



Daffodil
International
University

MoBook

By

Md. Shakibur Rahman

(171-35-1839)

A thesis submitted in partial fulfillment of the requirement for
the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL
UNIVERSITY

Semester – 2022

Approval

This Project titled on “MoBook”, submitted by **Md.Shakibur rahman (ID: 171-35-1839)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

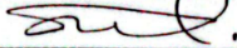


Chairman

Dr. Imran Mahmud
Head and Associate Professor

Department of Software Engineering

Faculty of Science and Information Technology
Daffodil International University



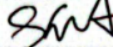
Internal Examiner 1

Md. Khaled Sohel
Assistant Professor

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University



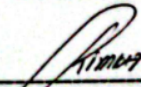
Internal Examiner 2

Md. Shohel Arman
Assistant Professor

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

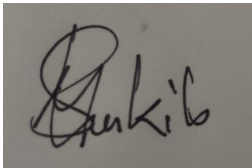


External Examiner

Rimaz Khan
Managing Director
Tecognize Solution Limited

DECLARATION

I hereby declare that project titled “MoBook” has been completed by me under the supervision of Ms. Nusrat Jahan, Assistant Professor, Department of Software Engineering, Daffodil International University for the purpose of achieving degree of Bachelor of Science from Daffodil International University. This is also declared by me that neither this project nor any part of this project has been used or submitted elsewhere for any kind of degree or awards.



.....
Md. Shakibur Rahman

ID: 171-35-1839

Department of Software Engineering

Daffodil International University



.....
Ms. Nusrat Jahan

Assistant Professor

Department of Software Engineering

Daffodil International University

Supervisor

ACKNOWLEDGMENT

The Almighty Allah deserves all of the praise and thanks, especially for his blessings on my project work, for providing me patience, and for allowing me to complete my project on time.

I want to show my sincere gratitude to my esteemed boss, Ms. Nusrat Jahan, for providing me with the right guidance and assisting me in completing the task at the end. I would like to especially thank my boss for helping me with the project. His great direction, counsel, and insightful constructive criticism have enabled me to confidently complete this project.

I consider myself extremely fortunate to have the chance to attend Daffodil International University. I want to express my sincere appreciation to our department's distinguished head of department, SWE, for supporting me and providing me with this chance. I want to express my gratitude to all of my esteemed professors for their excellent, fascinating, and clear instruction. They made my travel the simplest and most fun, and I am appreciative of that.

Thanks to my parents and all my family. Without their constant support and cooperation, my whole school trip wouldn't have been so easy.

Thank you to all department staff, lab technicians and non-educated staff for their ultimate support throughout my journey.

Finally, I would like to thank the DIU members and my fellow classmates for their great assistance and comfort in getting all of the work done.

ABSTRACT

This project consists of a website-based event management platform (Mobook). This project will provide the ability to remotely create, delete and retrieve information and assign responsibilities to different participants at events. This project can grant necessary access rights to administrators and everyone involved in a particular event. Event managers can view the guest list and attendees. Additionally, you have the option to add or remove events. This project streamlines the event management process by reducing paperwork and labor requirements. Any need can be tracked. Humans can provide input, collect it and use it for improvement. You can request employees, assign tasks, track income surveys and more through this platform. The aim of this project is the digitization of event management. It is also easy to use, allowing anyone to perform event management tasks anytime, anywhere. Most of the basic functionality an organizer needs to run an event properly is provided by the organizer. This is done to meet the needs of both event administrators and participants. You can enter your information to register for the event. Admins can receive this information and use it to communicate with participants.

Keywords: Functional, Participant, Event Management, and Traceable

Table of Contents

APPROVAL	i
DECLARATION.....	ii
ACKNOWLEDGMENT	iii
ABSTRACT.....	iv
Table of Contents	v
List of figures	x
List of Tables	x
CHAPTER 1	1
INTRODUCTION	1
1.1 Project Overview	1
1.2 Project Purpose	1
1.3 Background	2
1.4 Benefits & Beneficiaries	2
1.5 Goals	2
1.6 Stakeholders	3
1.7 Project Schedule.....	3
1.8 Gantt Chart.....	4
1.9 Project Timeline.....	5
CHAPTER 2	6
SOFTWARE REQUIREMENT SPECIFICATION	6
2.1 Functional requirement: Customer	6
2.2 Functional requirement: Provider (Admin)	7
2.3 Performance Requirements	6
2.4 Capacity Requirements	6
2.5 Requirements for Reliability.....	6
2.6 Performance Conditions.....	6

2.7 Defect-Tolerance Conditions	6
2.8 Critical Safety Requirements	6
2.9 Maintainability and Supportability Requirements	7
2.10 Maintenance Requirements.....	7
2.11 Supportability Requirements	7
2.12 Adaptability Requirements	7
2.13 Security Requirements	7
2.14 Access Requirements	7
2.15 Integrity Requirements.....	7
2.16 Privacy Requirements	7
2.17 Usability and Human-Interaction Requirements.....	7
2.17.1 Ease of Use Requirements	8
2.17.2 Understandability Requirements.....	8
2.17.3 Accessibility Requirements.....	8
2.17.4 User Documentation Requirements	8
2.17.5 Training Requirements.....	8
CHAPTER 3.....	9
SYSTEM ANALYSIS & DESIGN	9
3.1 User profile in use case: 1	9
User profile: 2	10
3.2 Utilization Case Drawing.....	11
3.2.1 Description of Use Case (Customer)	12
3.2.2 Description of Use Case (Provider).....	13
3.3 Diagram of Activities.....	14
3.3.1 Action of User Login	14

3.3.2 Customer Registration Action.....	15
3.3.3 User View service Action	16
3.3.4 Service provider puts Action	17
3.3.5 Provider edit service Activities	18
3.3.6 Provider delete service Activities	19
3.3.7 User add to cart Action	20
3.3.8 User manage cart Action.....	21
3.3.9 User edit cart data Action	22
3.3.10 Delete cart Action	23
3.3.11 Order place Action.....	24
3.3.12 Pay order Action	25
3.3.13 Show order list Action	26
3.4 Sequence Diagram	27
3.4.1 Registration Sequence Diagram.....	27
3.4.2 Login Sequence Diagram.....	28
3.4.3 Add product Sequence Diagram	29
3.4.4 View service Sequence Diagram	30
3.4.5 Add to cart Sequence Diagram	31
3.4.6 My order Sequence Diagram	32
3.5 ERD.....	33
3.6 Design and Implementation Constraints.....	34
3.6.1 Programming Language or Environment.....	34
3.6.2 Tools and Technology for Development	34
3.7 Management of projects.....	34
3.7.1 Software Process Model	34

CHAPTER 4.....	34
SYSTEM TESTING.....	34
4.1 Exploring Features	34
4.1.1 Specific Features to be Explored	34
4.2 Evaluating Methods	34
4.2.1 Test Methodology	34
4.2.2 Test Group	34
4.3 Criteria for Success or Failure	35
4.4 The Trial Context.....	35
4.5 Testing.....	36
1. Unit Testing	36
2. Validation Testing.....	36
3. System Testing.....	36
4. Acceptance Teasing	37
5. Regression Testing.....	37
6. Database Testing.....	37
4.5 Test Cases	38
Testing Case, No-2 (Module Testing)	39
CHAPTER 5.....	40
USER MANUAL	40
5.1 User Sign in Page.....	40
5.2 User Register Page	41
5.3 User Dashboard.....	41
5.4 All Companies details.....	42
5.5 All Events list.....	42

5.6 Event details.....	43
5.7 My Order page	43
5.8 Provider registration.....	44
5.9 Provider Login	44
5.10 Provider Dashboard	45
5.11 Add Service Page	45
5.12 Services details page	46
5.13 Provider Order list page	46
5.14 Add Company details page	47
5.15 Company details Page.....	47
CHAPTER 6.....	48
PROJECT SUMMARY	48
6.1 GitHub Link	48
6.2 Limitations.....	48
6.3 Challenges and Successes	48
6.4 Future Scope.....	48
6.5 References	49
6.6 Plagiarism Report	50

LIST OF TABLES

CHAPTER 1

Table 1.8.2: Project Timeline	5
-------------------------------------	---

CHAPTER 4

Table 4.5.1: Testing Case (Admin and User login).....	41
Table 4.5.2: Testing Case (Provider).....	42

LIST OF FIGURES

CHAPTER 1

Figure 1.4.1: Gantt chart	4
---------------------------	---

CHAPTER 3

Figure 3.1: Use Case	11
Figure 3.3: Action Diagram	14
Figure 3.4: Sequence Diagram	27
Figure 3.5: Entity Relation Diagram	33
Figure3.9.1: Agail Model	35

CHAPTER 5

Figure 5.1 UI (User Login Page)	43
---------------------------------	----

Figure 5.2 UI (User Register Page)	43
Figure 5.3 UI (User Dashboard Page)	44
Figure 5.4 UI (All Companies details)	44
Figure 5.5 UI (All Event List)	45
Figure 5.6 UI (Event details)	45
Figure 5.7 UI (My order)	46
Figure 5.8 UI (Provider register)	46
Figure 5.9 UI (Provider login)	47
Figure 5.10 UI (Provider dashboard)	47
Figure 5.11 UI (Add service)	48
Figure 5.12 UI (Services details)	48
Figure 5.13 UI (Provider Order list)	49
Figure 5.14 UI (Add company details)	49
Figure 5.15 UI (Company details)	50

CHAPTER 1

INTRODUCTION

1.1 Project Overview

"MoBook" was developed as a solution to address the inefficiencies inherent in manual work systems. It aims to support, eliminate, and potentially mitigate the challenges present in our current system. Moreover, the program is tailored to meet the specific operational needs of companies, ensuring efficiency and effectiveness.

The design of the program prioritizes simplicity to minimize typing errors, with error messages promptly alerting users to inaccurate data entry. No specialized training is required to navigate the system, making it highly user-friendly. As mentioned, an error-free, secure, reliable, and swift event management system is essential. This frees up users to concentrate on other tasks, facilitating better goal attainment and resource utilization within organizations.

Businesses of all sizes encounter difficulties in managing various aspects such as activities, events, attendees, payments, and conductors' data. Given the diverse requirements of each event management system, our platform offers a bespoke talent management solution to meet specific management needs. This aids in strategic planning, ensuring organizations possess the requisite knowledge and specificity to achieve long-term objectives.

Our solution caters to busy executives who require remote access and the capability to manage employees on-the-go. Ultimately, these integrated solutions empower efficient resource management, facilitating improved organizational performance.

1.2 Project Purpose

- Computer systems necessitate the completion of various forms, with the ability to generate copies simultaneously.
- They produce diverse types of information suitable for multifaceted applications.
- Designed for user and operator comprehension.
- Tailored to fulfill user requirements.

- Ensures ease of understanding for users and operators alike.
- User-friendly interface.
- Boasts a well-designed user interface.
- Designed for scalability and expansion.

1.3 Background

The main goal of this project is to develop a typical "Event Management System" for a "Event Management Company." A consumer requests event management from the business. He describes the occasion's specifications in detail. He describes the event's objectives, the time and location of the event, how long it will last, the format (presentation, workshop, or exhibition, for example), the anticipated number of attendees, the equipment and furniture needed, whether any promotional materials or delegate packs will be distributed, and any other facilities needed.

The event manager carefully considers the event's needs before determining an estimated cost and sharing that information with the client. The buyer has the option to verify if the price is in line with the event's budgetary requirements. The business may also provide a selection of pre-made bundles.

If the client concurs, the event is scheduled and the firm receives the advance payment. Different reservations are made depending on the event's requirements. A thoughtful timetable is created to ensure the event runs well. The manager may complete many activities related to event planning, scheduling, and execution with the aid of the event management system. Instant access to event-related information is provided by this technology. As a result, resources are used effectively and affordably. The system generates the invoices when the event is successfully completed. The system can be expanded. When necessary, owing to shifting requirements, the system may be expanded to include new features.

1.4 Benefits & Beneficiaries

This solution has a beneficial side. They are listed below:

- Reduce time-consuming
- Extra effort

- Digital solution
- Easy to manage
- Ensure better service for other event management system
- Friendly service for users.

1.5 Goals

- Construct this program on a live server.
- One stop service
- Digitize your system.

1.6 Stakeholders

Our solution has two types of stakeholders. they are:

- **Customer**

Citizens play the most important and sensitive role in this system. Citizens can generate digital driver licenses, file complaints, generate QR codes and download PDFs.

- **Provider (Admin)**

This system allows authorities to play a secondary role, assessing driver information, investigating complaint reports and taking action.

1.7 Project Schedule

To finish every project on schedule, you need a project plan.

1.8 Gantt Chart

This is a graphical representation of completed jobs over time. Visualizing what you have to do for the day is a great technique. It also helps to indicate project start and end dates.

Activities		W	W	W	W	W	W	W	W	W	W	W	W	W	W
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Planning	Concepts														
	Definition of the problem														
	Suggestion														
Requirements	Need Details														
	Analysis of requirements														
QA -1	Assurance of quality														
System Design	Design guidelines														
	Design of interfaces														
	Designing databases														
Development	Development of machine modules														
	Integrate system modules														
QA -2	Test Cases														
Testing	Unit testing														
	Black box testing														
Resolve Issues	Resolve issues found														
Release	Software release														

Figure 1.8.1: Gantt chart

1.9 Project Timeline

The project timeline record is shown below:

Task	Date
Theme selection seminar	
Brainstorming Choosing a project topic and name	
Submit a project proposal	
Requirement specification	
Mid Term Defense	
System Analysis & Design	
Development System	
System Testing	
Complete Project	

Table 1.8.2: Project Timeline

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional requirement: Customer

Requirement ID	FR.C.1
Requirement Name	Registration
Description	Users that meet the requirements can register on the system.

Requirement ID	FR.C.2
Requirement Name	Login
Description	Users can login using email and password.

