CERTIFICATE

THE KELKAR EDUCATION TRUST'S

VINAYAK GANESH VAZE COLLEGE

OF ARTS, SCIENCE & COMMERCE

(AUTONOMOUS)

Mithagar Road, Mulund(E), Mumbai – 400 081

Certified that the Project Dissertation ~~practical~~ experiments as have been duly signed by demonstrators were performed by Mr./~~Miss~~. Desai Atharv Ankush. Aishwarya of the TY-IT A Class (Roll No. 3945A004) in the INFORMATION TECHNOLOGY Laboratory of V. G. Vaze College during the year 2021 - 2023.

Rakhee P 26/09/2022 26/09/22

Staff Member in Charge

Poornima B

Head of the Department

Date : _____201

Examined

THE APPROVAL PROJECT PROPOSAL

(Note:All entries of the proforma of approval should be filled up with appropriate andcomplete information. Incomplete proforma of approval in any respect will besummarily rejected.)

PNR No : 202001641307855

1. Name of the Student	Atharv Ankush Desai
2. Title of the Project	Vaccine Managment System
3. Name of the Guide	Nanda Rupnar (Ma'am)
4. Is this your first submission?	Yes

Student Name :	Atharv Ankush Desai
Student Control ID :	20200801 89
Student Roll No. :	3844A004

17/6/22

Vaccine Management System

(Drug_Lord_v.1.10)

[Le-Vaccine]

Index

Project Synopsis 1
Title 1
Problem Statement 1
Why this Topic? 1
Objective and Scope 1
Methodology for developing project. 1
Proposed Architecture 2
Requirements 2
Software Requirements 2
Hardware Requirements 2
Platform 2
Contribution 2
Conclusion 2
1. Introduction 3
1.1 Background 3
1.2 Objective 3
1.3 Purpose 3
1.4 Application 3
1.5 Scope 3
1.6 Achievements 3
2. Survey of Technology 4
3. Requirement and Analysis 5
3.1 Problem Definition 5
3.1.1 Sub Problems 6
3.1.2 Problem Description. 6
3.2 Requirements Specification 6
3.2.1 Requirement analysis 9
3.2.2 Functional Requirements 9
3.2.3 Non-Functional Requirements 9
3.2.3 Sub- Systems 11
3.3 Planning & Scheduling 11
3.3.1 Activity Table 11
Activities 11
Start-Date 11
End-Date 11
3.3.2 Gantt Chart 13
3.4 Hardware & software requirements 14
3.5.1 Hardware 14
3.5.2 Software 14

3.5 Conceptual Model 15
3.5.1 Data Model 15
3.5.2 Data-Flow Diagram 17
3.5.3 Class Diagram 20
3.5.4 Use-Case Diagram 22
3.5.5 Sequence Diagram 26
3.5.5 State Activity Diagram 32
3.5.6 Activity Diagram 33
4. System Design 31
4.1 Interface Design 32
4.2 Test Cases 38
5. Bibliography 44
Websites 44
Reference books 45

Table of Figures

Figure No	Figure	Page No
1.	Figure 1.1 (Architecture Design)	3
2.	Figure 3.2.1 (Survey)	11
3.	Figure 3.2.2 (Survey)	11
4.	Figure 3.2.3 (Survey)	11
5.	Figure 3.2.4 (Survey)	11
6.	Figure 3.2.5 (Survey)	12
7.	Figure 3.2.6 (Survey)	12
8.	Figure 3.2.7 (Survey)	13
9.	Figure 3.1 (GANTT Chart)	16
10.	Figure 3.2 (Data Model)	19
11.	Figure 3.3 (Dataflow 0)	20
12.	Figure 3.4 (Dataflow 1)	21
13.	Figure 3.5 (Dataflow 2)	21
14.	Figure 3.6 (Class Diagram)	22
15.	Figure 3.7 (Use-case Diagram)	24
16.	Figure 3.9 (Sequence 1)	28
17.	Figure 3.10 (Sequence 2)	28
18.	Figure 3.11 (Sequence 3)	28
19.	Figure 3.12 (Sequence 4)	29
20.	Figure 3.13 (Sequence 5)	29
21.	Figure 3.14 (Sequence 6)	29
22.	Figure 3.15 (Sequence 7)	29
23.	Figure 3.16 (Sequence 8)	30
24.	Figure 3.17 (Sequence 9)	30
25.	Figure 3.18 (Sequence 10)	30
26.	Figure 3.19 (Sequence 11)	31
27.	Figure-3.20 (Sequence 12)	31
28.	Figure 3.21 (Sequence 13)	31
29.	Figure 3.22 (Sequence 14)	32
30.	Figure 3.23 (Sequence 15)	32
31.	Figure 3.24 (Sequence 16)	32
32.	Figure 3.25 (Sequence 17)	34
33.	Figure 3.26 (State Activity Diagram)	36
34.	Figure 3.27 (Activity Diagram)	36

35.	Figure 4.1 (Interface 1)	38
36.	Figure 4.2 (Interface 2)	39
37.	Figure 4.3 (Interface 3)	39
1.	Figure 4.4 (Interface 4)	39
2.	Figure 4.5 (Interface 5)	40
3.	Figure 4.6 (Interface 6)	41
4.	Figure 4.7 (Interface 7)	41
5.	Figure 4.8 (Interface 8)	42
6.	Figure 4.9 (Interface 9)	42
7.	Figure 4.10 (Interface 10)	42
8.	Figure 4.11 (Interface 11)	43
9.	Figure 4.12 (Interface 12)	43

List of Tables

Figure No	Table	Page No
1.	Table 3.1 (Activity & Planning)	15
2.	Table 3.2 (Dataflow notations)	23
3.	Table 3.3 (Class Diagram notations)	27
4.	Table 3.4 (Use-Case Diagram notations)	33
5.	Table 3.5 (Sequence Diagram notations)	35
6.	Table 3.6 (State Machine notations)	37
7.	Table 3.7 (Activity State Diagram notations)	39
8.	Table 4.1 (Test Case)	44

Synopsis

Project Synopsis

Title.

Vaccine Management System Codename: DrugLordv1.10

Problem Statement

The main objective of this System is to maintain records of vaccinations to monitor the quality of vaccines and timely assurance in rural areas where health centres are not easily accessible.

Why this Topic?

To reduce the paperwork required for maintaining records instead digitizing them for data visualization to minimize reading of records and provide easy access to third-party bodies to get, deliver, and administer vaccines.

Objective and Scope.

- To reduce the paperwork required
- To maintain a detailed record of each individual with ease
- To analyse data easily for the person/body using this system
- To create a system that manages both deliveries along with a facility for vaccine administration

Methodology for developing project.

In this system, I am going to use Extreme Programming for developing an appropriate system as a solution for rapidly changing requirements

Advantages: Communication, Simple, Easy, Agile.

Proposed Architecture

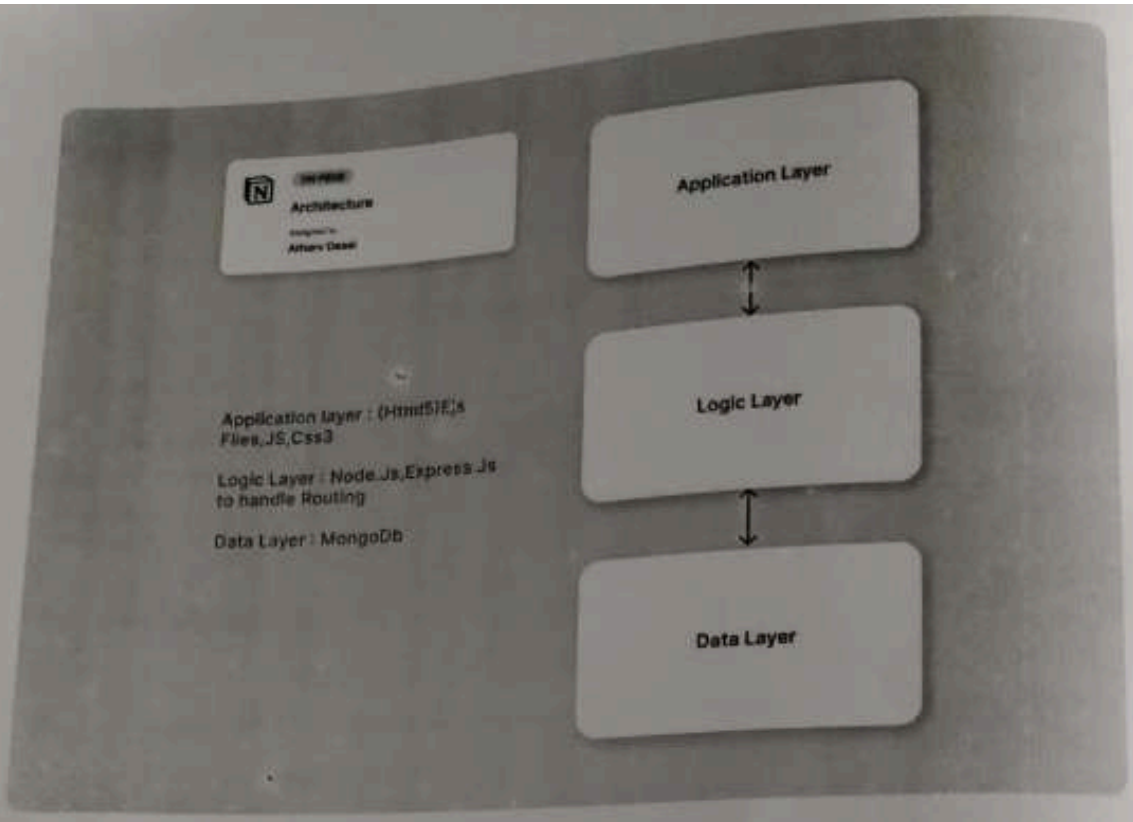


Figure 1.1 Architecture Design

Requirements

Software Requirements

- Front-end : HTML5, CSS , JS,Tailwindcss
- Back-end :,ExpressJs, NodeJS, MongoDb.
- Operating System: Windows 7.0 +.

Hardware Requirements

- Processor: Intel Core Duo 2.0 GHz or more.
- RAM: 2 GB or more.
- Monitor: 17 CRT or LCD.
- Hard disk: 500 GB or more.
- Keyboard: Normal or multimedia.

Platform

Visual Studio Code

Contribution

This system will help people in rural areas get access to vaccines along with reduced paperwork and record generation and keep each individual to date.

Conclusion

This system will help to reduce paperwork and provide access to people in rural areas.

1. Introduction

1.1 Background

The current Vaccination Systems which are used are not centralized i.e., the data generated is confined to an organization or a company except due to a recent pandemic newer methods are cultivated

References

- https://www.servicenow.com/solutions/vaccine-management.htm
- https://www2.deloitte.com/us/en/pages/public-sector/solutions/vaccine-management-system.htm
- https://www.intelex.com/products/applications/vaccine-management} {Intelex Vaccine Solution

1.2 Objective

The main purpose of this project is to help benefit NGOs in providing vaccine access to rural areas people without using many resources, by providing verified personnel of NGOs or any individual to request access to medical vaccines under the guidance of a medical practitioner.

1.3 Purpose

- Provide rural areas with exposure to vaccine
- Create a platform where people can register for an appointment, and delivery as well as monitor the records.

1.4 Application

The idea can be fundamentally used in any management system of products in any medical field.

1.5 Scope

Creating a platform for all domestic producers to sell their vaccines to local consumers, NGOs, etc, and to simplify the delivery process, regulations and management.

1.6 Achievements

Through this project, I will get to learn various technologies used to develop a system. This system will give a digital platform to the all-Domestic producers of vaccines which will increase the vaccine production rate and simplify the delivery process as well as help them to manage their data. Once they get satisfactory results from the system, users will start trusting the system and eventually, most of the tasks will get digital. Encourage B2B and C2C relations

2. Survey of Technology

The number of Technologies available for the implementation is listed below

- ASP.NET(MVC) Core 5.0+
  - Is a framework developed by Microsoft to create enterprise-grade web apps in a short amount of time
  - Languages : C#, HTML, CSS, jQuery
  - Backend: Microsoft SQL Serve
- Django (MVC)
  - Framework to create scalable single-page web apps,web-API using Python as the main programming language
  - Language : Python,HTML,CSS,jQuery,JS
  - Backend: MongoDB
- Ruby on Rails
  - Framework based on Rudby Programming Language for easy file directory structure and dataflow subroutines
  - Language: Ruby,HTML,CSS,JS
  - Backend: MySql Server
- Node.js
  - Backend as well and Frontend using vanilla Javascript as main primary language useful in API creation
  - Language: JS (EMAC5), HTML(ejs), JS,jQuery
  - Backend: MongoDB
- Flask
  - Python Backend Framework, useful in developing fast prototypes of ML and AI searches, etc
  - Language: Python

For my current project, I am going to use Node.js as a development platform for easy implementation of the requirements proposed.

Why Node.js?

Easy routing through Express.JS library and ejs content to render with minimum requirements to hosting a server and is platform independent

Required: Node.js is installed

3. Requirement and Analysis

3.1 Problem Definition

What to expect about the system?

