they were working correctly and efficiently.

5.2 Black Box Testing

Black box testing was a key component of the testing strategy. This method of testing focuses on the functionality of the software without considering its internal structure or workings. The system was treated as a ‘black box,’ and tests were conducted based on the input and the output alone.

Black box testing was utilized to validate that the system met the specified requirements. It ensured that all the functions of the software were working as expected. This type of testing was particularly useful in identifying any discrepancies between the software’s intended behavior and its actual output.

Examples of black box testing include:

1. **User Registration Process:** The registration form was tested to ensure that users could successfully create an account with valid information and receive appropriate error messages with invalid information.
2. **Invoice Generation:** The process of generating and displaying invoices was tested to ensure accuracy and completeness, including calculations and formatting.
3. **Search Functionality:** The search feature was tested to ensure that it returned the correct results for various queries and handled invalid queries gracefully.

By employing both white box and black box testing methods, the project ensured a comprehensive validation of the system, leading to a reliable and robust web-based application for T.A Enterprises.

5.2.1 Functional Testing

Functional testing was conducted to verify that each function of the software application operated in conformance with the requirement specification. This type of testing mainly involved feeding input and examining the output. Functional testing ensured that the functional requirements were properly satisfied by the application.

Integration Testing

Integration testing in GearGrid involves ensuring that various components of the system work together seamlessly. Here are some examples of integration tests performed on the system:

- User Registration and Login

Test Description: After creating a new user account, an integration test involves attempting to log in with the newly created account.

Purpose: This tests the integration between the user registration and login functionalities, ensuring that a newly registered user can log in without issues.

- Equipment Addition and Display

Test Description: Adding a new piece of equipment using the equipment management interface and then verifying if it appears correctly in the equipment inventory.

Purpose: This tests the integration between the equipment addition functionality and the inventory display functionality.

- Inventory Update and Display

Test Description: Updating the inventory status of a piece of equipment (e.g., marking it as rented) and then verifying if this change is reflected in all relevant displays.

Purpose: This tests the integration between inventory management and the display functionalities, ensuring real-time updates are visible to users.

- Payment Processing and Invoice Generation

Test Description: Processing a payment for a rental and then verifying if the invoice is correctly generated and displayed to the user.

Purpose: This tests the integration between payment processing, invoice generation, and user notification functionalities.

By performing these integration tests, you can ensure that the components of GearGrid function together correctly, providing a seamless and reliable user experience.

Interface Testing

Interface testing was a crucial component of the integration testing process. This testing type was centered around ensuring that the interactions between various software systems were executed as anticipated. Here are some examples of the interface tests conducted:

1. Workflow Testing:

Workflow testing involved checking the end-to-end functionality of the application to validate proper data transfer. The aim was to ascertain if the software workflow aligned with the business process, which included executing a sequence of steps to determine if the information/data workflow was correct.

An instance of Workflow testing was validating the transition from placing an order to cancelling it on the application. This process verifies the flow between two functionalities using accurate data. Negative data sets were also used to test if the workflow rejects incorrect information.

2. Database Interface Testing:

The interface of the system with its database was tested to ensure accurate storage and retrieval of data. For instance, when an order was placed, the order details were correctly stored in the database. When an order cancellation request was made, the relevant information was accurately updated in the database.

These tests confirmed that all interfaces within the system operated correctly, offering a smooth user experience. It was essential to check the interfaces to ensure that the servers were

functioning correctly and had appropriate error handling capabilities, along with providing an error message option for queries made by the application.

System Testing

System Testing is a level of functional testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems.

5.2.2 Non-Functional Testing

Non-functional testing was carried out to evaluate the system's performance under a particular workload, its reliability, scalability, and other non-functional parameters. This type of testing was designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. However, due to time constraints, not much emphasis could be placed on this aspect of testing in this project.

The following types of non-functional tests were planned to be conducted in the future to ensure that the non-functional requirements that were identified during the analysis stage were met:

- Performance Testing: This test was planned to verify if the system could process a minimum of 1000 transactions per minute, with a maximum response time of 3 seconds per transaction.
- Load Testing: This test was intended to ensure that the system could support at least 500 concurrent users without exceeding a specified response time threshold.
- Security Testing: This test was designed to confirm that only authenticated administrators could access the back-end servers and that all transactions involving any confidential customer information were encrypted using SSL (secured socket layer). It would also verify that all sensitive data was encrypted before being transmitted over insecure connections.
- Reliability Testing: This test was planned to ensure that the system had a minimum uptime of 99.9% per month, measured over a period of at least one year.
- Compatibility Testing: This test was intended to confirm that the system was compatible with multiple web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge.
- Usability Testing: This test was designed to ensure that the system was responsive and worked well on a range of devices, including desktops, smartphones, laptops, and tablets running on Windows 10 or upwards or iOS 11 or upwards.
- Other tests like Maintainability Testing, Scalability Testing, Volume Testing, Portability Testing, and Efficiency Testing were also planned to be conducted in the future. These tests would provide a comprehensive evaluation of the system's non-functional aspects.