Requirement ID	FR.C.3
Requirement Name	View service
Description	User can select the event and also see all event details

Requirement ID	FR.C.4
Requirement Name	Add to cart
Description	customers to choose items to purchase without actually completing the payment

Requirement ID	FR.C.5
Requirement Name	Order place
Description	To register an order for some service that one will receive in the near future.

Requirement ID	FR.C.6
Requirement Name	Payment
Description	Users can select payment method

Requirement ID	FR.C.7
Requirement Name	View order
Description	Users can view their orders

2.2 Functional requirement: Provider (Admin)

Requirement ID	FR.A.1
Requirement Name	Registration
Description	Any Provider can register on the system by giving require info.

Requirement ID	FR.A.2
Requirement Name	Login
Description	Provider can login using email and password

Requirement ID	FR.A.3
Requirement Name	Add service
Description	Provider can add the event and also added all event details

Requirement ID	FR.A.4
Requirement Name	Manage service
Description	Provider can manage my service

Requirement ID	FR.A.5
Requirement Name	Manage order
Description	The process of receiving, tracking, and fulfilling customer orders.

Requirement ID	FR.A.6
Requirement Name	Company details
Description	Provider can manage own company details.

Requirement ID	FR.A.7
Requirement Name	View order
Description	Provider can view all orders.

2.3 Performance Requirements

Quality is the most crucial of the reliability requirements, which also include security and availability. I also require these standards.

2.4 Capacity Requirements

It is necessary for devices to process both kinds of user data.

CR-1	Large volumes of data can be handled by the system.
Description	The system needs to be capable of handling different kinds of data.
Stakeholder	Provider

2.5 Requirements for Reliability

Reliability is gauged across four dimensions, including availability, dependability, and security. Gadgets must meet these four criteria to ensure optimal performance and trustworthiness.

2.6 Performance Conditions

Reliability refers to the probability of a machine operating smoothly without any issues.

RR-1	The systems have to be operational. Every day, 24/7
Description	The systems must constantly be reachable, virus-free, and current.
Stakeholder	

2.7 Defect-Tolerance Conditions

Maintaining fault tolerance requires providing clients with correct performance and zero crashes.

FTR-1	Without any system problems, the system manages all user access.
Description	The system should handle the request without error and allow both users to access it at the same time.
Stakeholder	N/A

2.8 Critical Safety Requirements

There are no safety-critical criteria for my project.

2.9 Maintainability and Supportability Requirements

Giving consumers assistance or service after the sale is crucial.

2.10 Maintenance Requirements

MR-1	Driver licenses and complaints are handled by this system.
Description	very important.

Stakeholder	Provider
--------------------	----------

2.11 Supportability Requirements

There are some support requirements. they have:

- maintainability
- configurability
- compatibility
- Easy maintenance

2.12 Adaptability Requirements

There is no adaptation in my project.

2.13 Security Requirements

One cannot stress the significance of equipment safety regulations. It ought to be grounded in practical needs. Application system security is the focus of software protection. There are some prerequisites for security. They have:

- Linking suppliers and customers
- Get access based on the user that is currently logged in.
- Break as suppliers and customers

2.14 Access Requirements

No input required in my work.

2.15 Integrity Requirements

The project does not enforce any authenticity regulations.

2.16 Privacy Requirements

Every system is mandated to establish privacy protocols. Users can register for the system by verifying their information and utilizing privacy settings to protect their confidentiality.

2.17 Usability and Human-Interaction Requirements

The foremost goal of any utility solution is to ensure user-friendliness and ease of operation.

2.17.1 Ease of Use Requirements

Our solution is simple to apply and comprehend.

EUR-1	Users will find the system easy to use.
Description	This solution simplifies system management for users.
Stakeholder	Customer and Provider

2.17.2 Understandability Requirements

There are no clear criteria in my project.

2.17.3 Accessibility Requirements

For my project, there are no distinct usability criteria.

2.17.4 User Documentation Requirements

For my project, user documentation is not required.

2.17.5 Training Requirements

No requirements for user documentation are present in my project.

CHAPTER 3

SYSTEM ANALYSIS & DESIGN

3.1 User profile in use case: 1

Class of User: Client	Features	Inferred Requirement
User type	Primary	Must give input
Age range	18-65	Minimal Design
Number of users	Unlimited	Bandwidth should be high
Education	Educated	Simple interface
Language Skill	English	Simple English
PC/ Mobile Knowledge	Yes	Type option or Click option
Training	Not required	Not required
Goal	Add to cart service, choose a service, Control my order and make digital payments.	Need to view product details, customer details, and add to cart Control my order and make digital payments.

User overview: 2

Class of User: Provider	Features	Inferred Requirement
User type	Primary	Must see output
Age range	20-65	Minimal Design
Number of users	2	Bandwidth should be normal
Education	Educated	Simple interface
Language Skill	English	Simple English
PC/ Mobile Knowledge	Yes	Only type or click option
Training	Required	2 days training will be provided
Goal	Self-registration Add services, manage service, manage order, Manage company details.	Must See Dashboard, add company details, Manage services Information.

3.2 Utilization Case Drawing

This use case illustration features two distinct actors. The diagram meticulously refines my project, enhancing its clarity and precision.

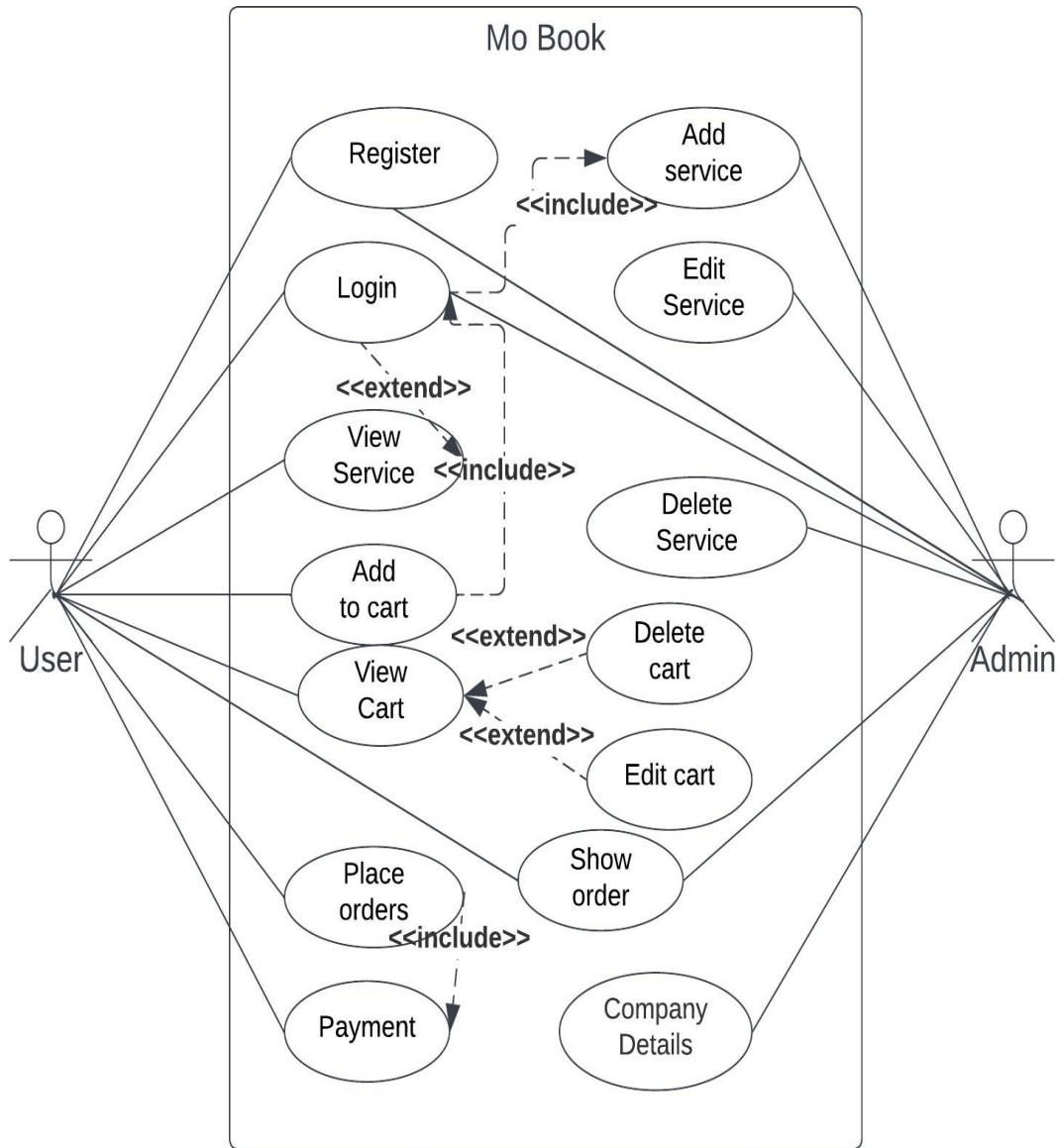


Figure 3.1 : Use Case for MoBook

3.2.1 Description of Use Case (Customer)

Name of use case:	Customer
Description:	Chose's service, add to cart service, Manage my order, Digital Payment
Actors:	Customer
Preconditions:	<ol style="list-style-type: none"> 1. Customer must be self-registered 2. Must be Verified
Postconditions:	
Flow:	<ol style="list-style-type: none"> 1. Customer login System 2. Chose's service 3. Add to cart service 4. Manage my order 5. Digital Payment
Alternative Flows:	
Exceptions:	Easy service booking solution
Prerequisites:	<p>To run the use case, the following conditions must be satisfied.:</p> <ol style="list-style-type: none"> 1. The Client Has to Sign Up 2. Servicing has to be Chose

3.2.2 Description of Use Case (Provider)

Name of use case:	Provider
Explained:	Every order and delivery is managed by authority.
Actors:	Provider (Admin)
Preconditions:	Registration of Authority Is Required
Postconditions:	
Flow:	<ol style="list-style-type: none"> 1. Supplier Signs in to their Online Account 2. Add service 3. Manage order 4. Manage company details
Differential Flows:	
Exclusivity:	Simple Evaluation in a Wise Approach, Time Consumption
Requirements:	<p>Before executing the use case, the following prerequisites must be satisfied.:</p> <ol style="list-style-type: none"> 1. Requires Self-Registration