The system will manage and record all the deliveries along with orderly administration of vaccines It will also keep a detailed record of all the domestic suppliers, users requesting appointments, providers buying vaccines, and local company producers producing vaccines

- [Users, Providers, Producers]
    - Will be allowed to log in as each unique individual account.
- [Producers, Providers]
    - Help manage, and administer vaccine drives with all the information centralized for easy fetch
    - The system will keep a detailed record of all individual's information both registered for vaccine drives as well as seeking appointments
- [Provider]
    - To set up appointments for the user or use for delivery purposes.
- [Producer]
    - Set up vaccine stocks for order processing
- [User]
    - Book a nearby appointment for quick administration

3.1.1 Sub Problems

- For Users
    - How users can add their information?
    - How to schedule and select an appointment?
    - How to get the vaccine?
- For Providers
    - How to get verified?
    - How to select your order?
    - How do schedule appointments for the people?
- For Producers
    - How to set up your vaccine Stocks?
    - How to process Orders?
    - How to Deliver?

3.1.2 Problem Description.

The system will help find required clients for domestic producers as well as enable recipients to get vaccine appointments in their nearby locality

3.2 Requirements Specification

3.2.1 Requirement analysis

Identify stakeholders i.e., in case the people who are going to use this app

- Capture requirements
- Holding One-One interviews
- Conduct Group Workshops
- Get Feedbacks
- Build Small Prototypes

For the current situation, I used the Feedback method to identify the requirements for the project using Google Forms as a means to collect the data

The below figures are the collected data that was generated.

Responses were based on:

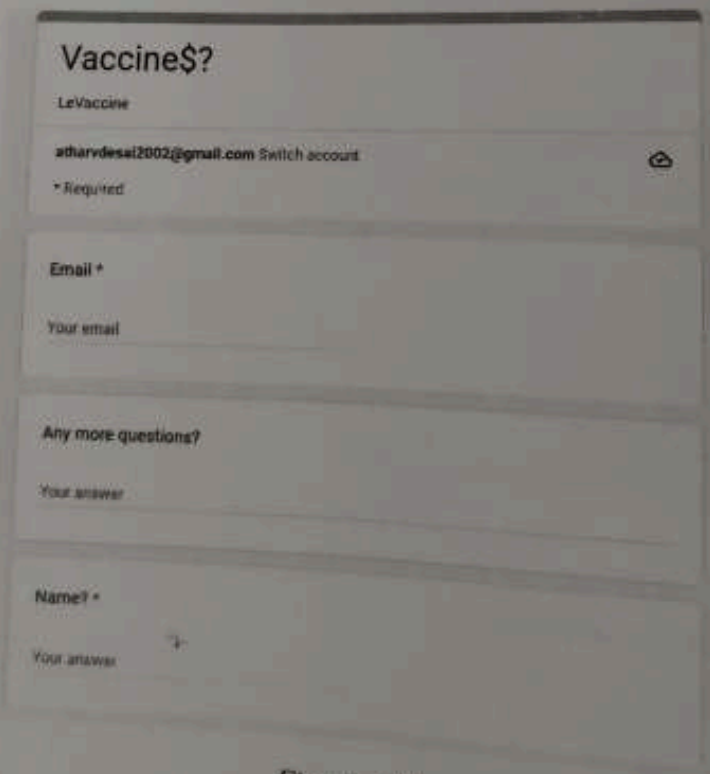


Figure 3.2.1

Was the registration process online ?(expect for CoWin) *

- Yes
- No

Ever had any medical query registration except CoWin?

- Yes
- No
- Maybe

(Start from img) there was a webapp/website which could manage deliveries /orders /vaccination adminstration / and visualise it and scalable would I have your attention?

Figure 3.2.2

Phone?

Your answer

Ever had a Vaccine? *

- Yes
- Maybe
- No

Getting appointments was time consuming?

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

Figure 3.2.3

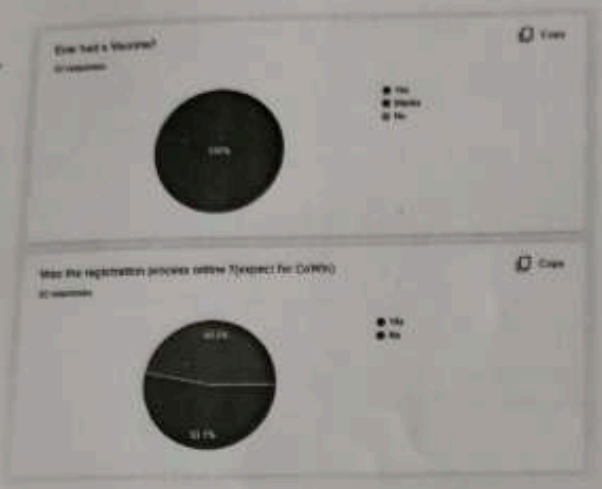


Figure 3.2.4

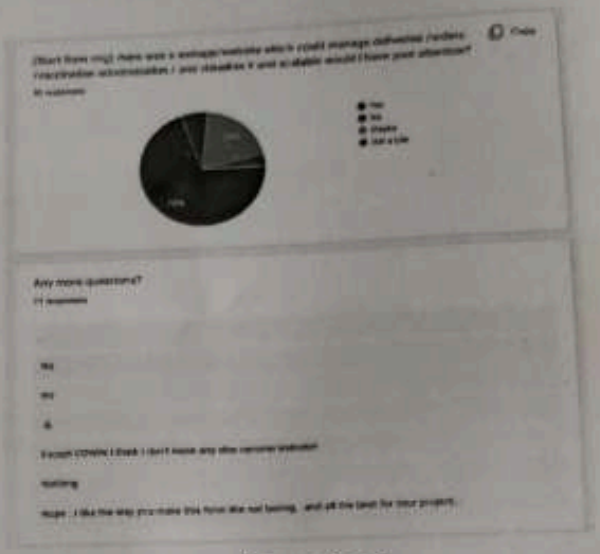


Figure 3.2.5

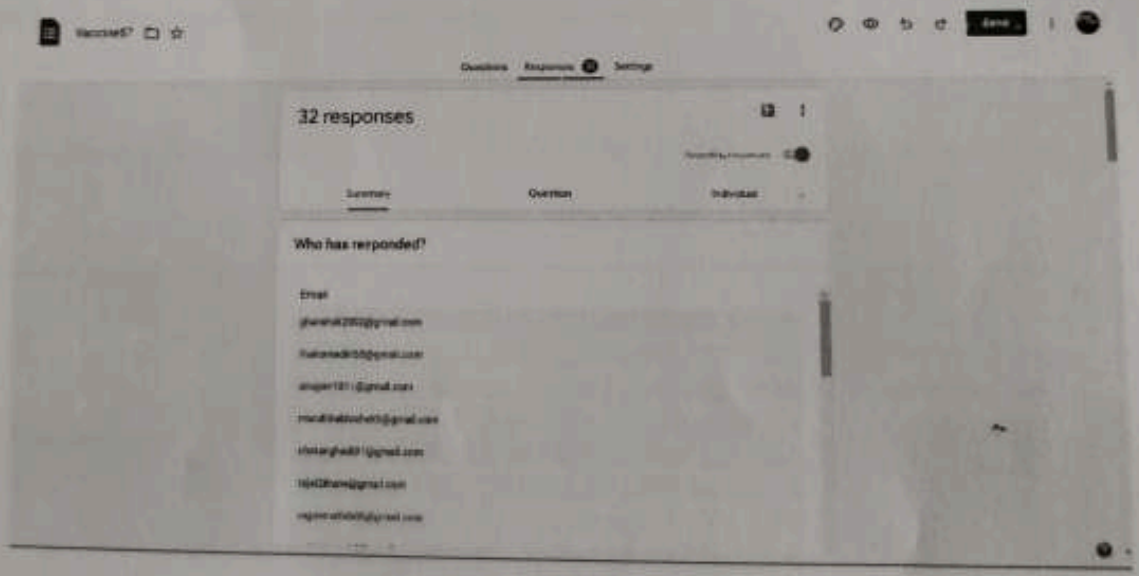


Figure 3.2.6

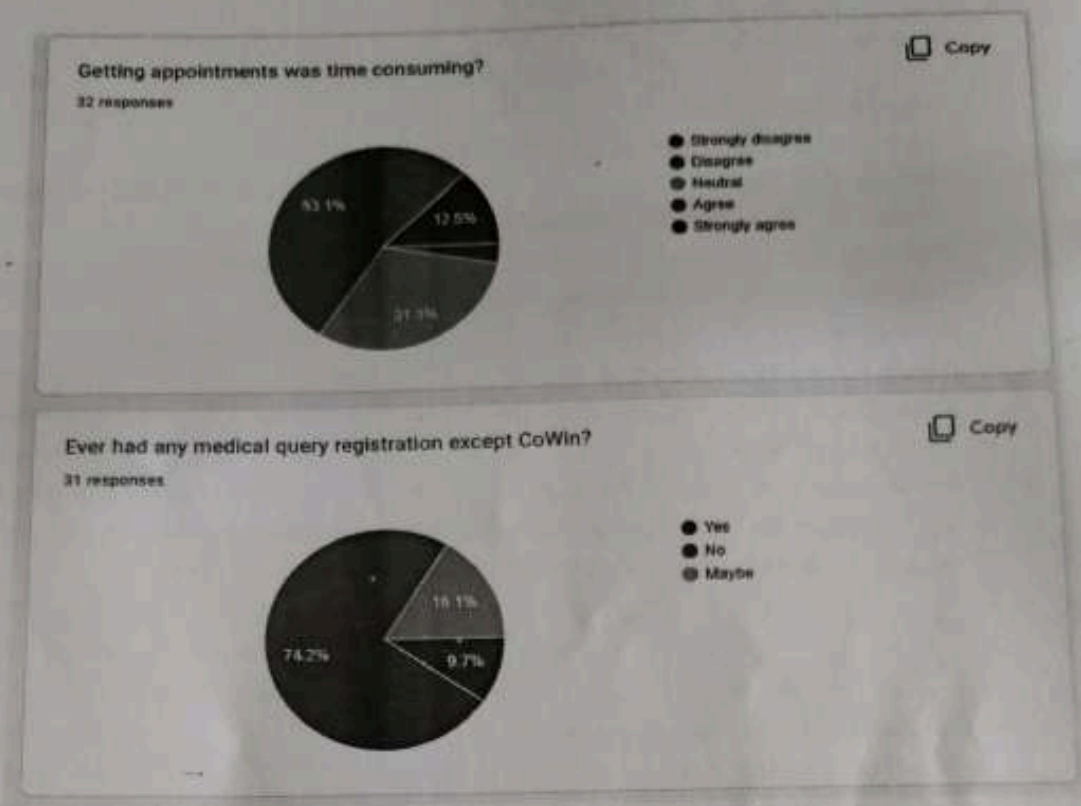


Figure 3.2.7

3.2.2 Functional Requirements

1. The system should allow users to fill up for an appointment in their locality
2. The system should keep a detailed record of user's info
3. The system should allow users to note any problems after vaccination is completed
4. The system should provide Producers, NGOs, and Companies to order and sell their vaccines
5. The system should keep a detailed record of all domestic producers, providers and along with the info of users

3.2.3 Non-Functional Requirements

1. Usability: Should be user-friendly and only required detail should be shown in a minimal way
2. Reliability: The system should be user-friendly to use.
3. Scalability: To increase the load of data handling
4. Flexibility: can run on any Platform

3.2.3 Sub- Systems

- Login

Function: Used for login
Input: User, Provider, Producer Login Details
Pre-Condition: None
Source: From User, Provider, Producer
Post-Condition: Redirected to Dashboard if Credentials are valid
Exception: An error message is shown if credentials don't match

- Register

Function: Used for registering an account
Input: User, Provider, Producer all details
Pre-Condition: None
Source: From User, Provider, Producer
Post-Condition: Redirected to log in if details are valid and correct
Exception: An error message is shown if any field is wrong or not valid

- Users-Info Update

Function: Used for updating account
Input: User, Provider, Producer all details to change
Pre-Condition: None
Source: From User, Provider, Producer
Post-Condition: The user's details are updated
Exception: The error message is shown if any field is wrong or not valid

- Post-Review

Function: Used for Posting Reviews
Input: User Reviews
Pre-Condition: None
Source: From User
Post-Condition: User's reviews get added
Exception: The error message is shown if anything goes wrong

- Book Appointment

Function: Used for booking an appointment
Input: User to book appointment
Pre-Condition: None
Source: From User
Post-Condition: Appointment is booked
Exception: The error message is shown if anything goes wrong

- Set appointments

Function: Used for setting appointments
Input: Provider, Producer all details for setting the appointment
Pre-Condition: None
Source: Provider
Post-Condition: Appointment is generated
Exception: The error message is shown if anything goes wrong

- Buy Vaccines
  Function: Used for Buying vaccines
  Input: Provider all details for buying vaccines
  Pre-Condition: None
  Source: Provider
  Post-Condition: Order is Placed
  Exception: An error message is shown if anything goes wrong