In conclusion, both black box and white box testing played crucial roles in this project. They complemented each other by providing a comprehensive evaluation of both the functionality and the internal structure of the software.

All the tests mentioned above, including unit, integration, system, and interface testing, were performed manually. This hands-on approach allowed for a thorough and detailed examination of the system's functionality and performance.

Acceptance testing should ideally be conducted before the implementation phase. Acceptance testing is a type of testing performed to engage end users to determine whether a system satisfies the specified requirements. It serves as a final verification before the system is released into the production environment.

Non-functional testing is also of paramount importance and should be properly conducted in the future before implementation. It ensures that the system can handle the expected load, has acceptable performance and response times, and provides a satisfactory user experience.

5.3 Summary

In conclusion, both black box and white box testing played crucial roles in this project. They complemented each other by providing a comprehensive evaluation of both the functionality and the internal structure of the software. This comprehensive approach to testing ensured that all aspects of the system were thoroughly evaluated, and any issues were identified and addressed before release.

6. IMPLEMENTATION

The “Implementation” chapter provides a comprehensive overview of the various processes involved in setting up and deploying the web application, which includes a website and an admin panel. It details the server requirements and setup, software and hardware considerations, installation guide, data migration strategies, and training requirements. The chapter underscores the importance of each of these aspects in ensuring a successful implementation of the system.

6.1 Installation

The installation process for the web application, which includes a website and an admin panel, involves setting up the server and ensuring that users meet specific hardware and software prerequisites.

6.1.1 Server Requirements

The hosting of the web application and database requires the following server-related components

1. Web Server: Microsoft Azure was selected as the web server to host the web application. Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides a robust and scalable environment for hosting web applications.

2. Database Server: A dedicated server should be set up to host the MySQL database. An appropriate version (MySQL 5.7.36 or compatible) should be selected and configured for optimal performance.

6.1.2 Software Considerations

Users, including the company and its customers, must meet the following software prerequisites to access the application:

1. Operating System: A compatible operating system (e.g., Windows 7/8/10/11) should be chosen to run the web and database servers.

2. Web Browser: The application should be accessed using modern and updated browsers like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge (latest versions). The browser should be updated to the latest version to ensure compatibility and security. JavaScript must be enabled in the web browser.

6.1.3 Hardware Considerations

Users must ensure they have adequate hardware resources:

- 1.Processor: A dual-core processor or the above is recommended.
- 2.RAM: A minimum of 4 GB RAM is required to handle concurrent requests and data processing effectively.
- 3.Storage: SSD storage of 60 GB or above is recommended to enhance data retrieval speed and system performance.
- 4.Device: Users need a device (computer/laptop/smartphone/tablets) with a modern web browser installed to access the web application. Admins are highly recommended to use laptops for better usability.
- 5.Internet Connection: A stable and high-speed internet connection is required for smooth operation of the web and database servers.

In conclusion, both software and hardware considerations are crucial for the successful installation and operation of the web application.

6.1.4 Data Migration

Data migration involves the process of transferring data between different storage systems, formats, or computer systems. It is an essential step in system implementation, upgrades, or consolidation projects, ensuring that historical data is accurately and efficiently moved to a new system without data loss or inconsistency. Data migration typically includes extracting data from the source system, transforming it into the required format for the target system, and loading it into the new system.

6.1.5 Installation Guide

This guide provides step-by-step instructions for installing the web application

Step 1: Setting up MySQL Database Server

- a) Visit the MySQL official website and download the MySQL installer that suits your operating system.
- b) Run the installer and follow the prompts to install MySQL Server. During installation, you will be asked to set a password for the root user. Make sure to remember this password as you will need it later.
- c) After installation, verify that MySQL server is running.

For more detailed instructions on how to set up MySQL server, refer to this guide.