3.3 Diagram of Activities

3.3.1 Action of User Login

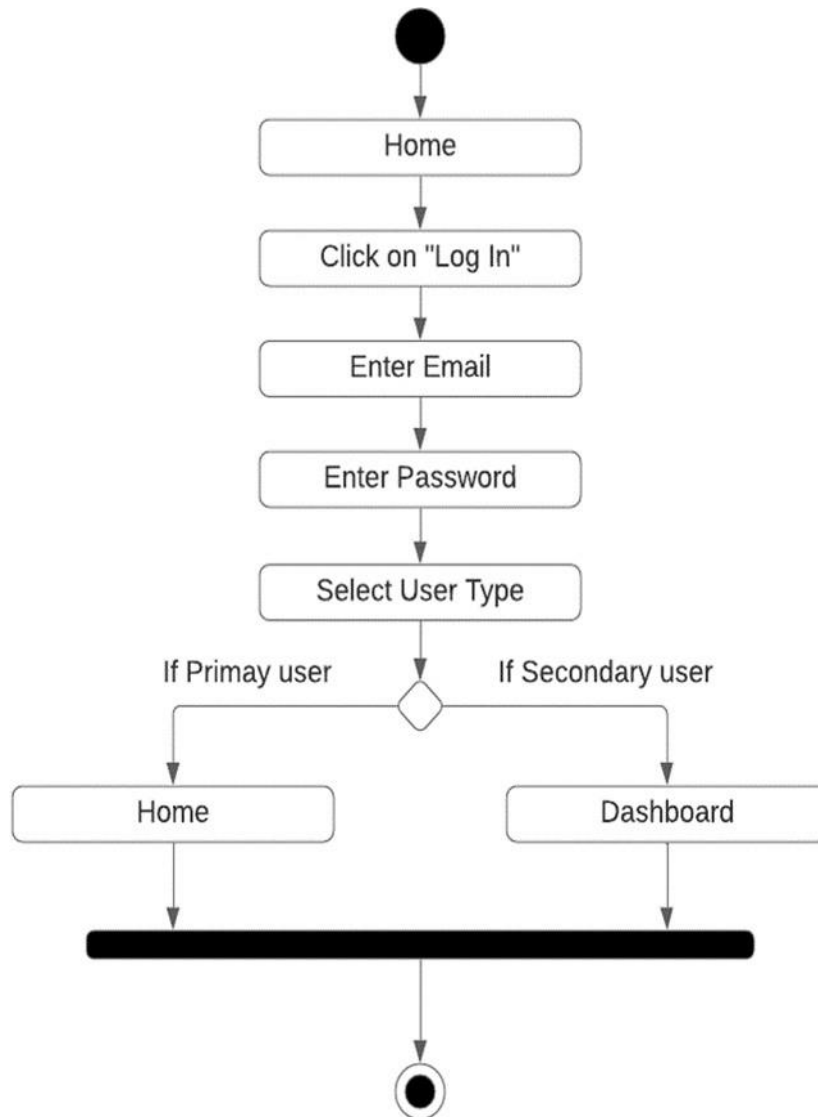


Figure 3.3.1 : Customer Sign in Diagram of Activities

3.3.2 Customer Registration Action

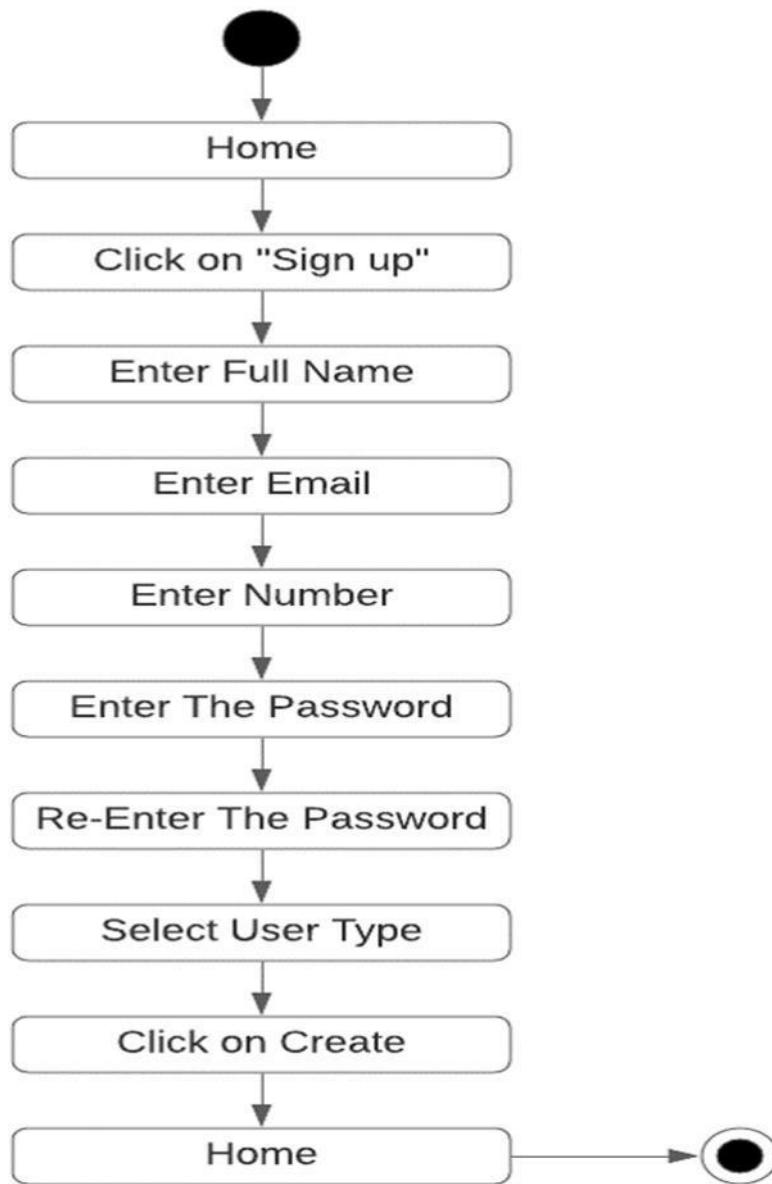


Figure 3.3.2: Action Diagram user registration

3.3.3 User View Service Action

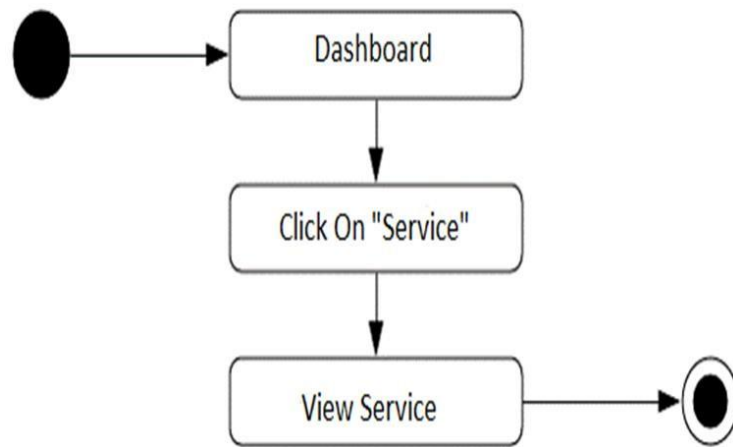


Figure 3.3.3: User View Service Action Diagram

3.3.4 Service provider puts Action

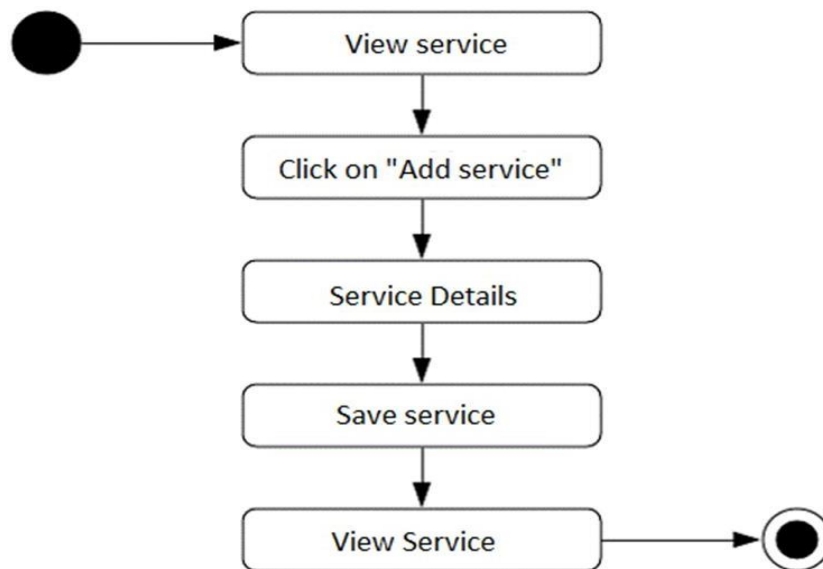


Figure 3.3.4: Service provider adds Diagram of Activities

3.3.5 Provider edit service Activities

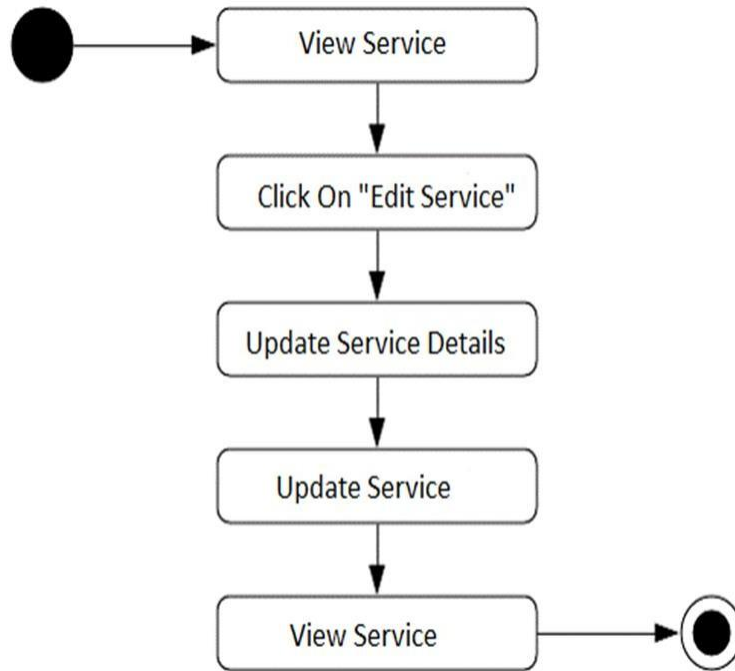


Figure 3.3.5: Provider edit service Activities Diagram

3.3.6 Provider delete service Activities

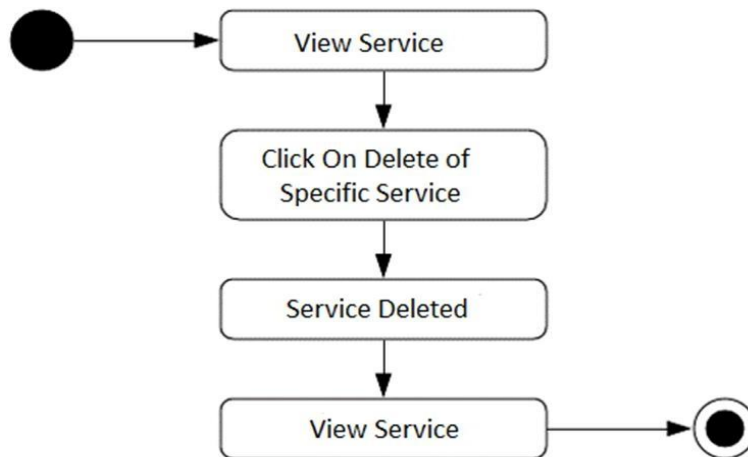


Figure 3.3.6: Provider delete service Activities Diagram

3.3.7 User add to cart Action

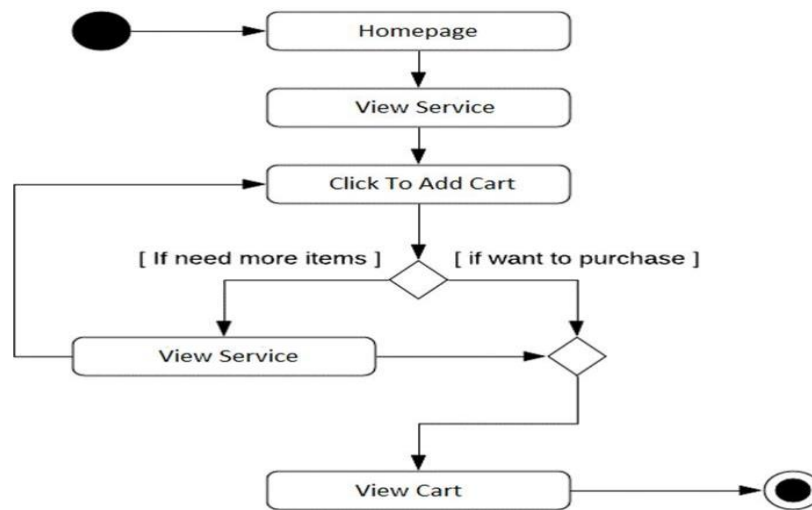


Figure 3.3.7: User add to cart Action Diagram

3.3.8 User manage cart Action

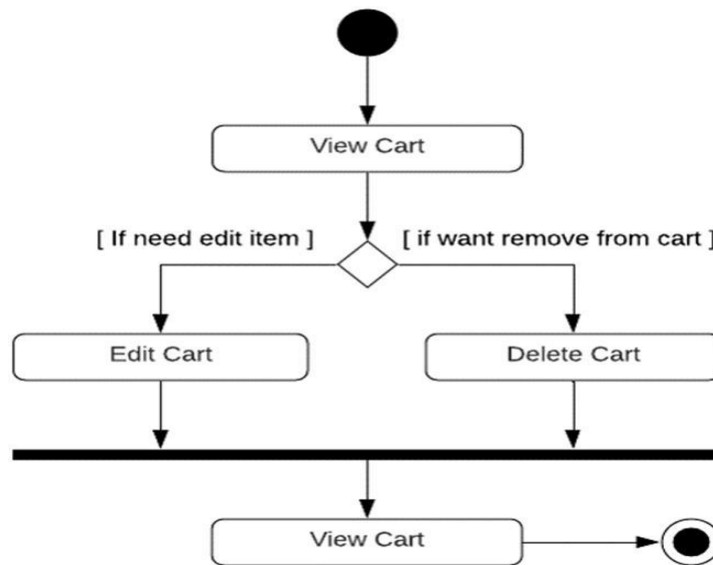


Figure 3.3.8: User manage cart Action Diagram

3.3.9 User edit cart data Action

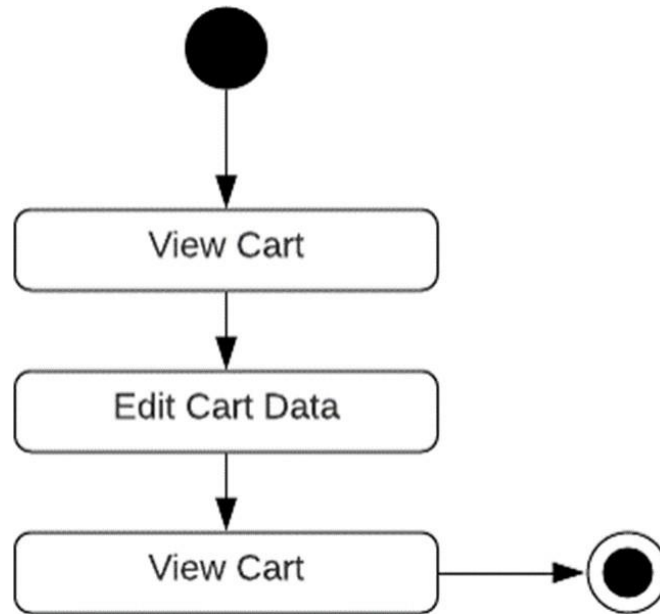


Figure 3.3.9: User edit cart data Action Diagram

3.3.10 Delete cart Action

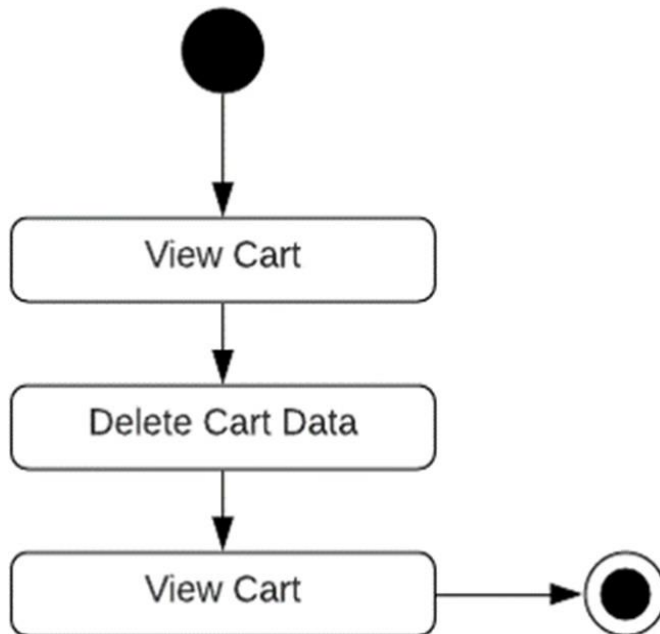


Figure 3.3.10: Delete cart Action Diagram

3.3.11 Order place Action

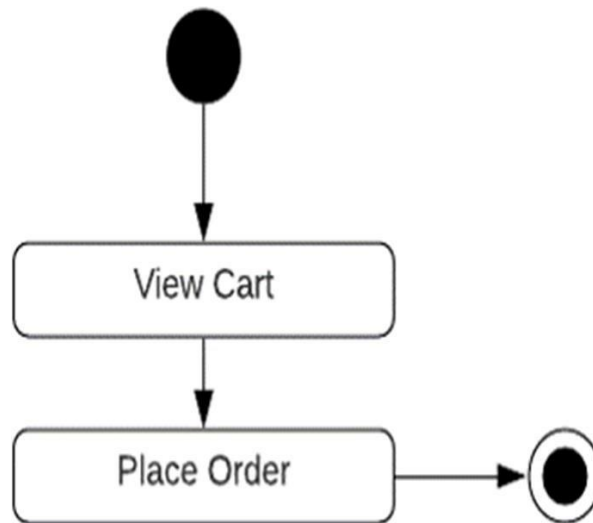


Figure 3.3.11: Order place Action Diagram

3.3.12 Pay order Action

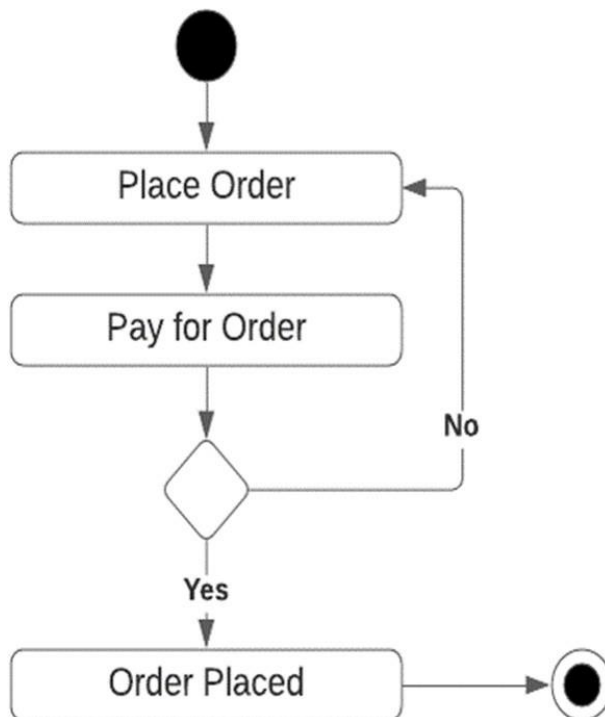


Figure 3. 3.12: Pay order Action Diagram

3.3.13 Show order list Action

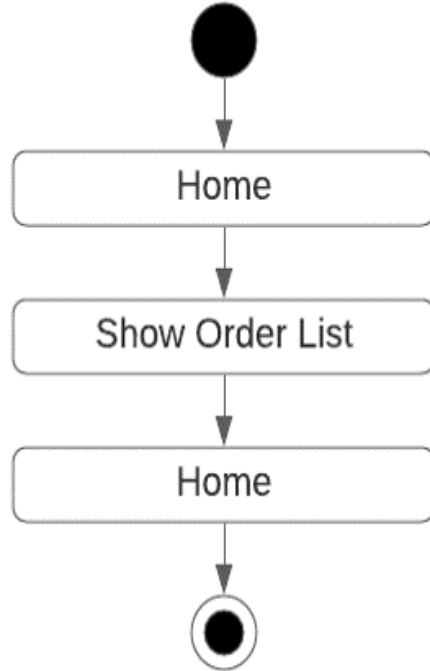


Figure 3. 3.13: Show order list Action Diagram

3.4 Sequence Diagram

3.4.1 Registration Sequence Diagram

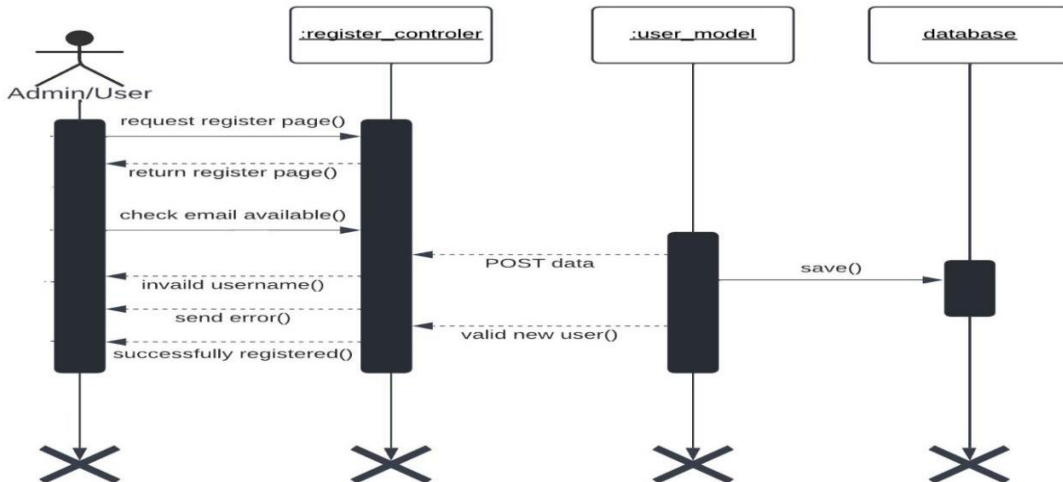


Figure 3.4.1: Registration Sequence Diagram

3.4.2 Login Sequence Diagram

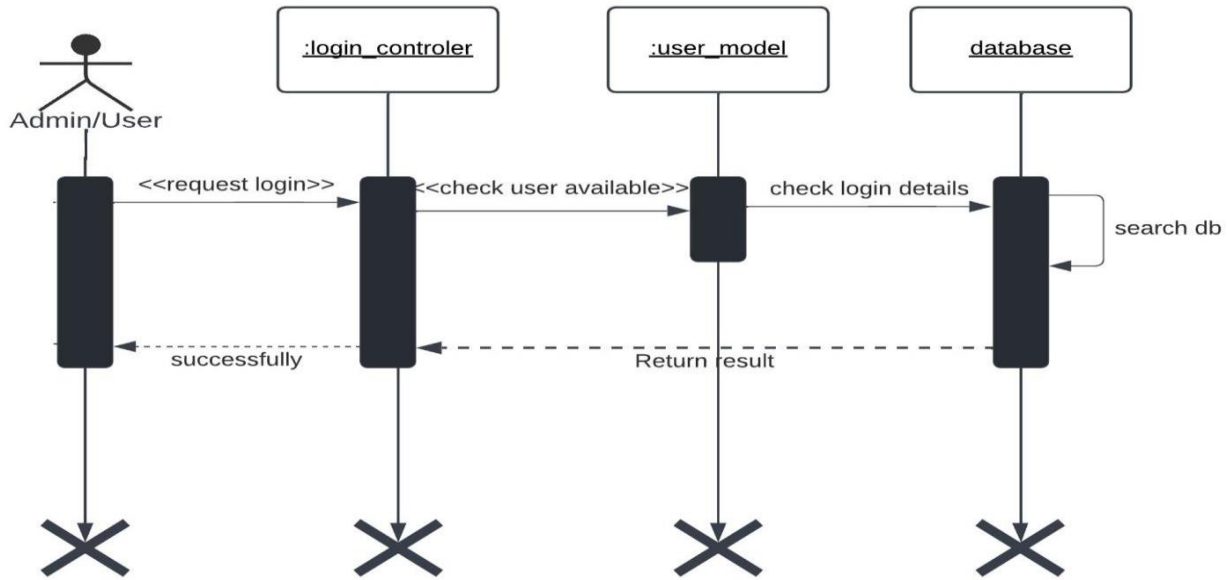


Figure 3. 4.2: Login Sequence Diagram

3.4.3 Add product Sequence Diagram

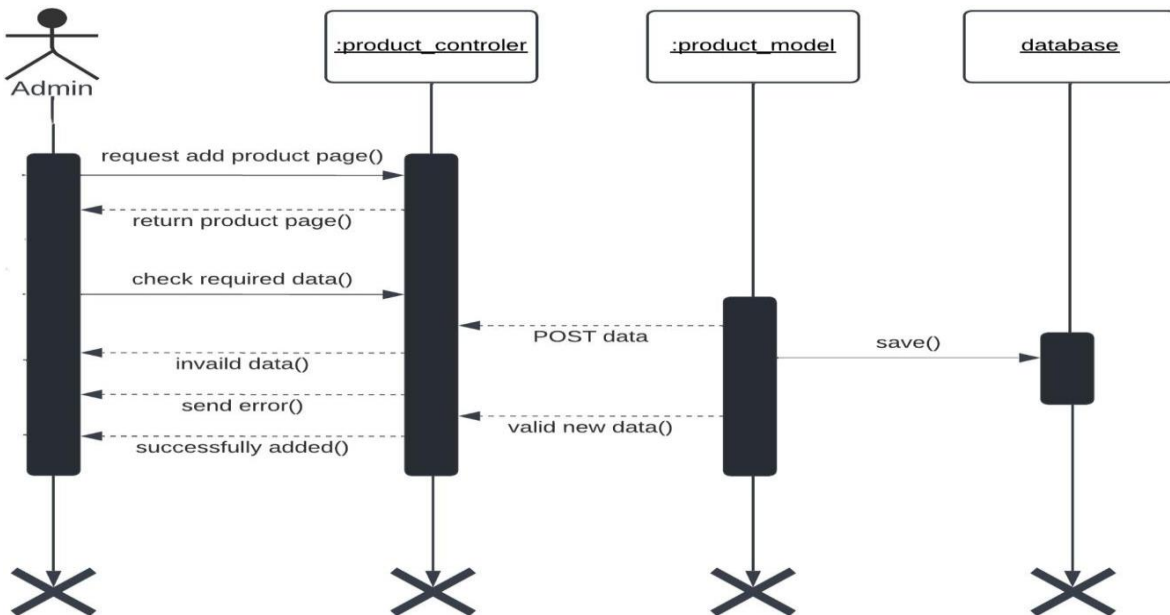


Figure 3.4.3: Add product Sequence Diagram

3.4.4 View service Sequence Diagram

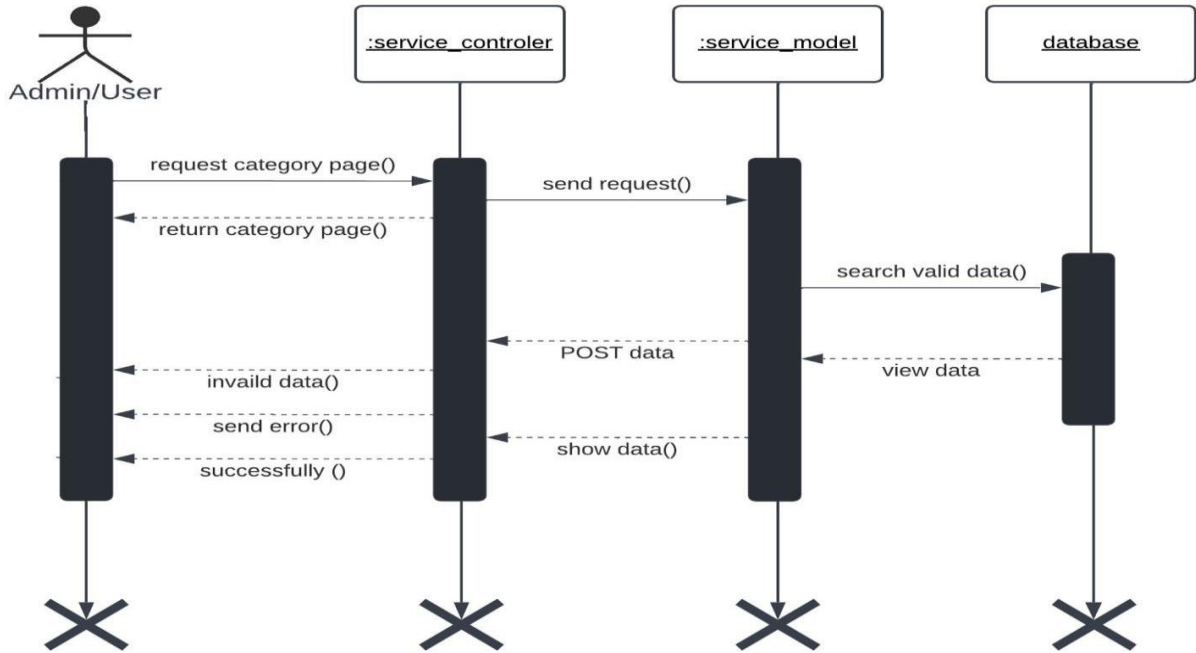


Figure 3.4.4: View service Sequence Diagram

3.4.5 Add to cart Sequence Diagram

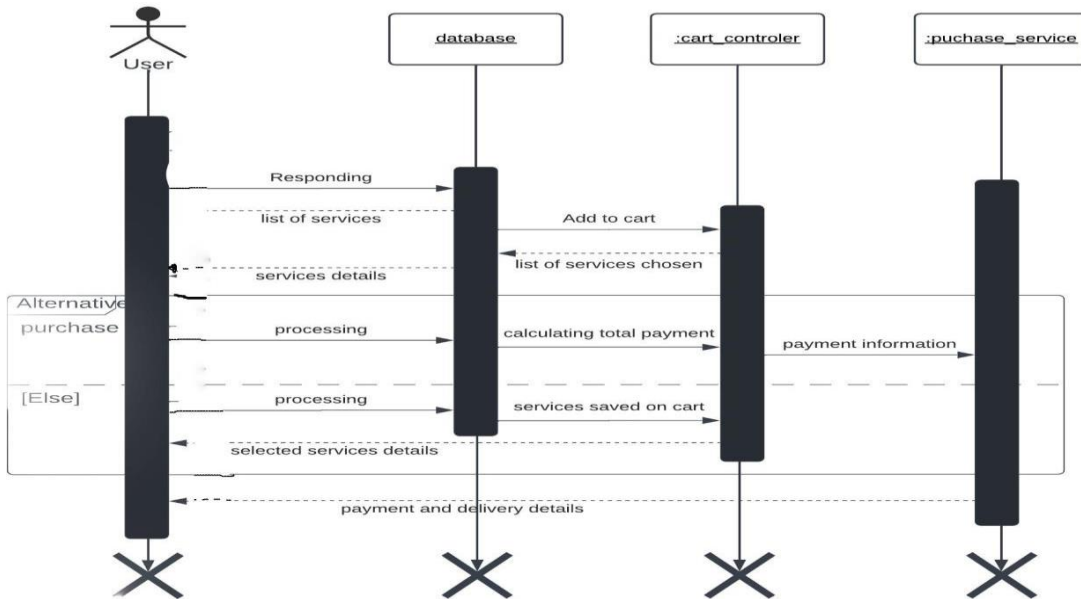


Figure 3.4.5: Add to cart Sequence Diagram

3.4.6 My order Sequence Diagram

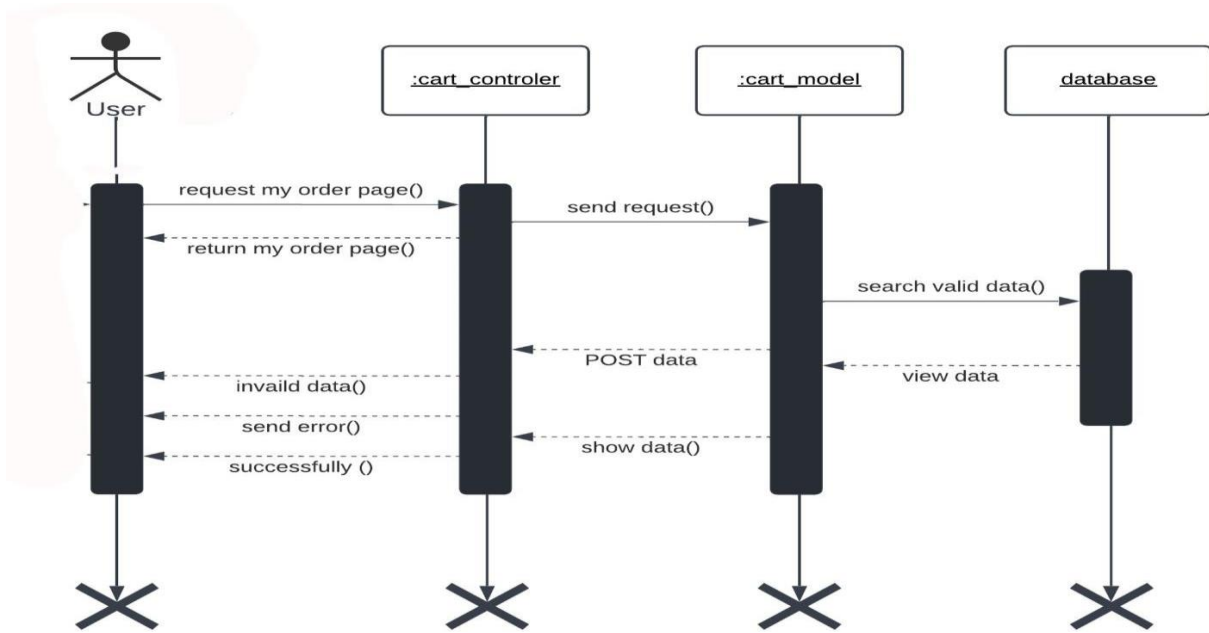


Figure 3.4.6: My order Sequence Diagram

3.5 ERD

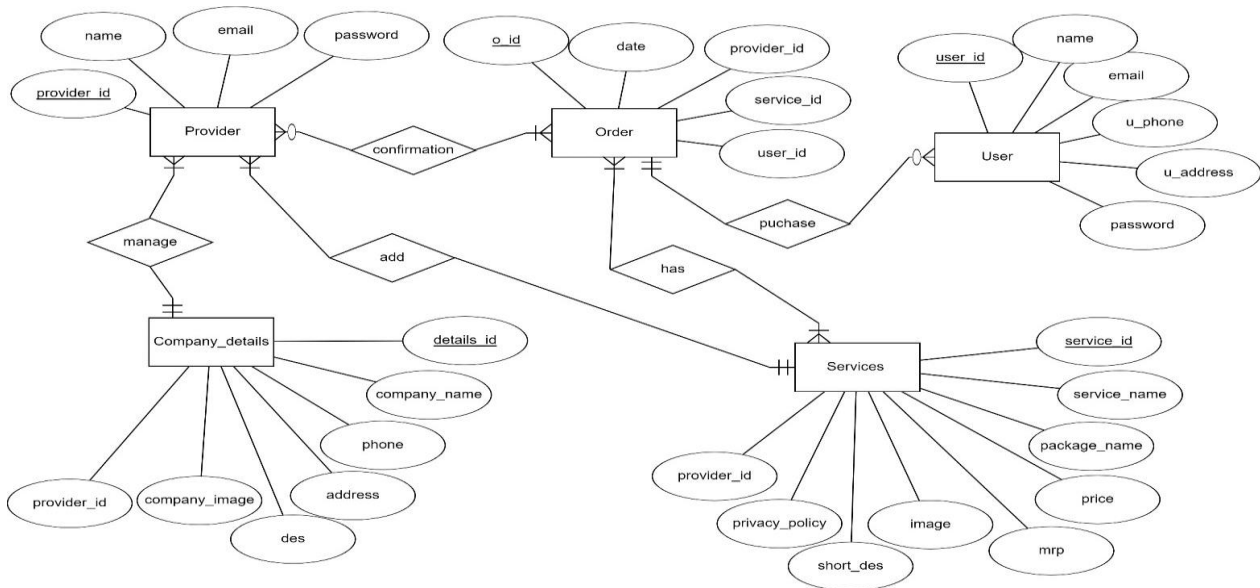


Figure 3.5: MoBook ERD

3.6 Design and Implementation Constraints

3.6.1 Programming Language or Environment:

"Mebook" is constructed in PHP with the Laravel framework; the data is kept in a MySQL database, and the user interface is created using bootstrap templates.