- Update Vaccine Stocks
  Function: Used for updating vaccine stocks
  Input: Producer all details of vaccine
  Pre-Condition: None
  Source: Producer
  Post-Condition: Vaccine Stocks are updated or created
  Exception: The error message is shown if anything goes wrong

3.3 Planning & Scheduling

3.3.1 Activity Table

Activities	Start-Date	End-Date
Synopsis	27/04/22	16/06/22
1. Introduction	20/06/22	25/06/22
2. Survey of Technology	20/06/22	25/06/22
3. Requirement and analysis	27/06/22	15/09/22
3.1 Problem Definition	4/07/22	9/07/22

3.2 Requirement Specification	11/07/22	16/07/22
3.3 Planning & Scheduling	18/07/22	23/07/22
3.4 Hardware & Software Requirements	18/07/22	23/07/22
3.5 Conceptual Models	26/07/22	15/09/22
3.5.1 Data Model	26/07/22	1/08/22
3.5.2 Data Flow	1/08/22	22/08/22
3.5.3 Class Diagram	22/08/22	29/09/22
3.5.4 Use Case Diagram	29/09/22	12/09/22
3.5.5 Sequence Diagram	12/09/22	15/09/22
3.5.6 State Machine Diagram	12/09/22	16/09/22
3.5.7 Activity Diagram	13/09/22	19/09/22
4. System Design	19/09/22	22/09/22
4.1 Interface Design	17/09/22	20/09/22
4.2 Test Cases	18/09/22	22/09/22

Table 3.1

3.3.2 Gantt Chart

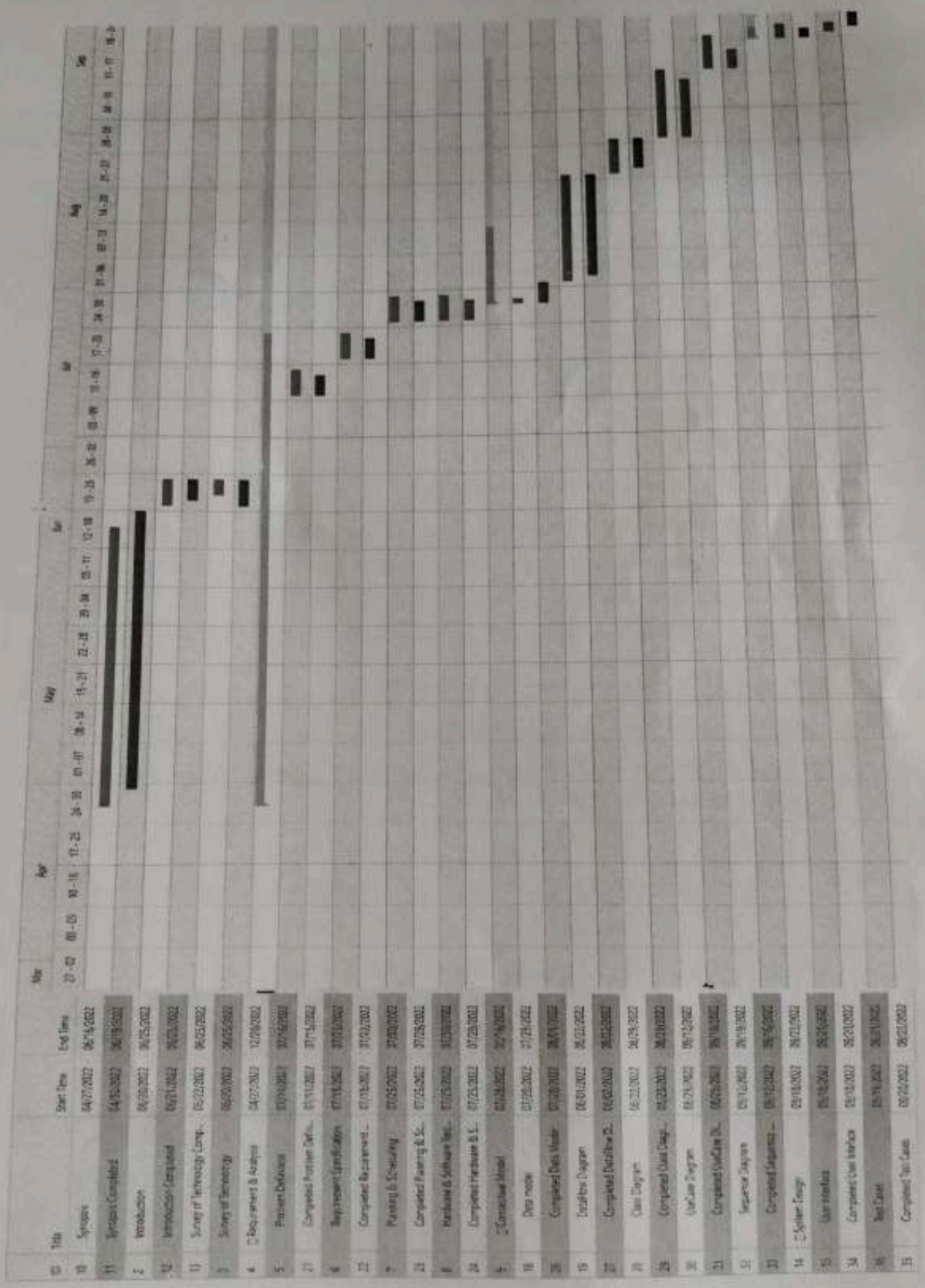


Figure 3.1 GANTT Chart

3.4 Hardware & software requirements

3.5.1 Hardware

- Processor: Intel Core 3.0 2.3 GHz or more.
- RAM: 4GB or more.
- Monitor: 17 CRT or LCD, Plasma, etc.
- Hard-Disk: 256 or more (SSD preferable)
- Keyboard: Normal or multimedia.
- Mouse: Compatible

3.5.2 Software

- System O.S: Window or Linux (Debian or Arch).
- Frontend: HTML(ejs),JS, TailwindCss.
- Backend: Node.js,Express.js
- Database: MongoDB (NoSql)

3.5 Conceptual Model

3.5.1 Data Model

Unlike SQL databases, where you must determine and declare a table's schema before inserting data, MongoDB's collections, by default, do not require their documents to have the same schema. That is:

- The documents in a single collection do not need to have the same set of fields and the data type for a field can differ across documents within a collection.

Embedded Data

Embedded documents capture relationships between data by storing related data in a single document structure. MongoDB documents make it possible to embed document structures in a field or array within a document. These *denormalized* data models allow applications to retrieve and manipulate related data in a single database operation.

The key decision in designing data models for MongoDB applications revolves around the structure of documents and how the application represents relationships between data. MongoDB allows related data to be embedded within a single document.

References :

- Practical MongoDB: Architecting, Developing, and Administering MongoDB Author :Shakuntala Gupta Edward & Navin Sabharwal,
- https://www.mongodb.com/docs/manual/core/data-modeling-introduction/

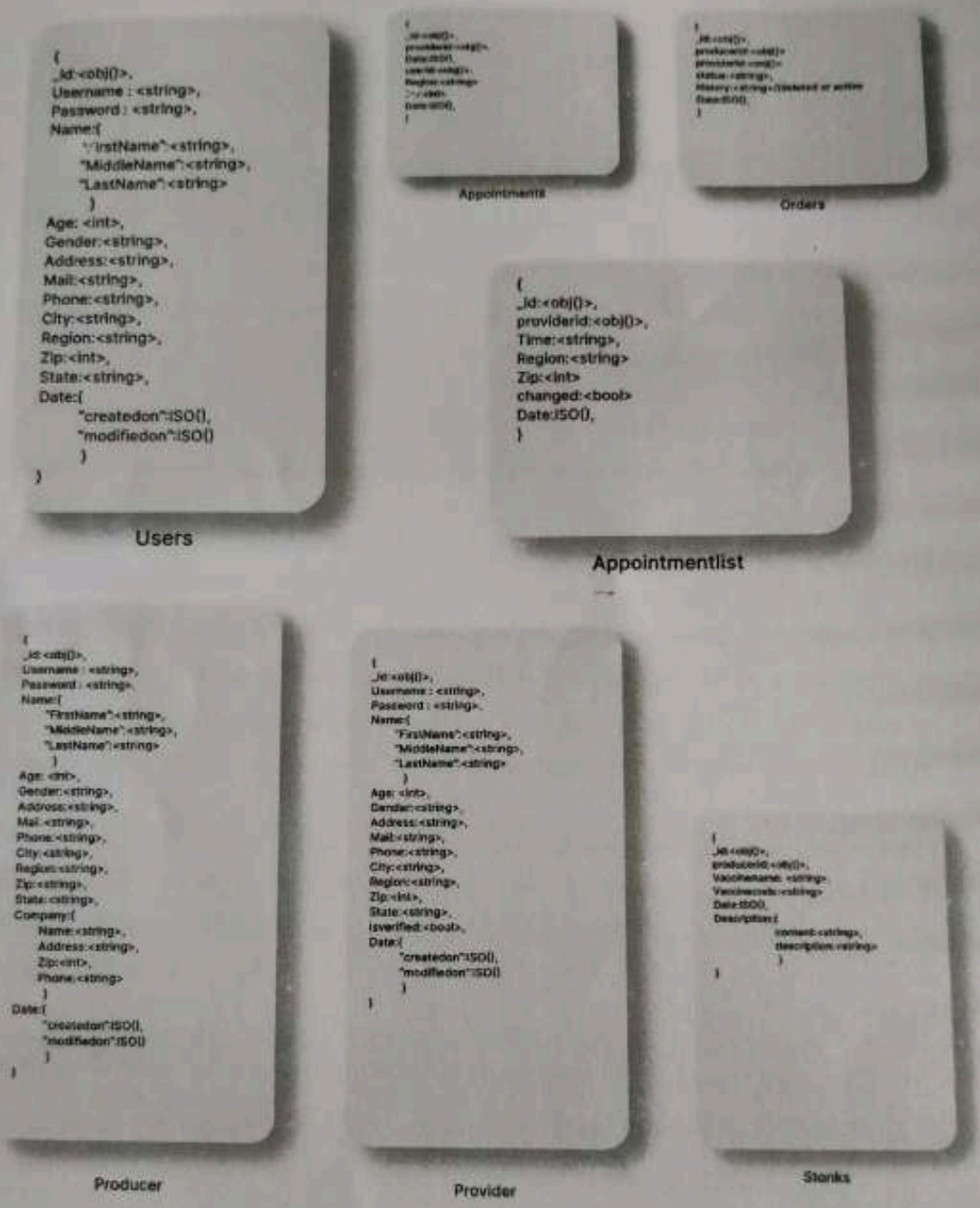


Figure 3.2 Data Model

3.5.2 Data-Flow Diagram

A **data flow diagram** (or DFD) is a graphical representation of the information flow in a business process. It demonstrates how data is transferred from the input to the file storage and reports generation. By visualizing the system flow, the flow charts will give users helpful insights into the process and open up ways to define and improve their business.

References:

- Software Engineering 9th Edition Author : Ian Sommerville
- https://www.lucidchart.com/pages/uml-data-flow-diagram

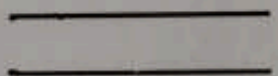
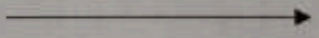
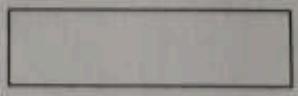
Name	Symbol	Description
Process		A process transforms incoming data flow into outgoing data flow.
Database		Data stores are repositories of data in the system.
Data Flow		Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
External Entity		External entities are objects outside the system, with which the system communicates

Table 3.2

DFD 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analysed or modelled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