Step 2: Installing PHPMyAdmin

- a) Visit the PHPMyAdmin official website and download the PHPMyAdmin package.
 - b) Extract the downloaded package to the root directory of your web server.
 - c) Open your web browser and navigate to your PHPMyAdmin installation. This will typically be at a URL like <http://localhost/phpmyadmin>.
 - d) Follow the prompts to complete the PHPMyAdmin setup.
- For more detailed instructions on how to install PHPMyAdmin, refer to this guide.

Step 3: Setting Up the Database Tables

Before your application can interact with the database, you'll need to set up the database tables. For this, run the database.sql file in the folder, you can run it using PHPMyAdmin. For this,

- a) Extract the zip file to a location on your computer.
- b) In PHPMyAdmin, select your database on the left-hand side.
- c) Click on the "Import" tab at the top of the page.
- d) Click "Choose File" and navigate to the sql file in the extracted folder.
- e) Click "Go" at the bottom of the page to run the SQL file.

Step 4: Configure Files

Your application will need to be configured to connect to the MySQL database you set up. For this,

- a) In the extracted folder, find the connection.php file.
- b) Open connection.php in a text editor (You can use Notepad App for this).
- c) Update the hostname, username, and password for your MySQL server in this file.
- d) Save and close connection.php.

Step 5: Setting up Microsoft Azure Web Server

- a) Visit the Microsoft Azure official website and sign up for an account if you don't have one already.
- b) Once you have an account, navigate to the Azure portal.
- c) In the portal, create a new Web App resource. During this process, you will be asked to select a subscription, resource group, name, runtime stack (select the PHP 7.4 runtime stack), region, and plan. Fill in these details as per your requirements.
- d) After the Web App resource is created, go to the "Deployment Center" in the menu of your Web App resource. Here you can set up a continuous deployment pipeline if your code is in a Git repository. If not, you can choose to deploy manually.
- e) To deploy manually, go to the "Advanced Tools" in the menu of your Web App resource and click on "Go". This will open up Kudu services where you can zip deploy your code.

To zip deploy your code:

- i) Navigate to the root directory of the project in your local system.
 - ii) In Kudu services, use the "Zip Push Deploy" feature to upload and deploy your .zip file.
- For more detailed instructions on how to host a web application on Azure, refer to this guide.

Step 6: Deploying and Running the Web Application

Once you have deployed your web application to Azure, you can run it by navigating to the URL of your Azure Web App. This will be in the format <http://<your-app-name>.azurewebsites.net>.

Step 7: Purchasing a Domain Name

- a) Visit a domain registrar website such as Domain.com or GoDaddy.
- b) Use their search tool to check if your desired domain name is available.
- c) If it's available, follow their prompts to purchase it.

Step 8: Obtaining an SSL Certificate

- a) Generate a Certificate Signing Request (CSR) on your server.
- b) Purchase an SSL certificate from a Certificate Authority (CA). There are several online services that offer SSL certificates.
- c) Download your certificates from the service that you purchased them from.

For more detailed instructions on how to get an SSL certificate, refer to this guide.

Once these steps are completed, you should be able to access the web application through your web browser.

6.1.6 Training

In this system implementation, training for company employees is not a prerequisite. The design and user interface of the application are intuitive and user-friendly, enabling employees to grasp and navigate the system seamlessly. The system is engineered to facilitate self-exploration and learning while using the platform. Key features and functionalities have been designed with simplicity and clarity, allowing employees to acquire a comprehensive understanding during their interaction with the software. As such, employees can efficiently adapt to the system's operations through on-the-job learning, without the need for dedicated training sessions.

6.2 Backups

In this project, the focus of the backup strategy is specifically on the database and related data. The tool that will be used for this purpose is mysqldump, a utility provided by MySQL. mysqldump performs logical backups by producing a set of SQL statements that can be executed to reproduce the original database object definitions and table data. It dumps one or more MySQL databases for backup or transfer to another SQL server. This approach ensures that all necessary data, including table structures and stored data, are backed up and can be restored as needed.

6.3 Summary

In conclusion, the implementation phase involved careful consideration of server setup, software and hardware prerequisites, installation guide, data migration strategies, backups, and training requirements. The successful completion of this phase ensures that all prerequisites and aspects of the system are thoroughly evaluated, and users can seamlessly transition to the new system.