3.6.2 Tools and Technology for Development:

- IDE: Visual Studio Code
- Database: MySQL
- UI: HTML, CSS, Bootstrap, JavaScript
- Framework: Laravel 8
- Language: PHP
- Web-Server: XAMPP Server

3.7 Management of projects

3.7.1 Software Process Model

Iterative and incremental process models are used in the agile software development life cycle (SDLC) paradigm, which prioritizes process flexibility and customer satisfaction through prompt delivery of functioning software. Agile techniques are used to break the product up into smaller, incremental builds. These builds' iterations are provided. An iteration typically lasts one to three weeks. Cross-functional teams work simultaneously on various different issues throughout each iteration, including.

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing

At the end of the iteration, a working product is shown to the client and other important stakeholders.

Explain Agile:

The agile methodology states that every project should be managed uniquely and that existing protocols should be modified as needed to better suit the needs of the project. Under the agile technique, tasks are divided into time boxes, or brief time intervals, to provide specific features for a release.

The Agile Model is displayed in the graphic below.

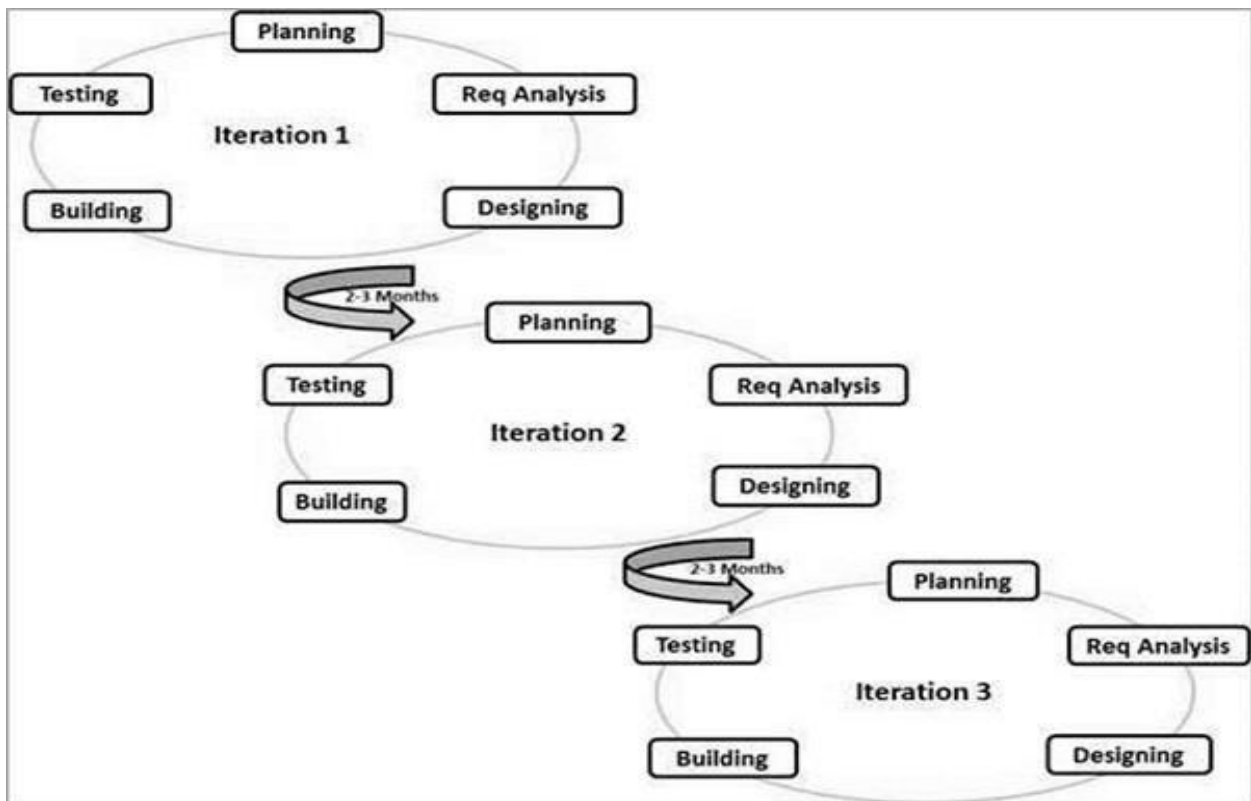


Figure3.7.1: Agile Model

Because of its adaptability and flexibility, the Agile thought process gained popularity over time and was first used in software development.

The most prominent Agile methodologies, including the Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995), were established prior to the release of the Agile Manifesto in 2001, subsequently recognized as Agile Methodologies.

The Agile Manifesto espouses the following guiding principles:

- Agile development places significant emphasis on the individuals involved and their interactions. It fosters self-organization, motivation, and collaborative practices such as pair programming and co-location.
- Rather than relying solely on documentation, Agile advocates for demonstrating usable software as the most effective means of engaging with customers and comprehending their needs.
- Recognizing the dynamic nature of project requirements, Agile underscores the importance of continuous customer collaboration to ascertain precise product demands.
- Agile development prioritizes continuous improvement and agility in responding to change, reflecting a commitment to adaptability and flexibility.

Agile Vs Traditional SDLC Models