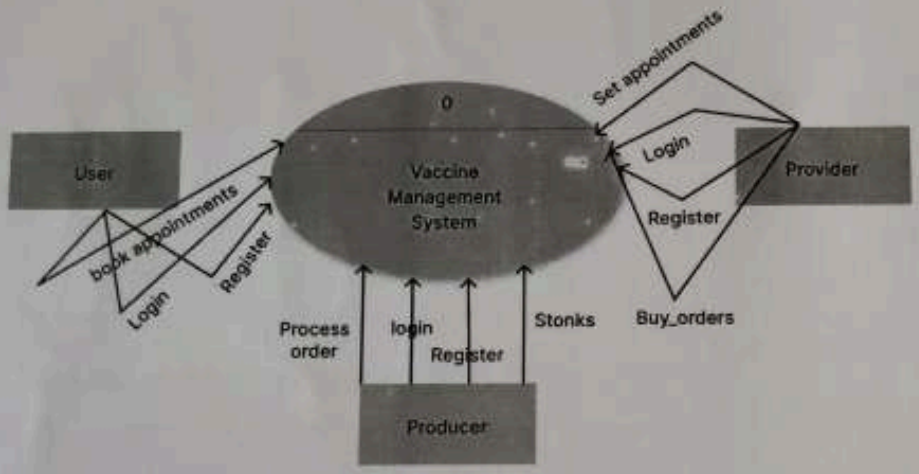


Figure 3.3 DFD 0

DFD 1

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses

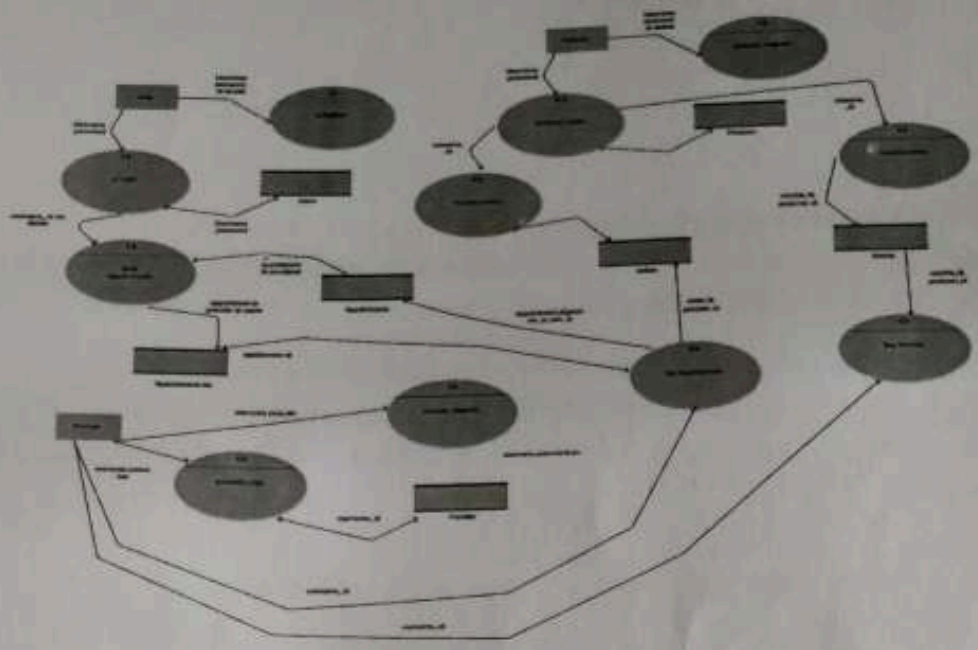


Figure 3.4 DFD1

DFD 2

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

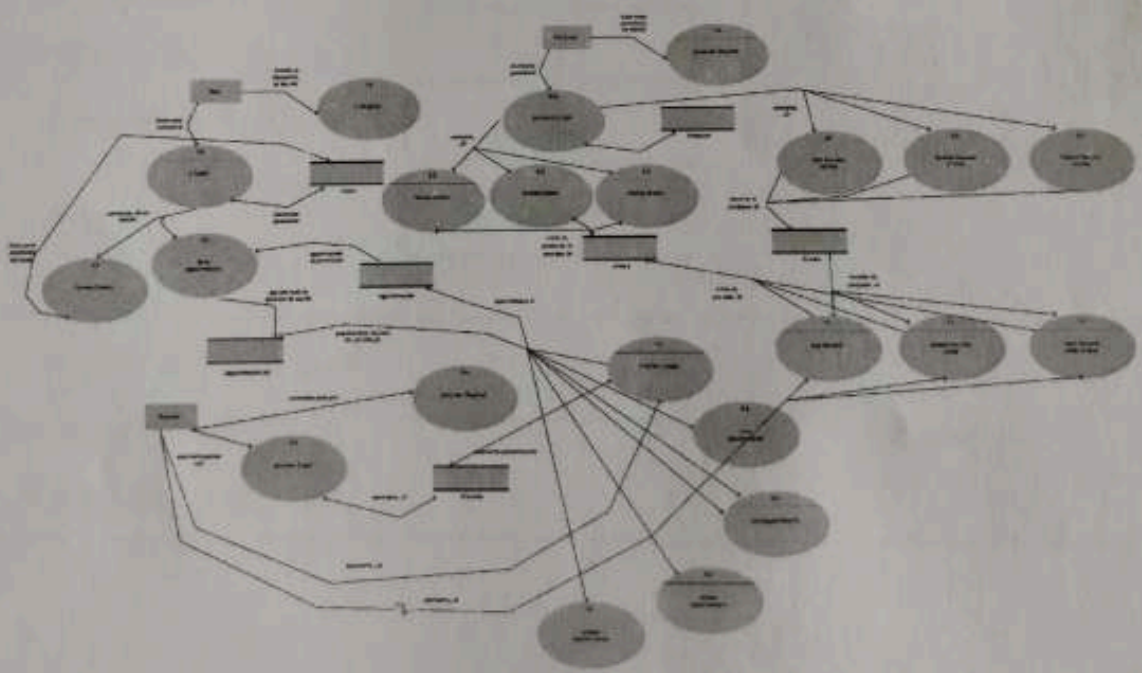


Figure 3.5 DFD 2

3.5.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

References:

- Software Engineering 9th Edition Author : Ian Sommerville
- https://www.lucidchart.com/pages/class-diagram

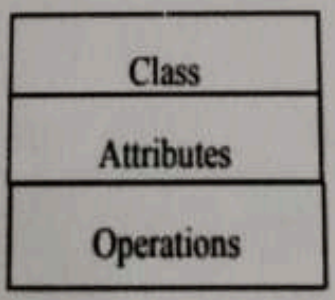
Name	Symbol	Description
Class		Classes and interfaces in UML show architecture and features of the designed system.
Association		Represents the static relationship shared among the objects of two classes

Table 3.3

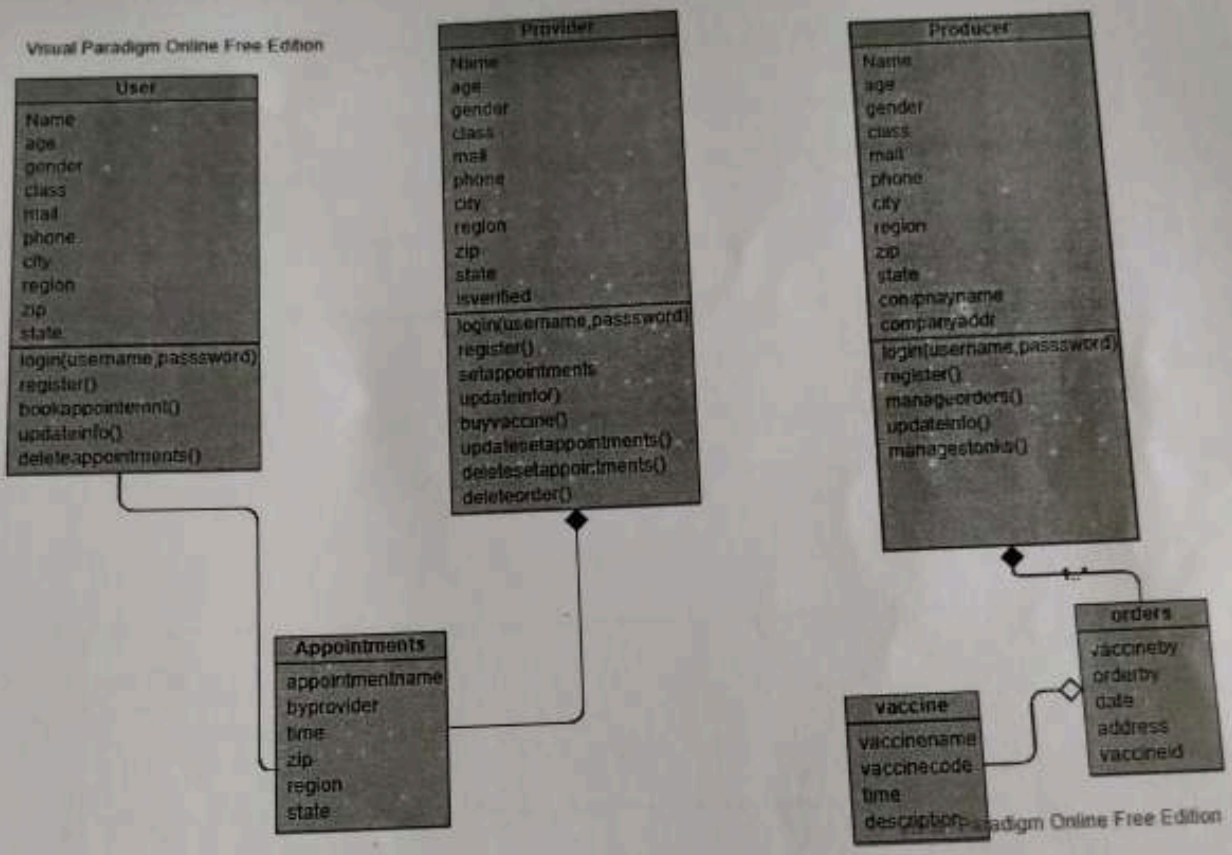


Figure 3.6 Class Diagram

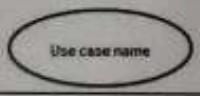
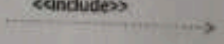
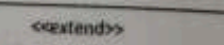
3.5.4 Use-Case Diagram

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

References:

- Software Engineering 9th Edition Author : Ian Sommerville
- https://www.lucidchart.com/pages/uml-use-case-diagram

	Actor
	Use case
	Generalization symbol used between actors and between use cases
	Association between actor and use case
	Include relationship between use cases
	Extend relationship between use cases

Symbols in a use case diagram

Table 3.4

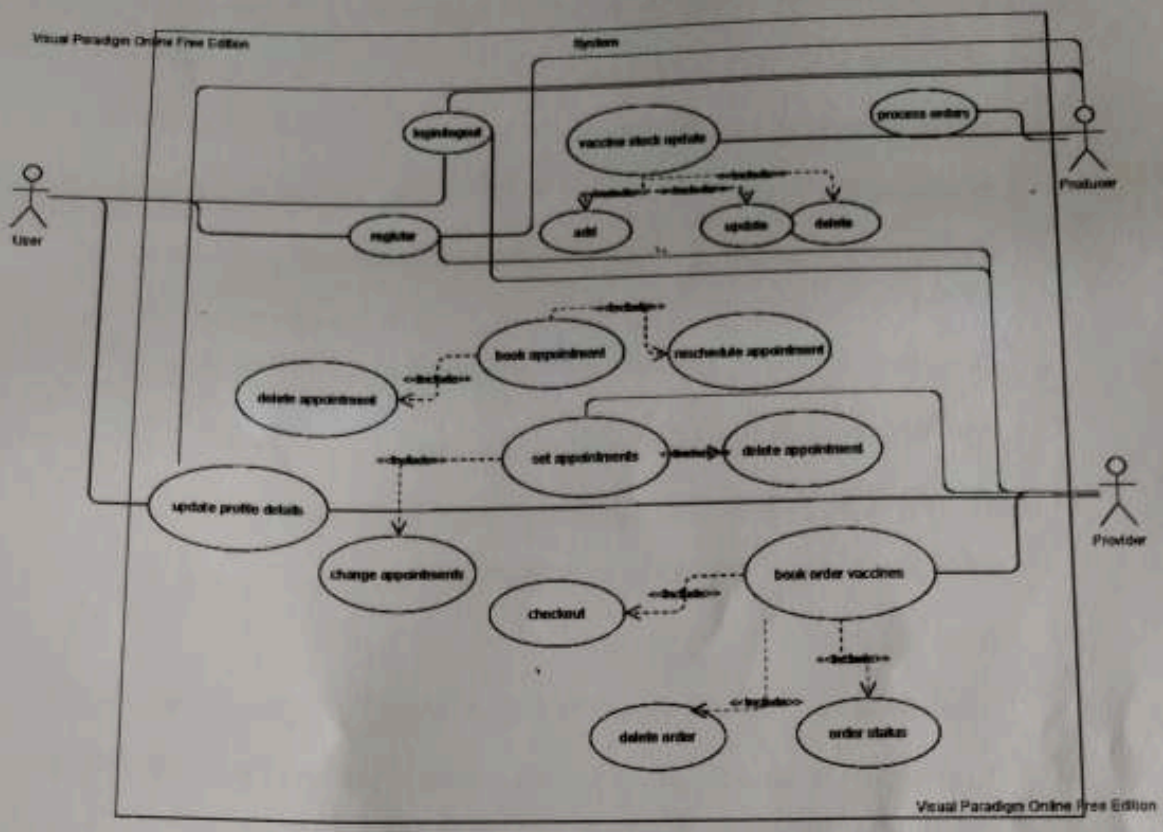


Figure 3.7

Use Case Description:

- Login

  Summary: Used to log in and Logout of the system

  Actor: User, Provider, Producer.

  Pre-condition: None.

  Description: The System waits for the credentials to be entered for validations and necessary for logout

  Exception: If credentials are not valid an error message is thrown

  Post-condition: Display user dashboard

- Register

  Summary: Used for registration

  Actor: User, Provider, Producer.

  Pre-condition: None.

  Description: Is used for creating an account in the service by fill certain details

  Exception: If proper fields are not valid an error message is thrown

  Post-condition: redirect to login

- Book an appointment

  Summary: Used for booking an appointment

  Actor: User.