7.EVALUATION AND CONCLUSION

The culmination of any substantial project lies in its evaluation and conclusion. Chapter 7 embarks on the task of assessing the project's outcomes, examining its performance against predetermined objectives, and gathering user feedback to gauge its usability, accessibility, and overall reliability. This chapter intends to mark the report's endpoint, where the system's degree of success is critically evaluated, and its strengths and limitations are reflected. Moreover, while delving into the user's perspective and understanding their responses, the room for future modifications will be explored in this chapter.

The outline of chapter 7 is as follows.

- 7.1 Degree of objectives met
- 7.2 Usability, accessibility, reliability and friendliness
- 7.3 User's response
- 7.4 Limitations and drawbacks
- 7.5 Future modifications
- 7.6 Conclusion

7.1 Degree of Objectives Met

Objective: To improve real-time visibility into inventory levels.

- Degree Met: The system successfully tracks inventory levels in real-time, reducing stockouts and overstocking issues. This ensures that the pharmacy knows exactly how much of each medication is in stock at any given time, which helps in maintaining optimal inventory levels.

Objective: To prevent the wastage of medicines due to expiration by actively monitoring stock levels and expiration dates.

- Degree Met: The system effectively tracks expiration dates and notify staff to use medications before they expire. This has significantly reduced the wastage of expired medications, resulting in cost savings and improved safety for customers.

Objective: To streamline and automate the inventory management process, making it more efficient.

- Degree Met: The system automates tasks such as stock tracking, supplier order placement, and report generation. This automation has reduced the time and errors associated with manual inventory management, allowing staff to focus on providing better customer service.

Objective: To reduce lost sales opportunities by ensuring the pharmacy consistently stocks the medications customers require.

- Degree Met: Achieved. By maintaining optimal stock levels and reducing stockouts, the system ensures that customers can find the medications they need. This has resulted in increased sales and improved customer satisfaction.

Objective: To enhance customer satisfaction by providing real-time information on medication availability and estimated prescription wait times.

- Degree Met: Achieved. The system provides customers with real-time updates on medication availability and estimated wait times for prescriptions. This transparency has improved the overall customer experience, leading to higher satisfaction levels.

7.2 Usability, Accessibility, Reliability and Friendliness

Usability

The usability of the system plays a crucial role in its overall success. During the testing phases, the system underwent usability testing to ensure that it aligns with the principles of user-centered design. User tasks, navigational elements, and information architecture were evaluated. The results of usability testing have been positive, indicating that the system offers an intuitive and efficient user experience.

Sales, Accounting and Legal Management System

Accessibility

Accessibility is a fundamental aspect of software applications, ensuring that all users can interact with the system. The system was designed with accessibility guidelines in mind, adhering to standards such as WCAG (W3C, 2018). This approach ensures that users with diverse needs can access and utilize the system without hindrance.

Reliability

Reliability is crucial for any software system. It signifies that the system consistently performs its intended functions without errors or disruptions. The testing phases, including unit testing, integration testing, and system testing, have verified the system's reliability. Additionally, monitoring tools and regular updates will be in place to maintain ongoing reliability.

User-Friendliness

A friendly and intuitive interface contributes significantly to user satisfaction. The user interface was carefully designed and continuously refined based on feedback from users. Elements such as form layouts and navigation pathways were adjusted to ensure user friendliness.

7.3 User's Response

User feedback is a valuable resource for understanding how the system performs in the real world. Throughout the testing and implementation phases, user responses have been gathered and analyzed to gain insights into their experiences and expectations.

The users were provided with an introduction to the overall system and were assigned modules relevant to their roles.

The selected users were given a feedback form to express their experience and give feedback.

7.4 Limitations and Drawbacks

1. Longer Development Time

Limitation: Developing a custom solution from scratch generally takes more time compared to implementing an off-the-shelf solution.

Impact: This longer development timeline could delay the benefits of the new system and potentially increase the time before the system starts to deliver a return on investment.

2. Higher Initial Costs

Limitation: Custom development incurs higher initial costs due to the need for specialized development and potentially higher implementation costs.

Impact: This could strain the budget, especially for small to medium-sized businesses, and may require careful financial planning to manage the upfront investment.

3. Complexity of Implementation

Limitation: Custom systems can be complex to implement, requiring extensive planning, testing, and potentially more iterations to get the system fully operational.

Impact: This complexity can lead to longer implementation phases and might require hiring specialized staff or consultants to manage the process effectively.

4. Scalability Challenges

Limitation: While custom systems are designed to be scalable, they might face challenges in adapting to rapid changes or unforeseen future requirements without additional development efforts.

Impact: This can limit the system's flexibility and may result in additional costs and time for future upgrades and modifications.