In contrast to traditional SDLC models such as the waterfall model, agile methodology emphasizes adaptive software development approaches. While conventional models prioritize predictive strategies, agile focuses on flexibility and responsiveness to changing requirements throughout the development process.

In traditional SDLC models, predictive teams engage in meticulous planning and have a detailed roadmap of activities and features to be delivered over the coming months or the product's lifecycle. Predictive approaches rely heavily on initial planning and demand analysis. Any proposed modifications typically undergo a rigorous change management and prioritization process.

Agile methodology, on the other hand, embraces adaptability by eschewing extensive upfront planning. Future tasks are only loosely defined in terms of the features to be produced, and development is driven by the incremental delivery of these features. Agile teams continuously adjust to evolving product needs, frequently testing and iterating on the product to mitigate potential issues early on.

Customer interaction lies at the core of the Agile approach, fostering open communication and minimizing bureaucratic paperwork. Agile teams often work in close collaboration, frequently co-located in the same physical space to facilitate seamless communication and teamwork.

Agile Model - Pros and Cons

Agile approaches' practical approach to software development has led to their broad adoption in the software industry in recent years. It's important to understand, though, that although agile might work wonders for a lot of projects, it might not be the greatest option for every kind of product or circumstance. Let's explore some of the benefits and drawbacks of the agile model:

Benefits of the Agile Model:

- **Realistic Approach:** Agile methodology offers a practical and realistic approach to software development, focusing on delivering working solutions incrementally rather than trying to predict and plan every detail upfront.
- **Teamwork and Cross Training:** Agile promotes collaboration and teamwork among developers, encouraging cross-training and knowledge sharing within the team.
- **Rapid Development and Demonstration:** Agile allows for the rapid development of functionality, which can be demonstrated to stakeholders early and frequently, facilitating quick feedback and course corrections.
- **Minimum Resource Requirements:** Agile projects typically require fewer resources compared to traditional development methods, making it suitable for resource-constrained environments.
- **Suitable for Changing Requirements:** Agile is well-suited for projects with evolving or uncertain requirements, as it allows for flexibility and adaptation throughout the development process.