  Pre-condition: The user should be logged in with proper account details.

  Description: Is used for creating an account in the service by fill certain details

  Exception: If slots get filled during registration error message is shown

  Post-condition: Appointment is booked.

- Book orders

  Summary: Used for buying vaccines.

  Actor: Provider.

  Pre-condition: Provider should be logged in and valid

  Description: For buying vaccines from a domestic producer

  Exception: If proper fields are not valid an error message is thrown

  Post-condition: Order is booked

- Set Appointments

  Summary: Used for setting up appointments

  Actor: Provider.

  Pre-condition: Provider should be logged in and valid

Description: For setting appointments for users in his/her locality

Exception: If proper fields are not valid an error message is thrown

Post-condition: Appointment is successfully registered.

- Process Orders

  Summary: Used for processing orders.

  Actor: Producer.

  Pre-condition: None.

  Description: For processing vaccines orders of a provider.

  Exception: If an internal problem happens error message is thrown.

  Post-condition: Order is successfully processed

3.5.5 Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

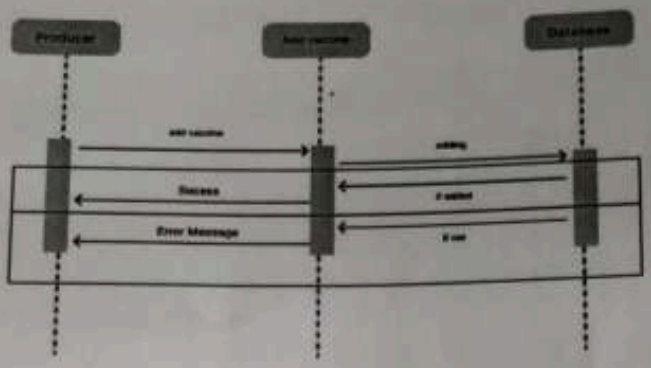
- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

References:

- Software Engineering 9th Edition Author : Ian Sommerville
- https://www.lucidchart.com/pages/uml-sequence-diagram

Name	Symbol	Description
Synchronous Message		An instantaneous communication between objects that conveys information, with the expectation that an action will be initiated as a result.
Activation Box		The period during which an object is performing an action.
Object		An object that is created, performs actions, and/or is destroyed during the lifeline

Table 3.5



Add vaccine

Figure 3.9

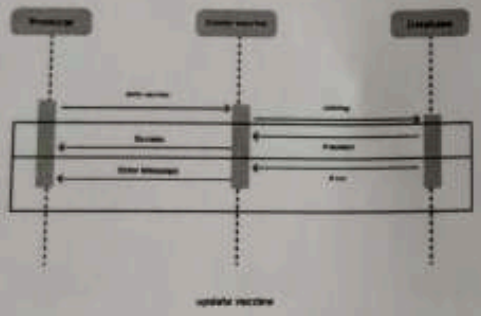
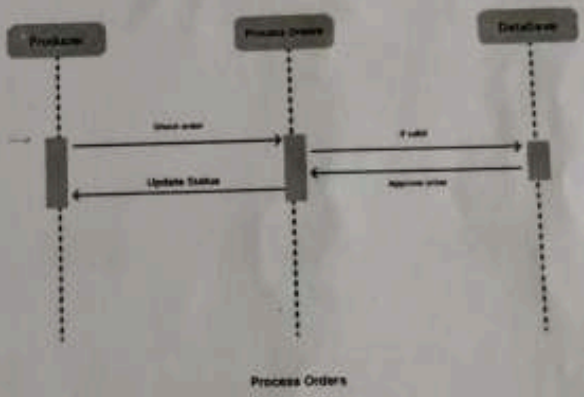


Figure 3.11



Process Orders

Figure 3.10

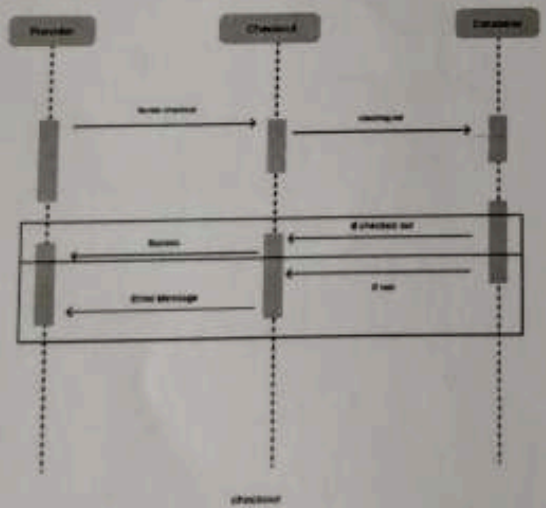


Figure 3.12

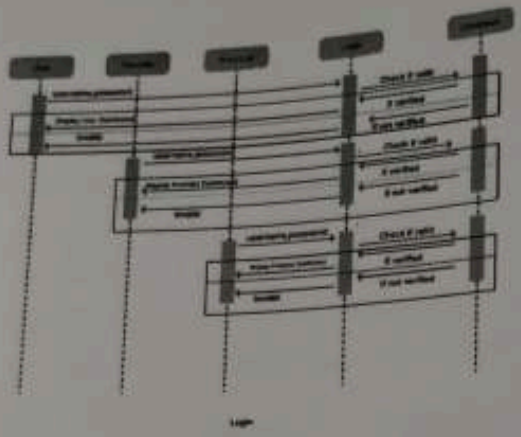


Figure 3.13

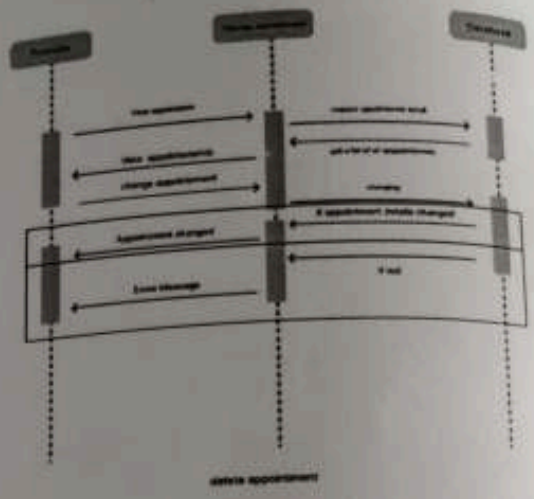


Figure 3.14

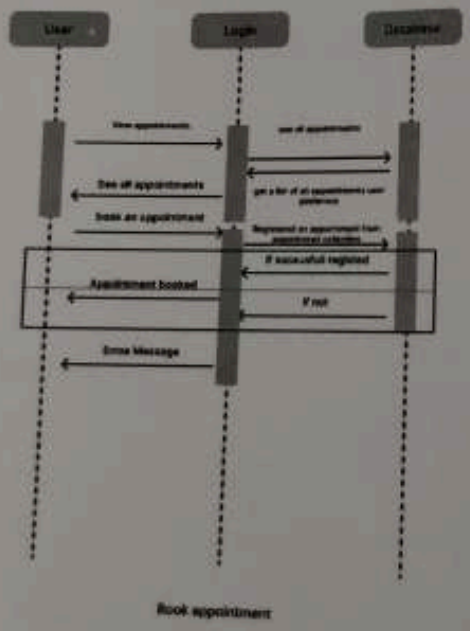


Figure 3.16

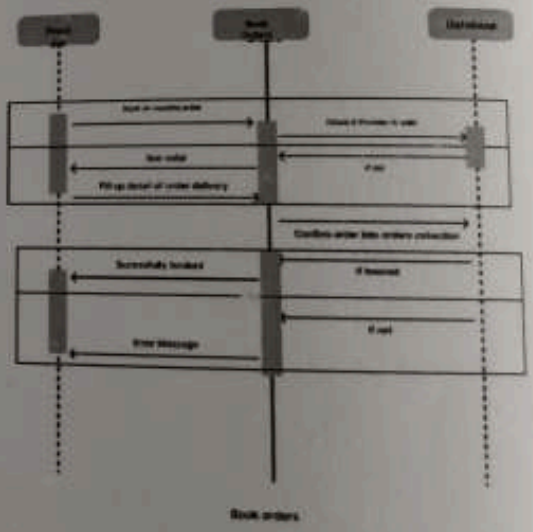


Figure 3.15

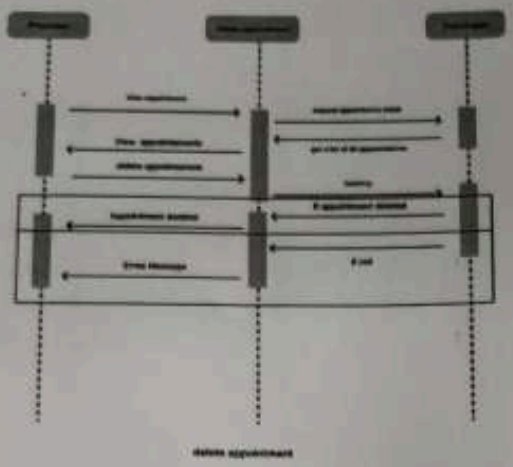


Figure 3.17

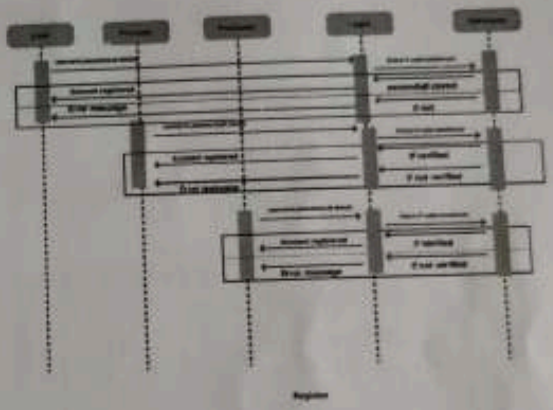


Figure 3.18

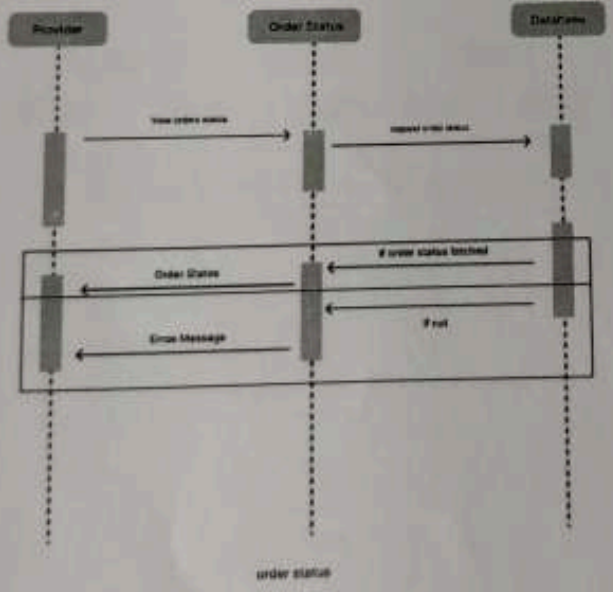


Figure 3.19

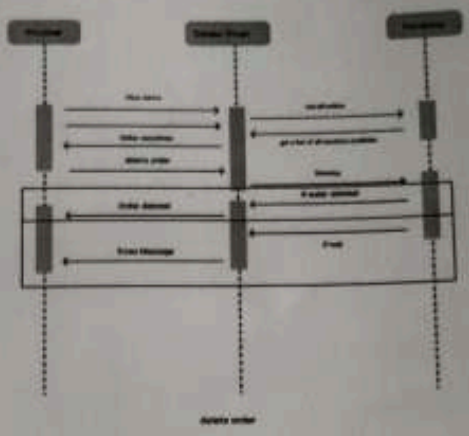


Figure 3.20

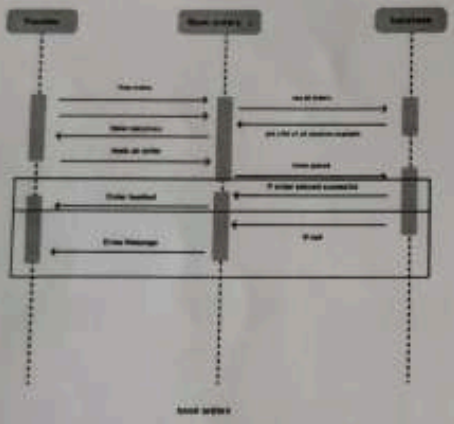


Figure 3.21

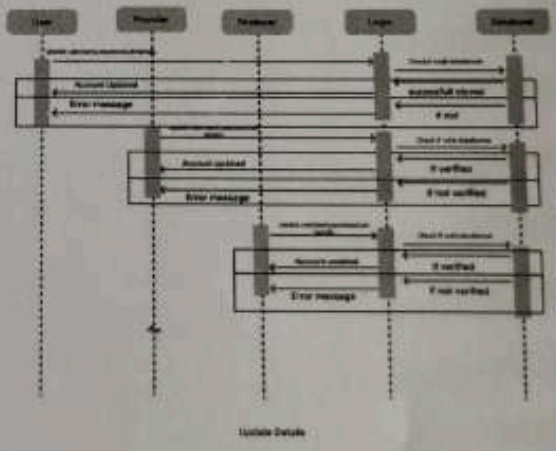
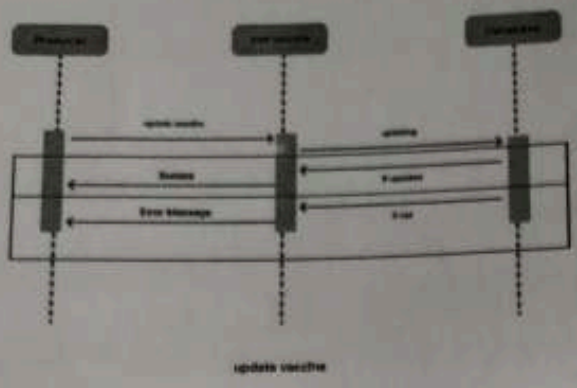
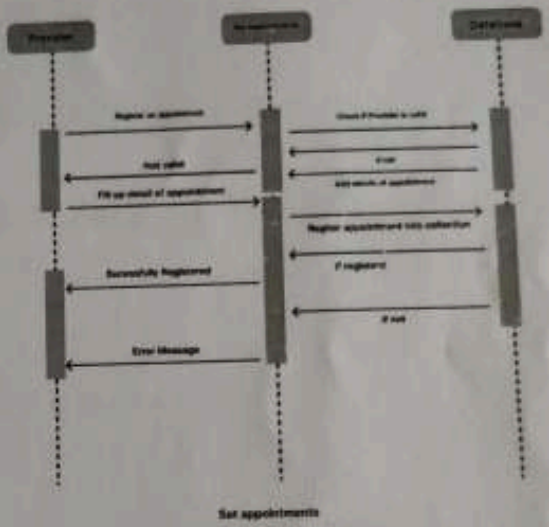