5. Maintenance and Support

Limitation: Custom systems require ongoing maintenance and support, which can be resource-intensive.

Impact: The need for continuous maintenance can increase operational costs over time and require dedicated IT support to handle issues and updates.

6. Risk of Over-Customization

Limitation: Over-customizing the system to meet very specific needs can lead to a solution that is too tailored, making it difficult to adapt to broader industry standards or integrate with other systems.

Impact: This can result in a system that is less flexible and harder to update or modify to meet new business needs or regulatory requirements.

7. Dependence on Specific Vendors or Developers

Limitation: The reliance on specific vendors or developers who created the custom system can create a dependency, making it challenging to switch vendors or get support if the original developers are no longer available.

Impact: This can lead to potential issues in terms of getting timely support and updates, especially if the original developers are no longer in business or have moved on to other projects.

Feel free to provide more specific details about your project if you need further customization.

7.5 Future Modifications

Mobile Responsiveness

- **Implication:** Ensuring the inventory management system is mobile responsive so that it can be accessed easily on various devices, including smartphones and tablets.
- **Benefit:** This increases accessibility for staff, allowing them to check inventory levels, update records, and manage other tasks on the go, enhancing operational efficiency and flexibility.

Equipment Repairing Part Add

- **Implication:** Adding a module for tracking equipment repairs within the inventory management system.
- **Benefit:** This allows for better management and maintenance of equipment, ensuring that all necessary repairs are logged, tracked, and completed in a timely manner, thus reducing downtime and improving operational efficiency.

Enhance the UI/UX

- **Implication:** Improving the user interface and user experience to make the system more intuitive, easier to navigate, and visually appealing.
- **Benefit:** A better UI/UX can reduce training time, minimize errors, and increase user satisfaction, leading to more efficient use of the system.

Barcode Reader for Scanning Bill ID

- **Implication:** Integrating a barcode reader feature to scan bill IDs for faster and more accurate billing processes.
- **Benefit:** This reduces manual entry errors, speeds up the checkout process, and ensures accurate billing and inventory updates.

Customer Login and Payment Through the System

- **Implication:** Developing a customer login portal that allows customers to view their purchase history, manage their profiles, and make payments directly through the system.
- **Benefit:** This enhances the customer experience by providing a convenient and streamlined way to interact with the pharmacy, potentially increasing customer loyalty and satisfaction.

Cloud-Based Solutions for Enhanced Accessibility

- **Implication:** Migrating the system to a cloud-based platform to enhance accessibility and ensure data can be accessed from any device with an internet connection.
- **Benefit:** This ensures business continuity, improves data security and scalability, and reduces the need for on-premises infrastructure, making the system more robust and reliable.

Calculate More Business Statements by Adding Detailed Accountings

- **Implication:** Adding more detailed accounting functionalities to the system to generate comprehensive business statements and financial reports.
- **Benefit:** This provides better insights into the financial health of the business, helping in making informed decisions and improving overall financial management.

7.6 Conclusion

The implementation of the custom web-based inventory management system for Wijaya Pharmacy has demonstrated significant success in meeting its primary objectives. The system has achieved real-time visibility into inventory levels, effectively reduced medication wastage by monitoring expiration dates, streamlined and automated inventory management processes, minimized lost sales opportunities by maintaining optimal stock levels, and enhanced customer satisfaction through real-time information updates. Additionally, usability testing has confirmed the system's user-centered design, ensuring an intuitive and efficient user experience. The system also adheres to accessibility guidelines, maintains high reliability through rigorous testing, and offers a user-friendly interface.

However, the project faced limitations such as longer development times, higher initial costs, implementation complexity, scalability challenges, ongoing maintenance needs, risk of over-customization, and dependence on specific vendors. Despite these drawbacks, the potential for

future modifications, including advanced analytics integration, personalized customer experiences, mobile application development, blockchain for supply chain transparency, IoT integration, automated reordering, regulatory compliance enhancements, multi-location management, cloud-based solutions, and patient health integration, promises continued improvement and adaptation to evolving needs. This comprehensive evaluation underscores the system's substantial benefits and highlights areas for future enhancement to sustain its effectiveness and relevance.

8. REFERENCES

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

<https://www.lucidchart.com/pages/uml-sequencediagram#:~:text=A%20sequence%20diagram%20is%20a,to%20document%20an%20existing%20process.>

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

<https://www.javatpoint.com/uml-class-diagram>