- **Early Partial Solutions:** Agile delivers partial working solutions early in the development cycle, providing stakeholders with tangible results and opportunities for early validation and feedback.
- **Good for Steady Change Environments:** Agile is particularly effective in environments where requirements or priorities change steadily, as it enables teams to respond quickly to evolving needs.
- **Minimal Rules and Documentation:** Agile emphasizes simplicity and minimalism, making it easy to adopt and implement without the need for extensive rules or documentation.
- **Concurrent Development and Delivery:** Agile enables concurrent development and delivery of features within an overall planned context, allowing for continuous integration and deployment.
- **Easy to Manage:** Agile projects are typically easier to manage due to their iterative nature and focus on incremental progress, providing clear visibility into project status and priorities.
- **Flexibility for Developers:** Agile gives developers the flexibility to adapt to changing requirements and priorities, empowering them to make decisions and drive the development process collaboratively.

The disadvantages of the Agile Model:

Complexity Management: Agile struggles to manage complicated dependencies between various components or features, which can lead to challenges in coordinating and integrating complex systems.

Vulnerability to Extension, Maintainability, and Sustainability: Agile projects may become more vulnerable to extension, maintainability, and sustainability issues over time, especially if proper attention is not paid to technical debt and continuous improvement.

Need for Comprehensive Strategy and Leadership: Success with Agile requires a comprehensive strategy, strong leadership from an Agile leader, and effective Agile project management practices. Without these elements in place, Agile projects may struggle to deliver desired outcomes.

Strict Delivery Management: Agile projects often rely on strict delivery management to ensure that scope, functionality, and modifications are delivered within deadlines. This can create pressure on teams and limit their flexibility in responding to changing priorities or requirements.

Dependency on Customer Engagement: Agile heavily depends on customer engagement and feedback. If customers are unclear or unavailable, it can lead to misunderstandings and misdirection within the team.

Lack of Comprehensive Documentation: Agile projects typically produce minimal documentation, which can lead to a high degree of individual dependence. This reliance on tacit knowledge can pose challenges when team members change or when onboarding new members.

Difficulty in Knowledge Transfer: Due to the minimal documentation produced in Agile projects, transferring technology and knowledge to new team members can be challenging. This lack of documentation may result in a steep learning curve for new team members, impacting project productivity and efficiency.

CHAPTER 4

SYSTEM TESTING

4.1 Exploring Features

Classified testing is a crucial Action aimed at confirming that the actual outcomes align with the anticipated results, thus ensuring the integrity of the system or equipment under scrutiny. This process serves as a means to identify and rectify any potential defects present within the system.

4.1.1 Specific Features to be Explored

A couple of the capabilities are listed below:

- Sign in
- Service information

4.2 Evaluating Methods

Your application's handling of each component is determined by the testing methodology. This is a comprehensive collection of guidelines covering every stage of a test kind and scenario. Indicate which modules should be tested.

4.2.1 Test Methodology

Testing constitutes a cornerstone of any software project, playing a pivotal role in showcasing its capabilities and functionality. Beyond mere demonstration, testing serves as a crucial mechanism for identifying and rectifying errors and glitches within the program. Consequently, users perceive the software as highly intuitive and devoid of flaws. Each function underwent rigorous independent testing and validation, ensuring its efficacy and reliability. This painstaking approach underscores our commitment to delivering software of utmost quality and performance.

4.2.2 Test Group

- Module Testing
- Integration Testing

4.3 Criteria for Success or Failure

The test engineer determines the pass/fail test criteria. It all relies on how well each specification functions. It can be determined whether the test was successful or not. In any case, I need to pass 100% of the time. An error will be assessed if the function behaves incorrectly when being evaluated.

4.4 The Trial Context

Several critical areas need to be established for testing purposes:

- **Operating System:** Ensuring compatibility and functionality across different operating systems, such as Windows, macOS, Linux, or mobile operating systems like iOS and Android.
- **Browser:** Verifying the performance and compatibility of the software across various web browsers, including popular ones like Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and others.
- **System:** Testing the software on different hardware configurations and setups to ensure optimal performance and functionality across a variety of devices and environments.
- **Application:** Conducting thorough testing of the application's features, functionalities, and user interface to identify and resolve any bugs or issues that may affect user experience.
- **Database Server:** Testing the integration and interaction between the application and the database server to ensure data integrity, security, and optimal performance.
- **Test Data:** Utilizing diverse sets of test data to evaluate the software's behavior under various scenarios, including edge cases, boundary conditions, and typical usage patterns.
- **Network:** Assessing the software's performance, reliability, and security in different network environments, including local networks, the internet, and varying network speeds and conditions.

4.5 Testing

Developing a test plan, establishing test cases and test candidates, and testing and fixing are important phases of testing. This is a cyclical process and the program goes through each phase until it reaches the desired quality level.

The check is done in the following steps,

- Unit Testing
- Validation Testing
- System Testing
- Acceptance Testing
- Regression Testing
- Database Testing
- Unit Testing

Testing individual programs is called unit testing. This is also known as software testing. This check should be performed throughout the programming phase to identify code and logic problems in each module's program. The smallest unit of a software design module is the unit under test. The user must complete every field in this project; otherwise, the user will be forced to enter the data.

- **Validation Testing**

The software must be programmed to generate valid and incorrect data, process the data and detect errors. When a user of the module accesses the page from the login her page, he must enter a username and password. If the user enters an incorrect password or uses rid, they are notified that they must enter their user ID and password.

User input is checked in this section. That is, the password is valid, the date format is correct, and the text fields are valid. Changes implemented as a result of these tests.

- **System Testing**

The term "system test" describes the process of testing the entire system "integration of all modules". Also look for discrepancies between the system and its original purpose, and between the current specification and the system's documentation. Go through the whole system and correct for deviations to achieve accuracy.

- **Acceptance Teasing**

Following functional testing, a set of business operations in the production environment are subjected to acceptance testing. Before the system is given the go-ahead to operate, this last test step must be successful. Real users and data should be used during these tests. This evaluation confirms that the system is prepared for production.

- **Regression Testing**

Regression testing involves retesting system features and parts to ensure that they continue to function properly after changes have been made. As errors are found and fixed, changes are made to the component.

- **Database Testing**

Database tests are used to check how well a database adheres to a set of standards. This is an ongoing process as no database is completely static. When you build a table, you have to build and store a mirror of it. The original should be left alone while the mirror replica is put through its paces. This process repeats until the original table is ready for changes.

4.5 Test Cases

Case No. 1 for Integration Testing Testing

Test Case ID.1	Module name: User and Provider
Test Priority: High	Test Date: 02.11.2022
Test Title: Citizen & Admin Login Verification with valid email & password	Test executed by: Sakib
Description: Test Admin & Student Login Page	Test executed date: 22.05.2021
Pre-condition:	Users and admin must have valid email and password.
Test steps:	First login page Provide valid email & password Click Sign in button
Test Data:	Email: admin@gmail.com Password: 12345678 mail: sakib@gmail.com Password:12345678
Expected Results:	User should able to login
Actual Result:	User logged in successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Logged in.

Table 4.5.1: Testing Case (User and Provider login)

Case No. 2 for Module Testing

Test Case ID.2	Module name: user
Test Priority: High	Test Date: 25-05-2021
Test Title: Add teacher with validation	Test executed by: shakib
Description: Admin can add teacher	Test Implemented date: 27-11-2022
Pre-condition:	User must login and insert valid data
Test steps:	After login go to add service Insert data all the field Click Submit button
Test Data:	Implemented all accurate data
Expected Results:	Successfully
Actual Result:	Successfully
Status (Pass/Fail):	Pass

Table 4.5.2: Testing Case (Provider)

CHAPTER 5

USER MANUAL

5.1 User Sign Page

Using their password and email address, users access this tab.

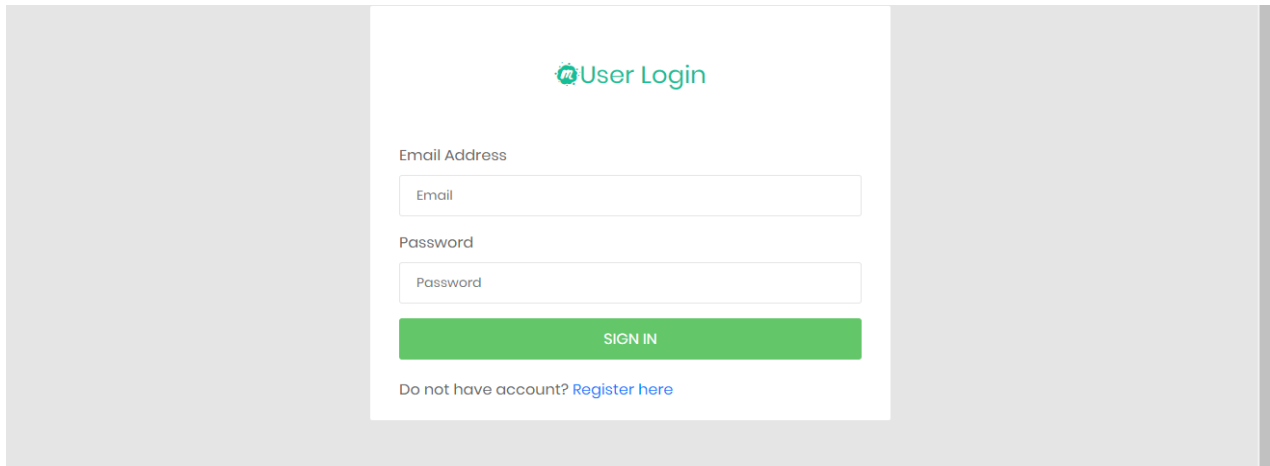


Figure 5.1 UI (User Login Page)

5.2 User Register Page

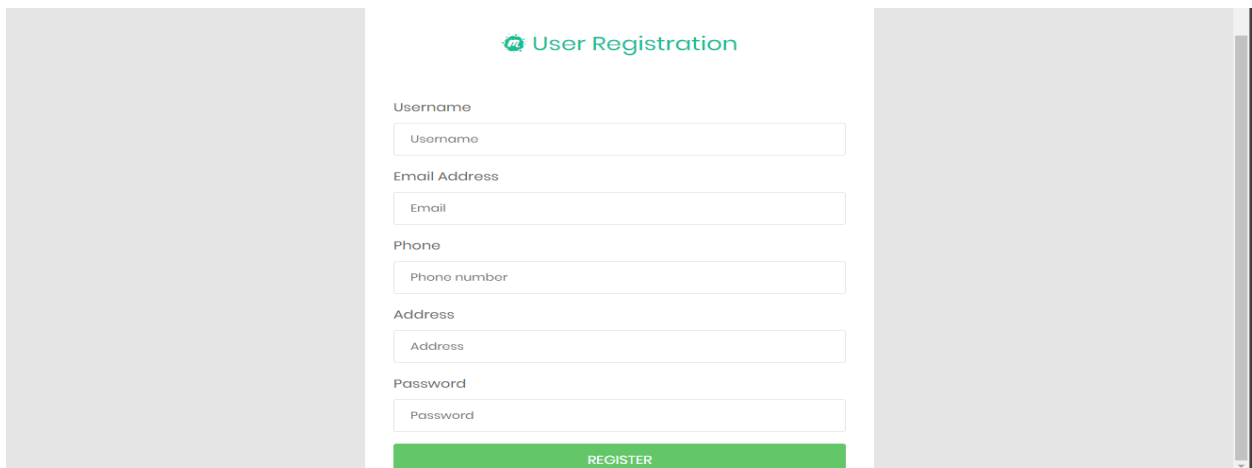


Figure 5.2 UI (User Register Page)

5.3 User Dashboard

The user has access to the entire menu and data dashboard.

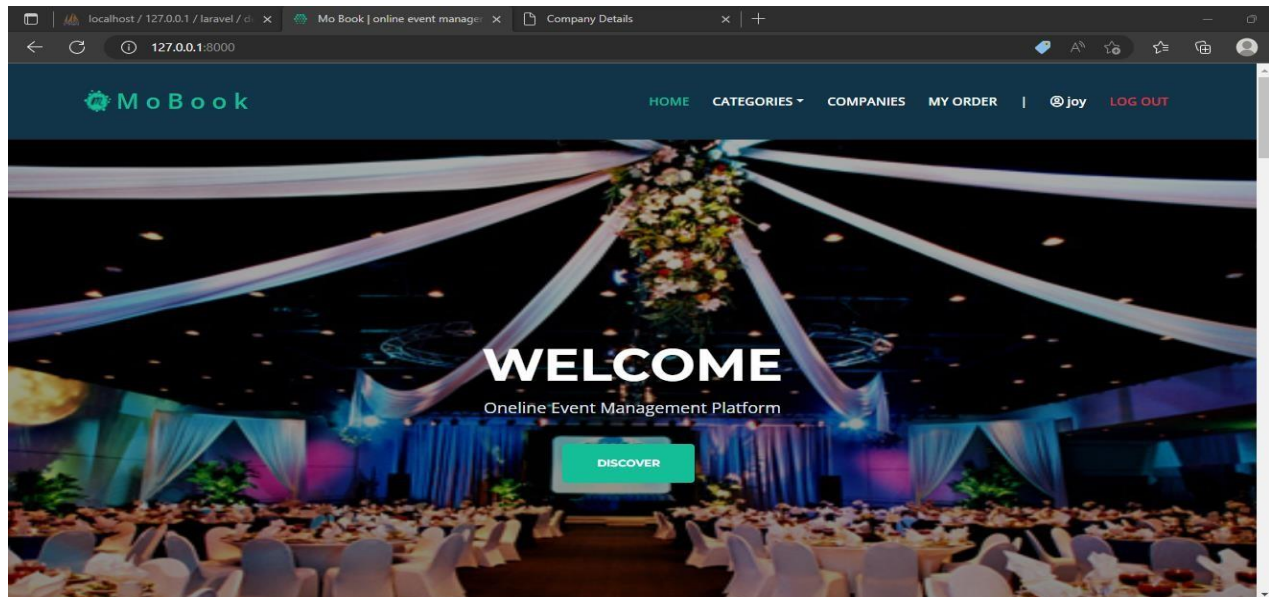


Figure 5.3 UI -User Dashboard Page

5.4 All Companies details

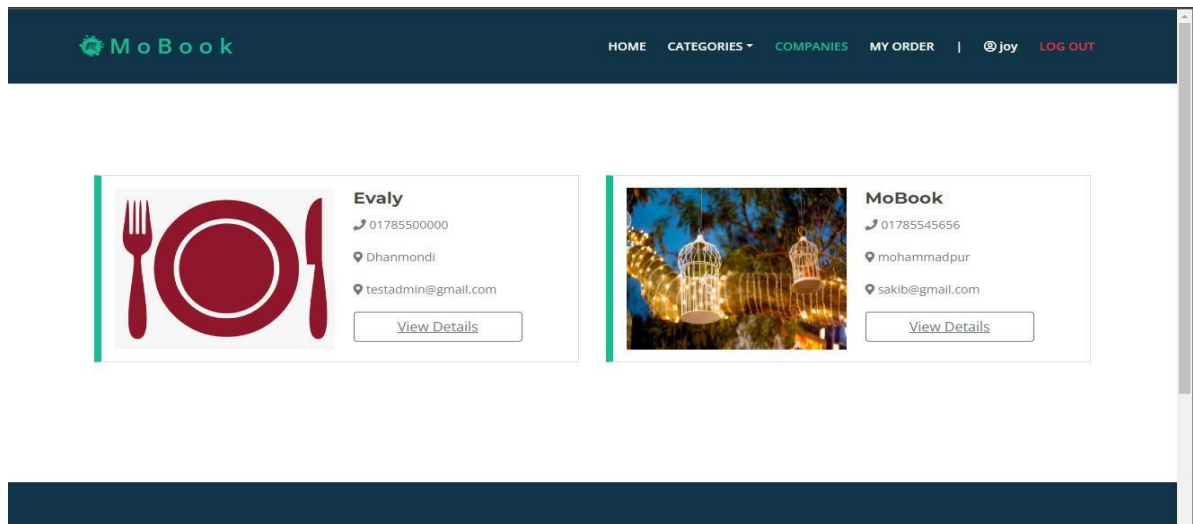


Figure 5.4 UI (All Companies details)