Figure 3.22



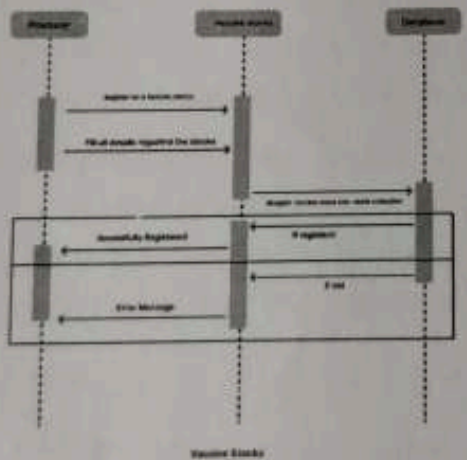
update vaccine

Figure 3.23



Set appointments

Figure 3.24



Vaccine Stocks

Figure 3.25

3.5.5 State Activity Diagram

Activity diagrams present several benefits to users. Consider creating an activity diagram to:

- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as method, function, and operation.

References:

- Software Engineering 9th Edition Author : Ian Sommerville
- https://www.lucidchart.com/pages/uml-state-diagram

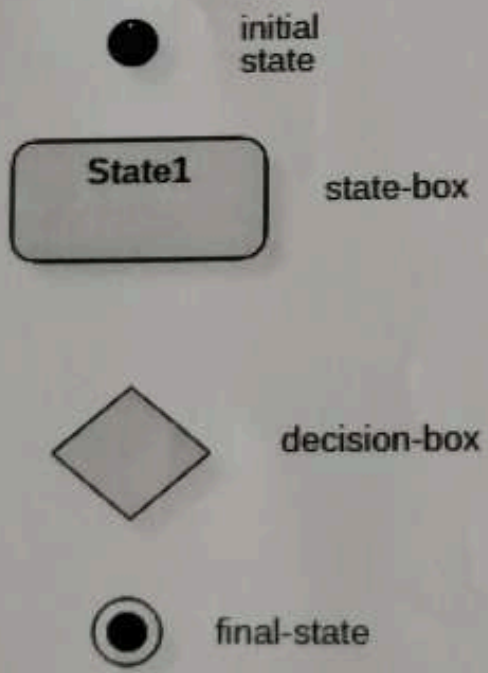


Table 3.6

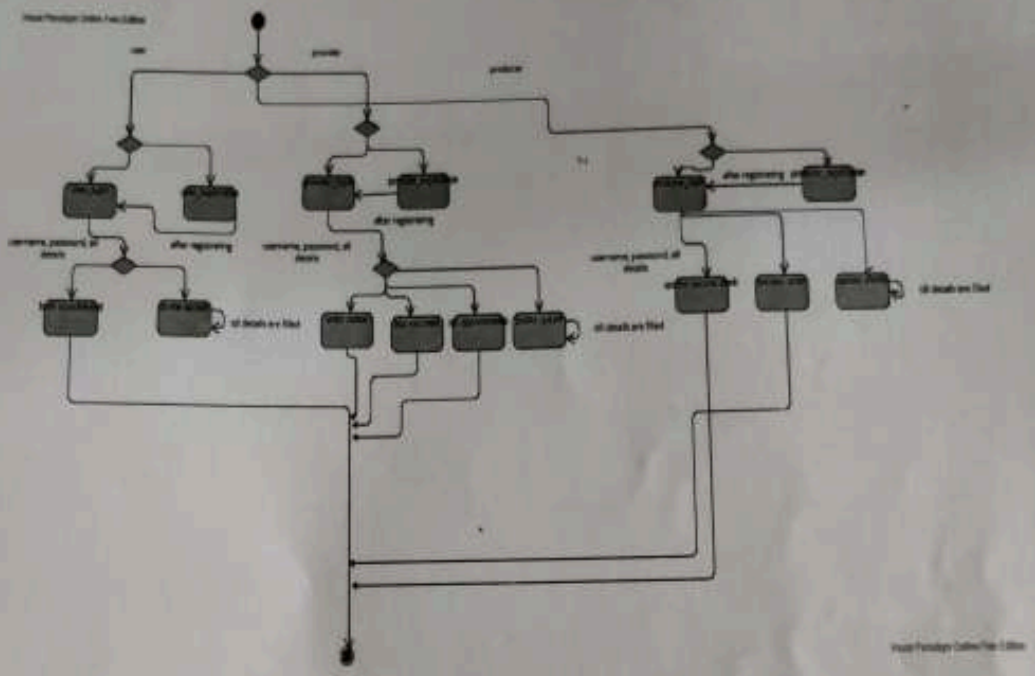


Figure 3.26 State Diagram

3.5.6 Activity Diagram

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane, etc.

Before drawing an activity diagram, we must have a clear understanding of the elements used in the activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

References:

- Software Engineering 9th Edition Author : Ian Sommerville

- https://www.lucidchart.com/pages/uml-activity-diagram

Sr. No	Name	Symbol
1.	Start Node	
2	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

Table 3.7

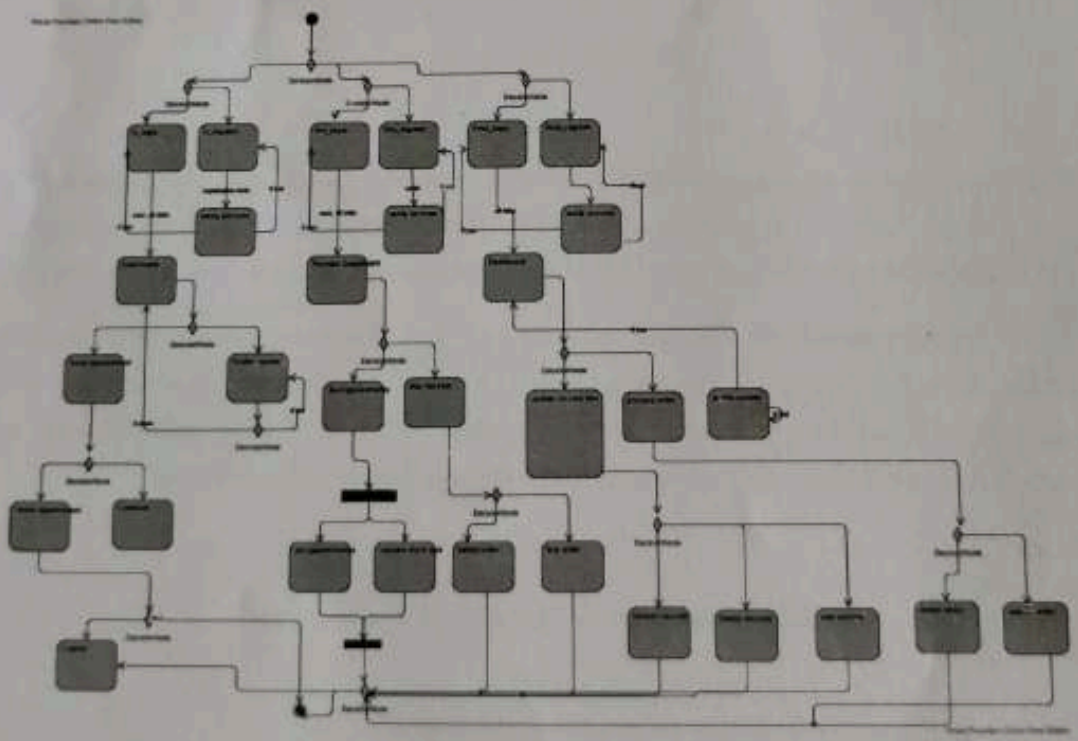


Figure 3.27 Activity Diagram

4.1 Interface Design

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

Interface elements include but are not limited to:

- Input Controls: buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field
- Navigational Components: breadcrumb, slider, search field, pagination, slider, tags, icons
- Informational Components: tooltips, icons, progress bar, notifications, message boxes, modal windows
- Containers: accordion