5.5 All Events list

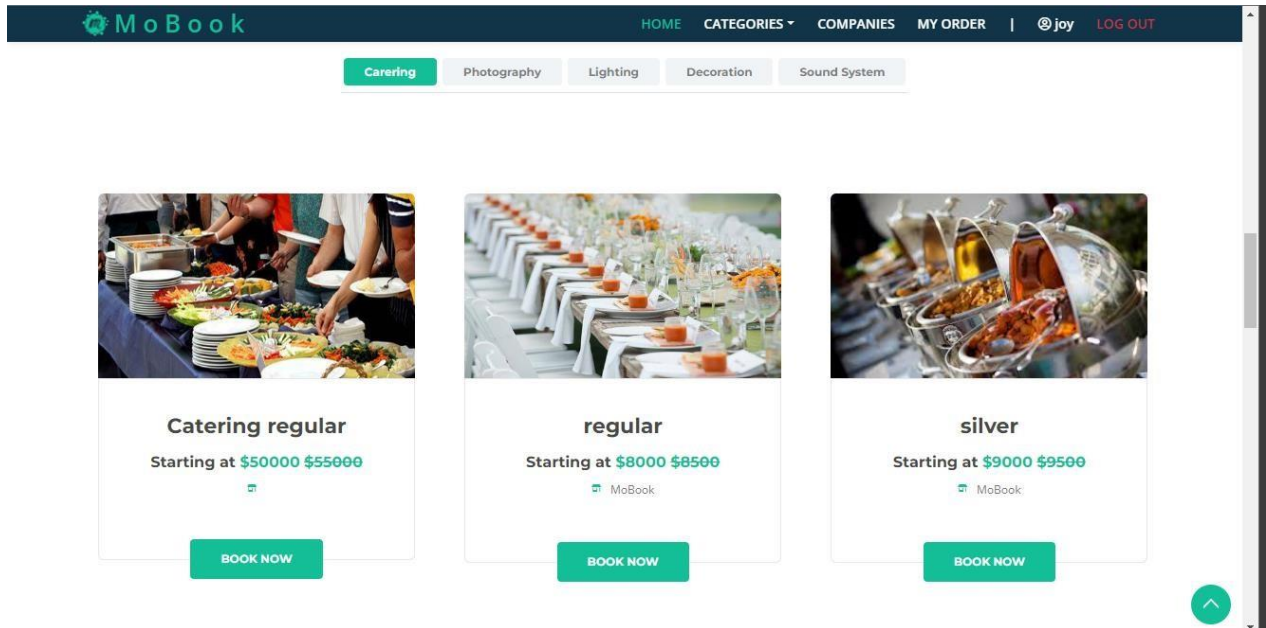


Figure 5.5 UI (All Events list)

5.6 Event Details

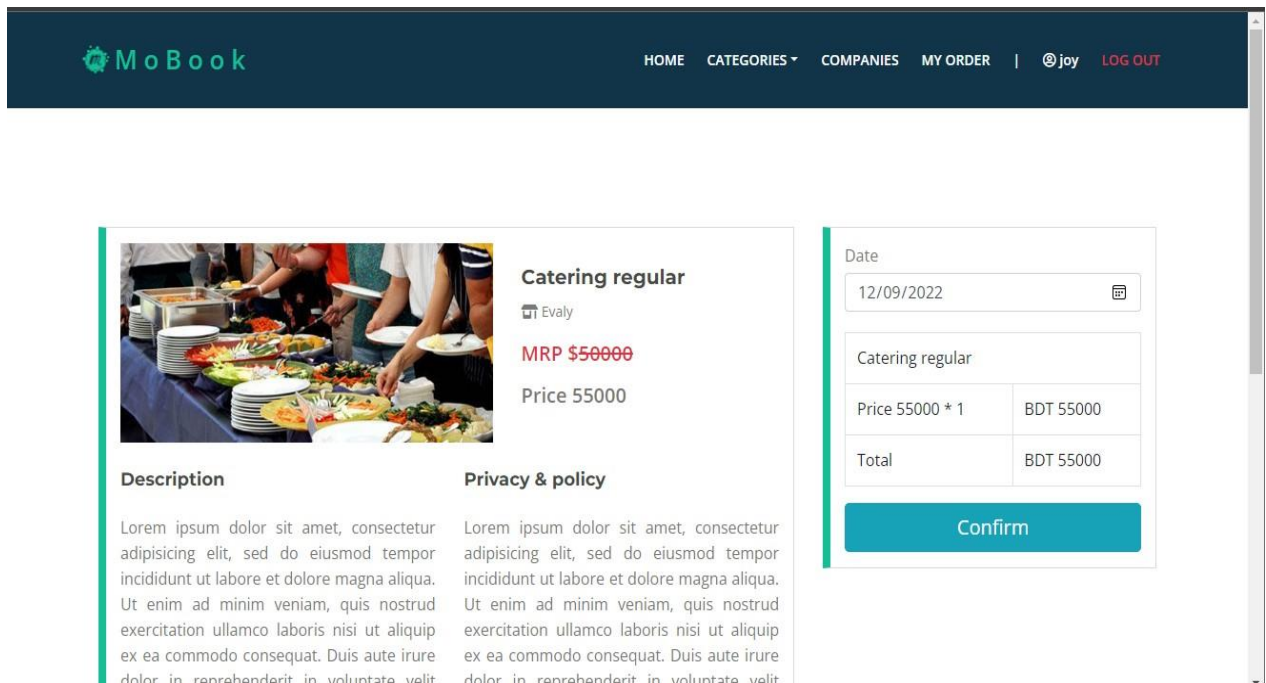


Figure 5.6 UI (Event details)

5.7 My Order page

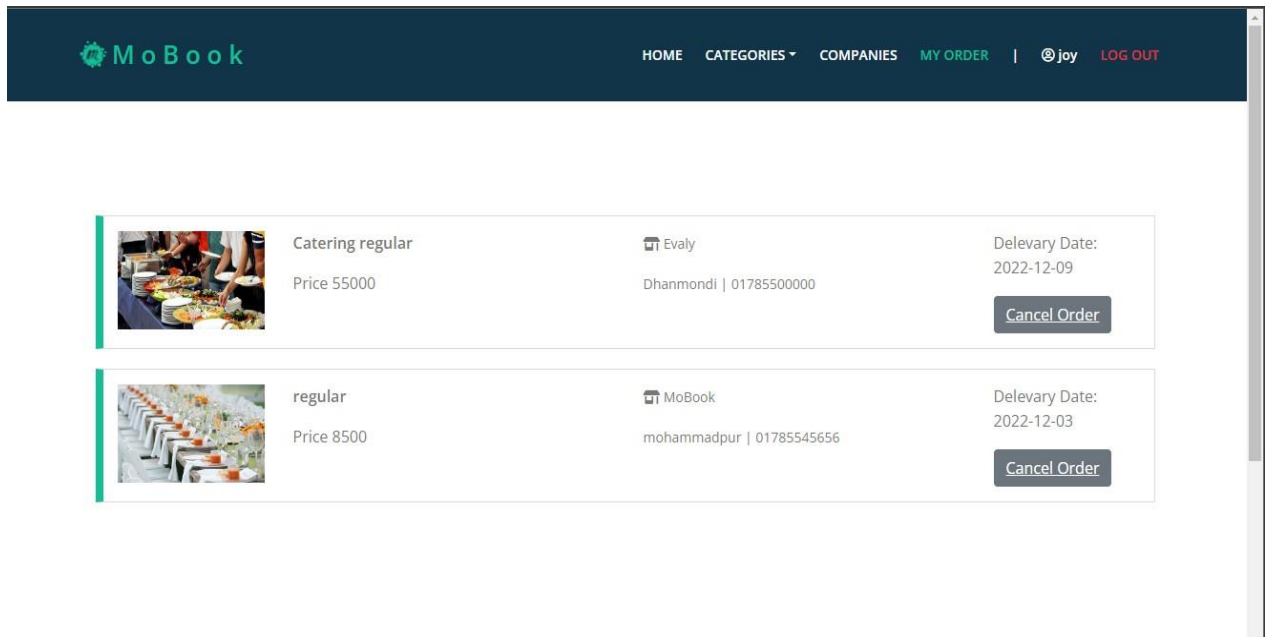


Figure 5.7 UI -My Order page

5.8 Provider registration

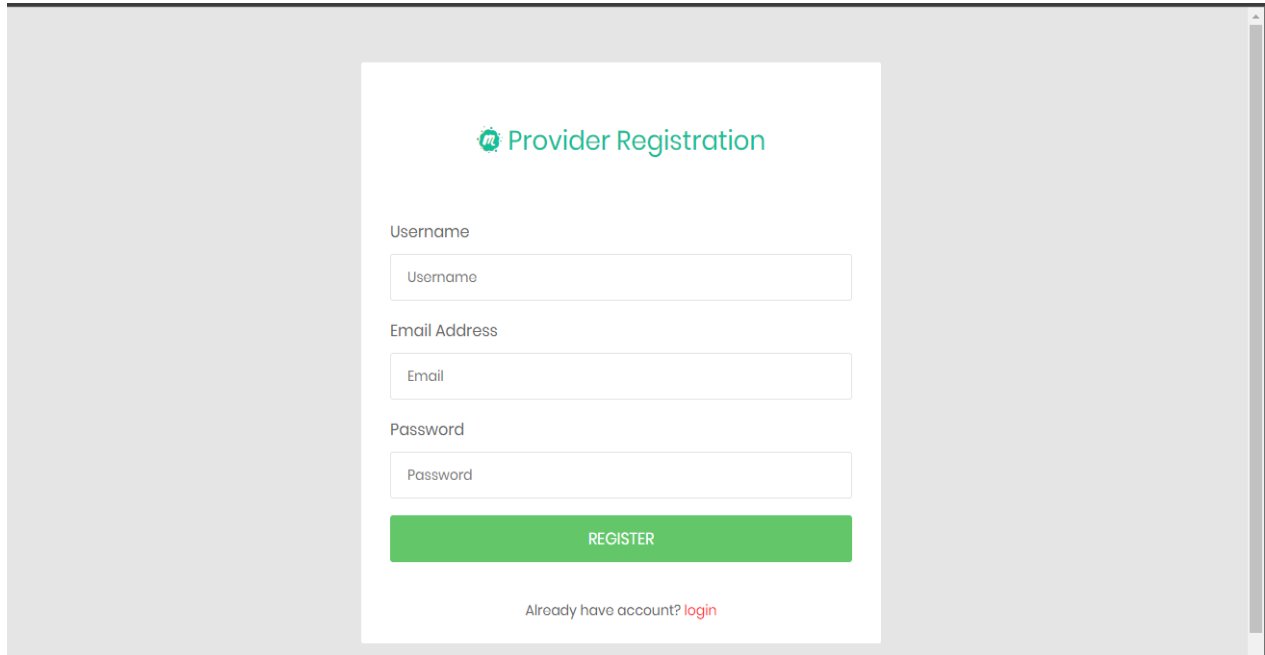


Figure 5.8 UI (Provider registration)

5.9 Provider Login

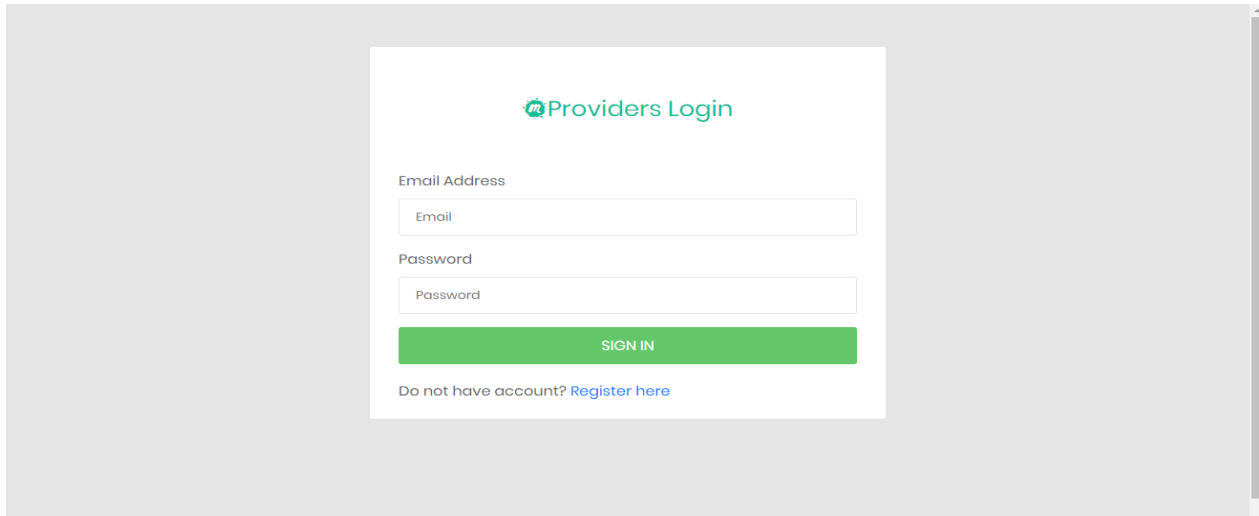


Figure 5.9 UI -Provider Login

5.10 Provider Dashboard

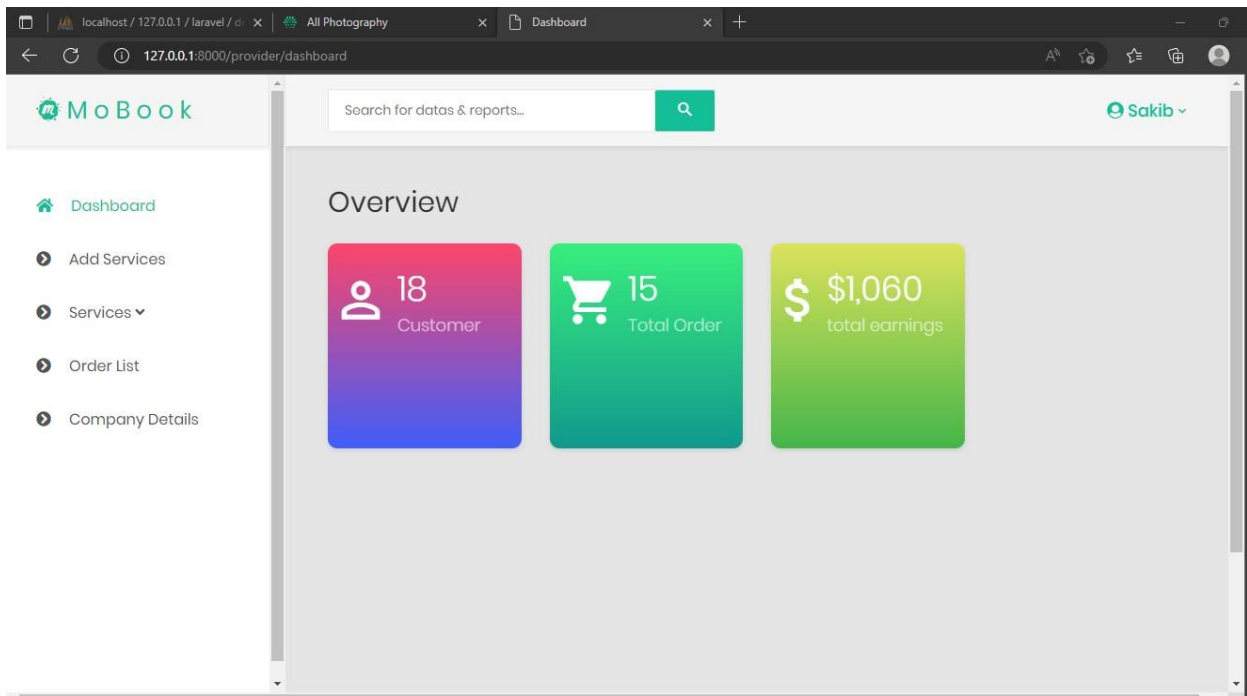


Figure 5.10 UI (Provider Dashboard)

5.11 Add Service Page

Service Name: Select

Cover Photo: Choose File | No file chosen

Package Name: [Text Input]

MRP: [Text Input]

Price: [Text Input]

Short Description: [Rich Text Editor]

Figure 5.11 UI (Add Service Page)

5.12 Services details page

Package Name	Cover Image	Price	Mrp	Action
gold		10500	10000	Enabled
silver		9500	9000	Enabled
regular		8500	8000	Enabled

Figure 5.12 UI (Services details page)

5.13 Provider Order list page

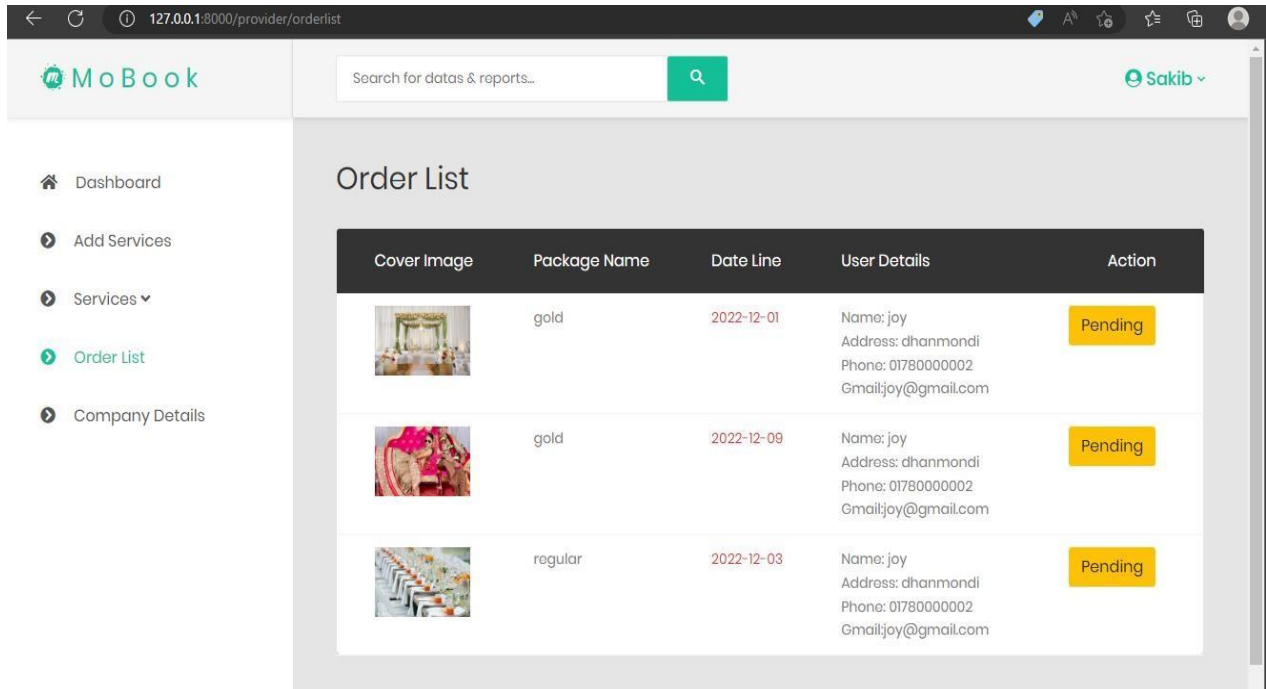


Figure 5.13 UI of Provider Order list

5.14 Add Company details page

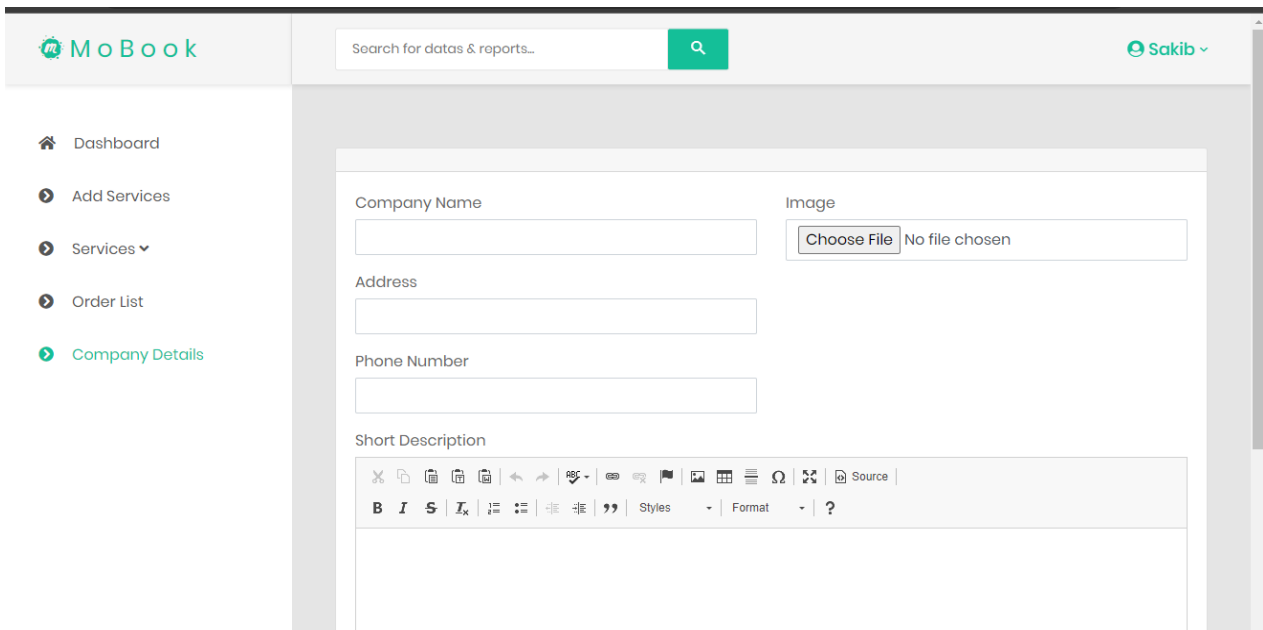


Figure 5.14 UI of Add Company Details

5.15 Company details Page

MoBook

Search for datas & reports...

Sakib

Dashboard

Add Services

Services

Order List

Company Details

Company Details

+ ADD DETAILS

Cover Image	Name	Address	Phone	Description	Action
	MoBook	mohammadpur	01785545858	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt	

Figure 5.15 UI of Company details

CHAPTER 6

PROJECT SUMMARY

6.1 GitHub Link

6.2 Limitations

Throughout my journey, I've encountered numerous obstacles that have challenged my progress. Despite my efforts to explore the requisite technologies, I find myself still grappling with these hurdles. Nevertheless, I remain steadfast in my conviction that with time, I can grasp more intricate concepts and significantly advance my coding skills.

Mobile version: Customers choose it for mobile devices since it is a travel software. We didn't have enough time or understanding about mobile devices to build a mobile version of the program.

6.3 Challenges and Successes

From the outset of this journey, we've gleaned a wealth of pertinent insights crucial for developers. Initially, I found myself utterly clueless about algorithm development, diagram creation, or crafting accurate project designs, particularly in areas such as database design. Until recently, the notion of algorithms and their importance in project development eluded me entirely. However, through diligent exploration, I've come to appreciate the intricate structure of programming languages, replete with numerous essential features. A pivotal realization dawned upon me: the indispensability of finalizing both the database architecture and user interface before delving into the logic aspect of a project. Adhering to this structured approach simplifies the code execution process significantly. In essence, the culmination of this endeavor marks a profound turning point in my career trajectory.

6.4 Future Scope

The software that is generated is excellent. I make every effort to comply with the real requirements of the procedural step. There is still room for improvement, though. I'm setting up a program and making every effort to create a workable method.

6.5 References

[1] "creately," creately, 2008-2021. [Online]. Available: <https://creately.com/>. [Accessed 2021].

[2] AdminLTE, "AdminLTE," AdminLTE, 2014-2019. [Online]. Available: <https://adminlte.io/>. [Accessed 2021].

[3] J. resig., "Jquery.," 2006. [Online]. Available: <http://jquery.com/>. [Accessed 12 09 2020].

[4] K. Sun, "Lucidchart," 2008. [Online]. Available: www.lucidchart.com/pages/. [Accessed 10 10 2020].

[5] s. c. j. k. Chad hurley, "YouTube," 2005. [Online]. Available: <https://www.youtube.com/>. [Accessed 06 08 2020].

Mobook

ORIGINALITY REPORT

23%

SIMILARITY INDEX

19%

INTERNET SOURCES

5%

PUBLICATIONS

12%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

8%

2

123dok.com

Internet Source

5%

3

Submitted to Daffodil International University

Student Paper

3%

4

Submitted to University of Greenwich

Student Paper

2%

5

www.slideshare.net

Internet Source

1%

6

Submitted to INTI University College

Student Paper

1%

7

Submitted to Caleb University

Student Paper

1%

8

Submitted to University of Wales Institute,
Cardiff

Student Paper

1%

9

Submitted to Asia Pacific Institute of
Information Technology

<1%

10	Submitted to Sim University Student Paper	<1 %
11	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
12	Submitted to Arab Open University Student Paper	<1 %
13	Submitted to Federal University of Technology Student Paper	<1 %
14	Submitted to Neath Port-Talbot College Student Paper	<1 %
15	www.top4download.com Internet Source	<1 %
16	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %
17	www.devx.com Internet Source	<1 %
18	Submitted to RDI Distance Learning Student Paper	<1 %
19	Submitted to Al Ain University Student Paper	<1 %
20	Submitted to American Intercontinental University Online Student Paper	<1 %

21

Submitted to Asia Pacific University College of
Technology and Innovation (UCTI)

Student Paper

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off