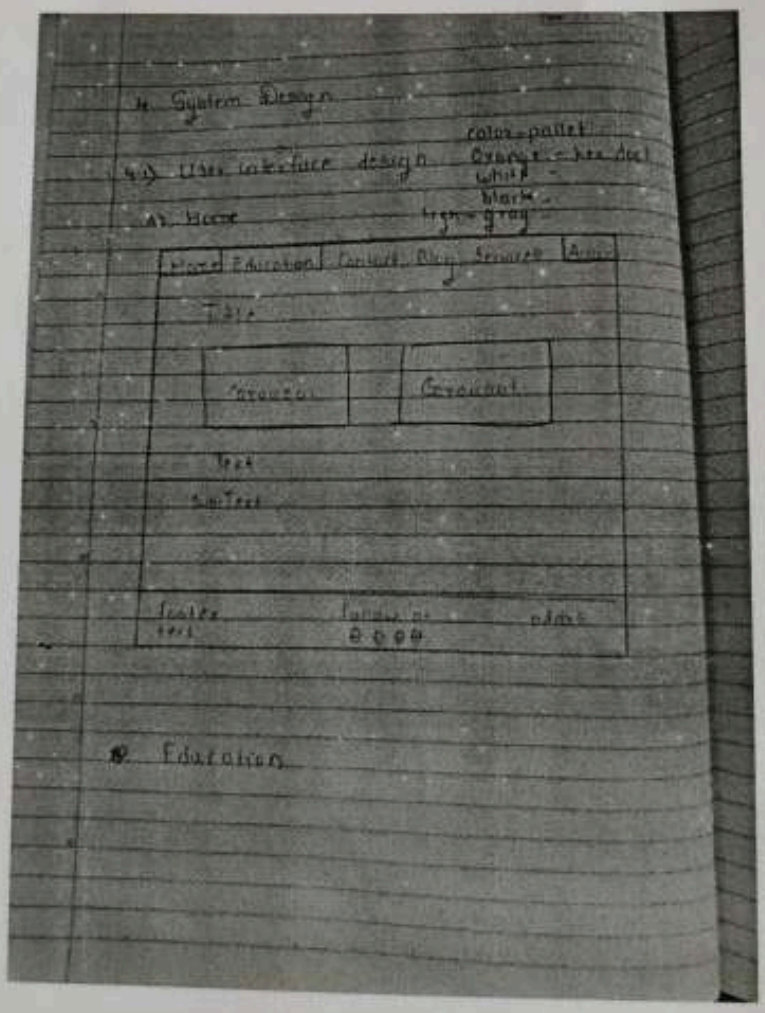


Figure 4.1

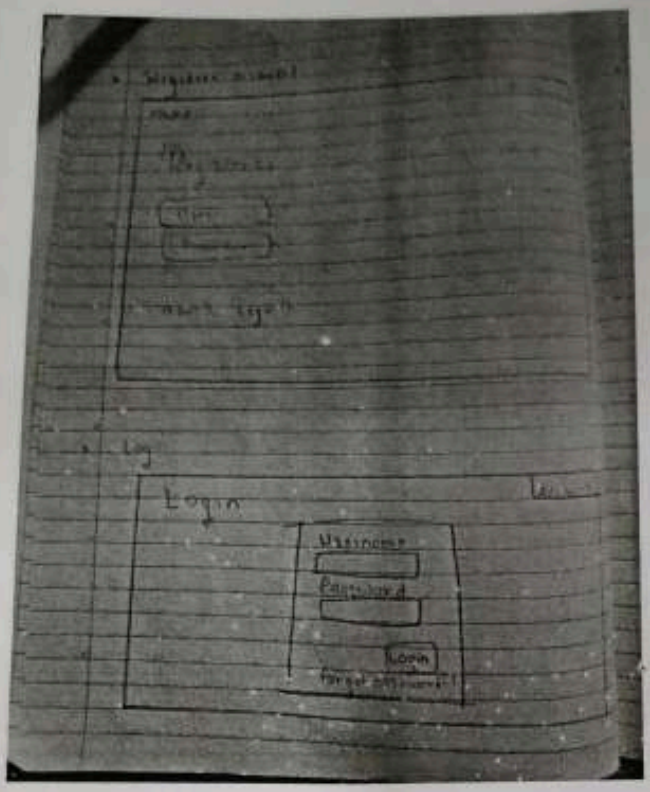


Figure 4.2

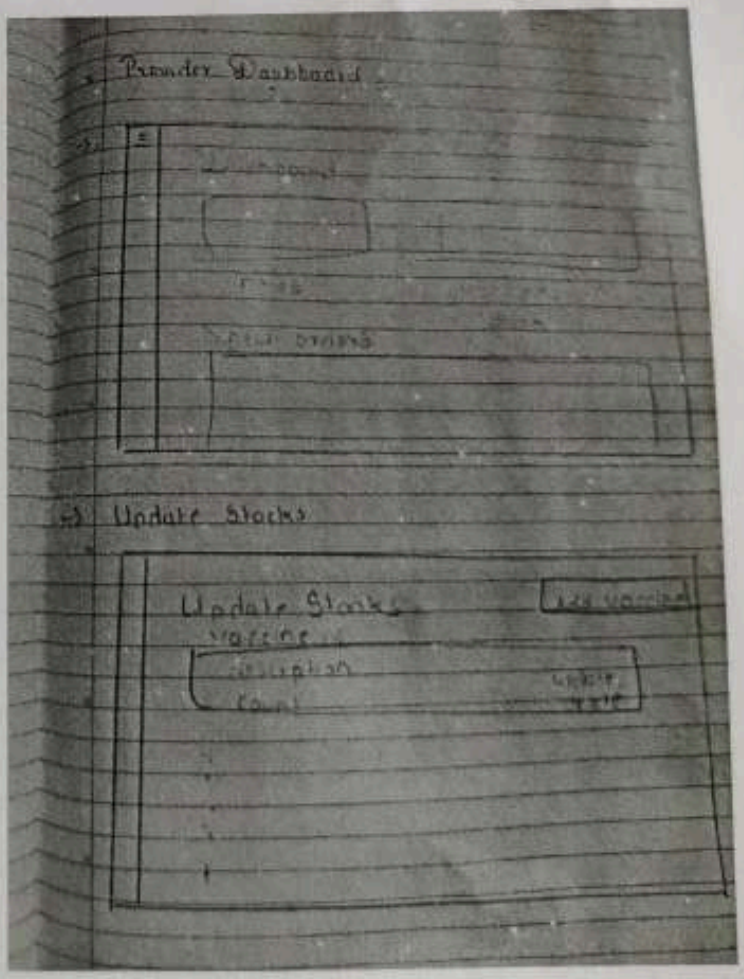


Figure 4.4

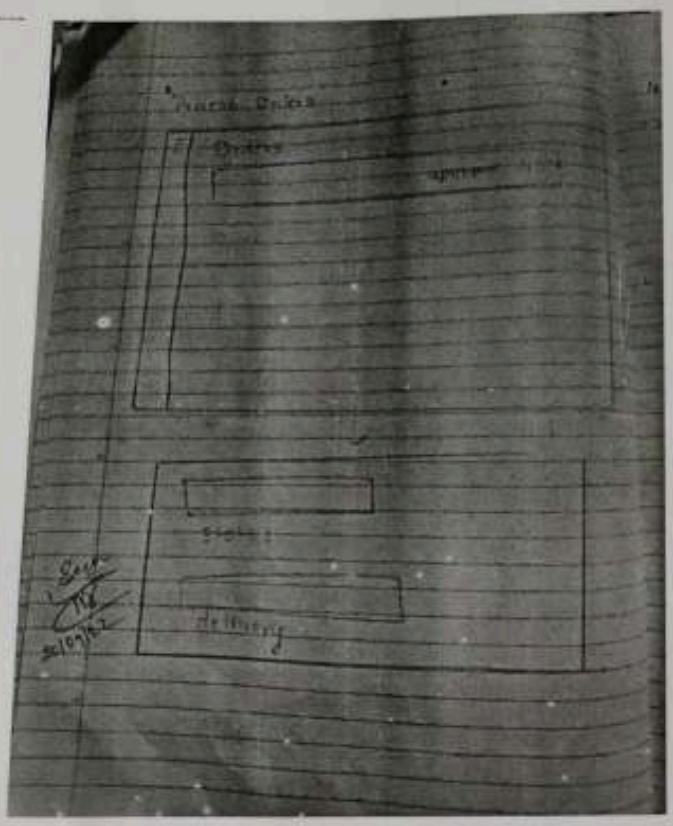


Figure 4.5

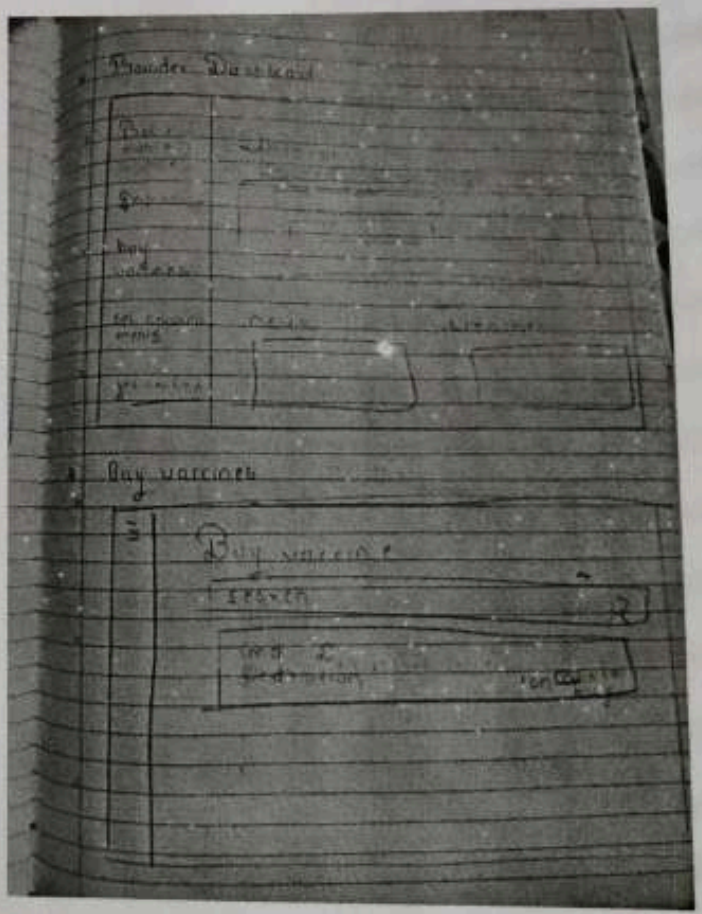


Figure 4.6

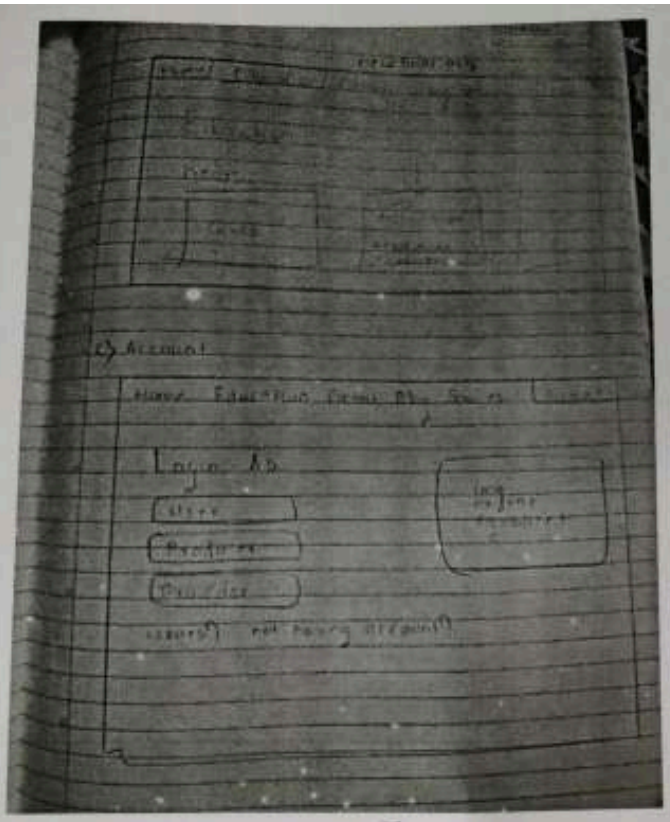


Figure 4.7

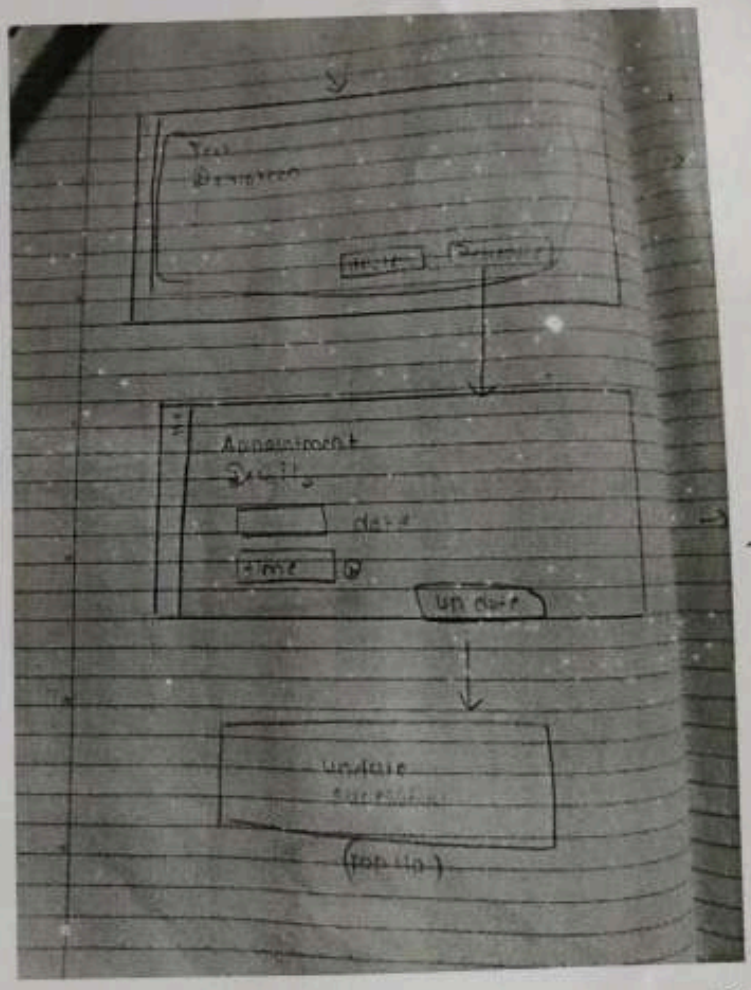


Figure 4.8

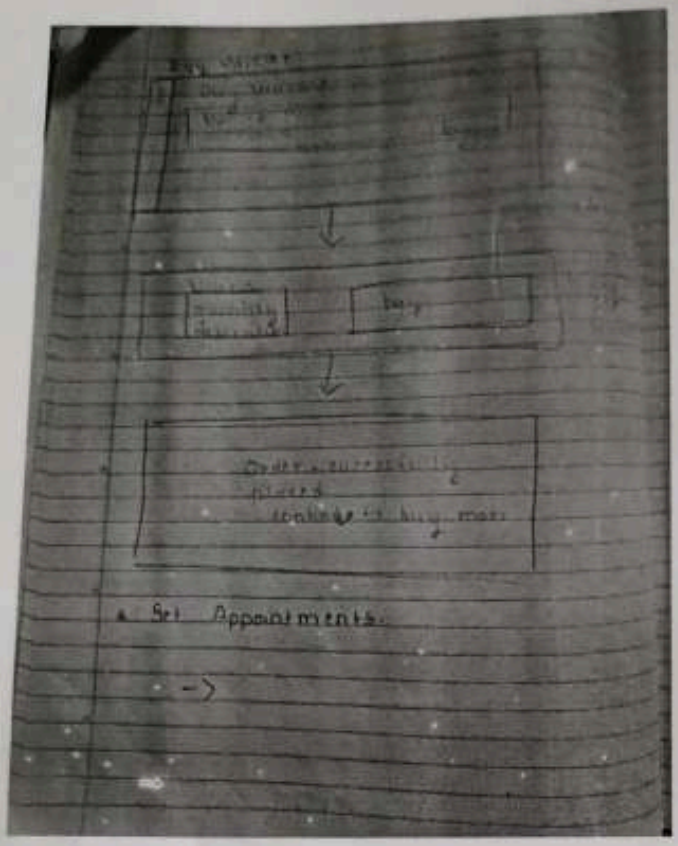


Figure 4.9

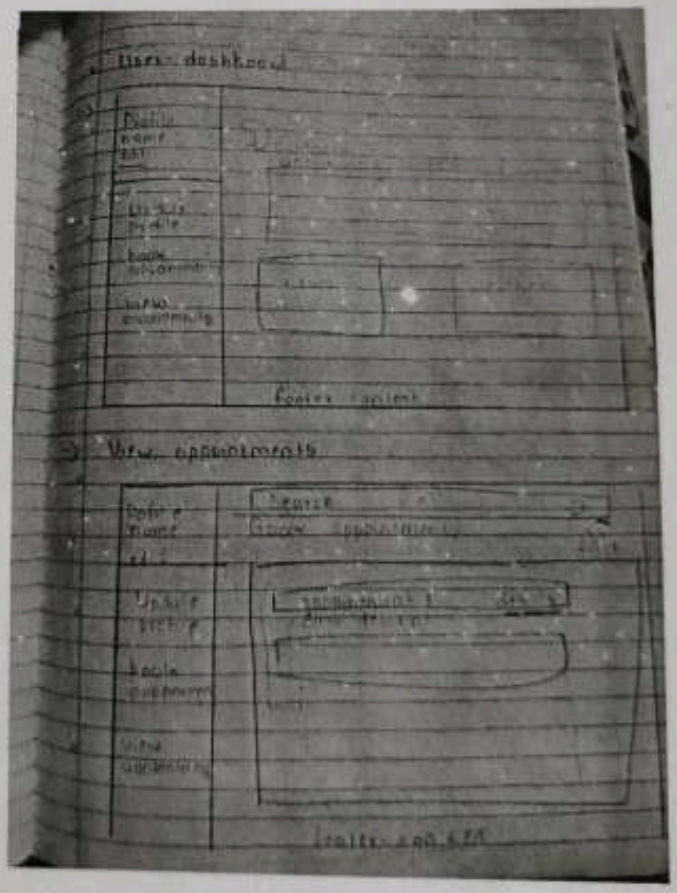


Figure 4.10

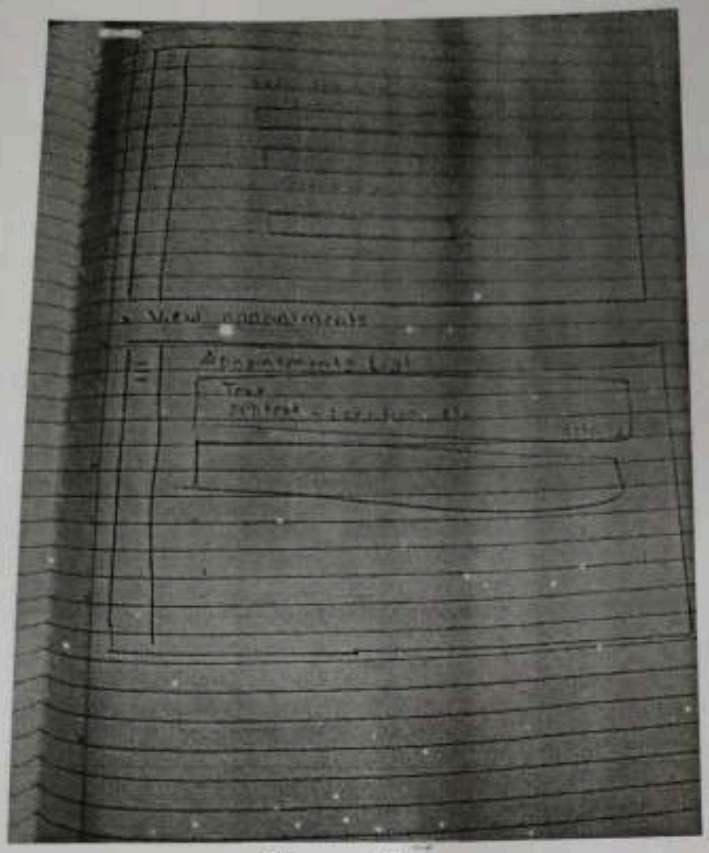


Figure 4.11

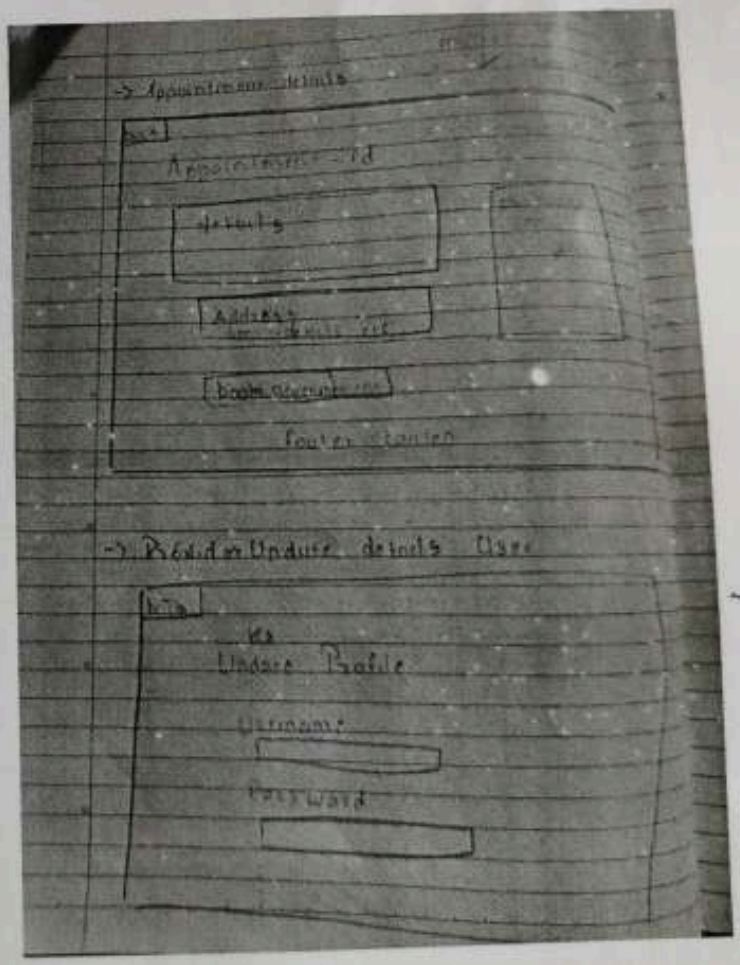


Figure 4.12

4.2 Test Cases

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case names, input conditions, and expected results. A test case is a first-level action derived from test scenarios.

ID	Test Case Description	Test Case Data	Expected Result	Actual Result	Remark
0	Login for user	Username: Otherwa Password: admin	Should get redirected to the dashboard		
2	Register for User	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 12121243445. pincode 42120,phone-no 8828388979,email atharvdesai2002@gmail.com	Error due to Adhar no		
3	Book Appointments	Appointment Name XXY Appointment Time 13:00 – 14:00 Date: 21/09/222 (On Book Button Click)	The appointment Should be booked		
4	Update Profile for user	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 2023023023. pincode 42120,phone-no 8828388979,email	User Profile Updated		

		atharvdesai2002@gmail.com			
5	Register for User	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 882838899,email atharvdesai2002@gmail.com	Error due to Phone number		
6	Set Appointments	Appointment Name : XYZ Appointment Location : Nalasopara/202,22e Appointment Timming : 13:00 – 14:00 Appointment Date : 24/09/22	Appointemnt should be setted up		
7	Buy Vaccine	On certain vaccine buy confirm order button click and checkout	Order should be placed		
8	Profile Update for provider	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 882838899,email atharvdesai2002@gmail.com NGO : Atmaram NGO	Profile Updated		
9	Profile Update for provider	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male,	Profile Not Updated NGO Field Blank		

		Adhar no 1212232pincode 421201,phone-no 882838899,email atharvdesai2002@gmail.com NGO :			
10	Update profile for provider	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email atharvdesai2002gmail	Error due to invalid email		
11	Set Appointments	Appointment Name : XYZ Appointment Location : Nalasopara /202,22e Appointment Timming : 13:00 – 14:00 Appointment Date :	Error due to no date specified		
12	Set Appointments	Appointment Name : XYZ Appointment Location : Nalasopara /202,22e Appointment Timming : 13:00 – Appointment Date : 20.09.22	Error due to no end time specified		
13	Process Orders	On button Click of delete order status order should be deleted	Order Should get deleted		
14	Process Orders	On button Click of Update order status order should be updated	Order Should get Updated		
15	Update Vaccine Stonks	Vaccine Name : BCG Vaccine Code : 2033 Description: wfwddssdsd Quantity:203	Vaccine stocks not Updated due		

			to faulty description		
16	Profile Update For Producer	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email atharvdesai2002@gmail.com Company Name : Le-Vaccine Company Address : C/202 Shree Samartha Krupa	Should not get updated Successfully Not valid Company Name		
17	Profile Update For Producer	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email atharvdesai2002@gmail.com Company Name : Bhartia Biotech Company Address : C/202 Shree Samartha Krupa	Should get updated Successfully		
18	Process Orders	On the button Click of Update order status order should be updated	Order Should get Updated		
19	View Order Status	On button click of view status	Order status should be shown		

20	Update Vaccine Stonks	Vaccine Name: BCG Vaccine Code: 2033 Description: For Immunity etc, Quantity:203	Vaccine stocks Updated		
21	Update Vaccine Stonks	Vaccine Name: BCG Vaccine Code: 2033 Description: For Immunity etc, Quantity:0	Vaccine stocks Updated		
22	Update Vaccine Stonks	Vaccine Name: BCG Vaccine Code: 2033 Description: For Immunity etc, Quantity:	Vaccine stocks are not Updated		
23	Profile Update For Producer	Username: Otherwa1 Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email atharvdesai2002@gmail.com Company Name: Bhartia Biotech Company Address: C/202 Shree Samartha Krupa	The username is already taken error		
24	Profile Update For Producer	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email	Should not get updated Successfully		

		atharvdesai2002@gmail.com Company Name: Bhartia Biotech Company Address :			
25	Profile Update For Producer	Username: Otherwa Password: admin Name: Atharv Ankush Desai, Age 20,gender male, Adhar no 1212232pincode 421201,phone-no 8828388979,email atharvdesai2002@gmail.com Company Name : Company Address : C/202 Shree Samartha Krupa	Should not get updated Successfully		

Table 4.1

5. Bibliography

Websites

- https://cdsco.gov.in/opencms/export/sites/CDSCO_WEB/Pdf-documents/biologicals/facilitiesLIST.pdf

  as of 19/05/22

- https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-vaccines?adgroupsurvey={adgroupsurvey}

  as of 22/09/22

- https://www.servicenow.com/solutions/vaccine-management.htm
  as of 01/09/22
- https://www2.deloitte.com/us/en/pages/public-sector/solutions/vaccine-management-system.htm
  as of 02/06/22
- https://www.intelex.com/products/applications/vaccine-management} {Intelex Vaccine Solution
  as of 05/05/22
- https://www.zenflowchart.com/guides/data-flow-diagram
  as of 01/09/22
- https://www.newthinktank.com/2019/01/latex-tutorial/
  as of 28/06/22
- https://online.visual-paradigm.com/drive/#diagram:proj=0&type=GanttChart&workspace=mhujsaxv&id=2
  as of 12./07/22
- https://www.lucidchart.com/pages
  as of 16/08/22
- https://www.mongodb.com/docs/manual/core/data-modeling-introduction/
  as of 15/08/22
- https://www.lucidchart.com/pages/uml-use-case-diagram
  as of 21/08/22
- https://www.lucidchart.com/pages/uml-sequence-diagram

as of 22/08/22

- https://www.lucidchart.com/pages/data-flow-diagram
  as of 23/08/22
- https://www.lucidchart.com/pages/uml-state-machine-diagram
  as of 21/08/22
- https://www.lucidchart.com/pages/uml-activity-diagram
  as of 20/08/22
- https://youtu.be/WnMQ8HlmeXc
  as of 07/09/22

Reference books

- Practical MongoDB: Architecting, Developing, and Administering MongoDB
  Author : Shakuntala Gupta Edward & Navin Sabharwal as of 16/07/22

- Software Engineering 9th Edition
  Author : Ian Sommerville as of 19/06/22

- Learning Node.js Development: Learn the fundamentals of Node.js, and deploy and test Node.js applications on the web
  Author : Andrew Mead as of 20/08/22

- Beginning Node.js, Express & MongoDB Development
  Author : Greg Lim as of 22/08/22

- Fundamentals of Creating a Great UI/UX
  Author: - Creative Tim as of 23/08/22