Pimpri Chinchwad Education Trust’s
Pimpri Chinchwad College of Engineering
Department of Computer Engineering

LPII

Mini Project Report

On

Healthcare Management System

Group Members

BECOA123:	Saurabh Damle
BECOA133:	Shubhankar Gengaje
BECOA137:	Bhavesh Gulabani
BECOA146:	Chinmay Joshi

Guide

Prof. Anagha Chaudhari

ACKNOWLEDGEMENT

We feel immense pleasure in presenting this project report on "**Healthcare Management System**". We wish to express true sense of gratitude towards **Prof. Anagha Chaudhari**, our project guide who at a very discrete step in study of this project, contributed her valuable guidance and helped to solve every problem that arose.

We take this opportunity to thank our Principal **Dr. G.V. Parishwad** and **Prof. Dr. K. Rajeswari** (HOD Comp Dept.) for opening the doors of the department towards the realization of our project, for their guidance and encouragement.

Most importantly, we would also like to express our sincere gratitude towards all the staff members of the Computer Department. We also express our thanks to all our friends for their support and suggestions shown during the completion of our project. We take this opportunity to express our thanks to all who rendered their valuable help, along with all those unseen people across the internet for maintaining those valuable resources for the successful completion of our project. We owe our success to all of them.

Index

List of Tables

List of Figures

Introduction

This project aims to develop a Healthcare Management System which will manage the medical, financial and administrative aspects related to a hospital. The system is expected to automate the existing manual processes of patients, doctors and appointment management

Motivation

The idea came by when we wanted to develop a lightweight system which can be used by every kind of hospital as well as small clinics.We came up with this idea when we visited a local government clinic where the documentation was done manually which was a tedious process for the person handling the things.

Problem Definition

Problem Definition

To overcome existing problems occurring in maintenance of patient information in hospitals.

Objectives

- The main objective of this project is to automate the maintenance of patient and staff details in the hospital.
- The information of patients, their appointments, and doctors will be stored in a structured manner for easy and efficient retrieval.
- It will also provide an interactive GUI which will be easy to navigate.

Scope

- This system will be able to manage the patient records and the processes like admission, medical record creation, booking appointment as well as billing procedure.

- This project can be used by any hospital and doesn’t require any state-of-the-art facilities to run the software.

Problem Description

The current system for managing patient records is paper based. This makes tracking and updating changes very difficult. Majority of the time the receptionist spends maintaining these records. This reduces their efficiency. To combat this issue, we came up with a solution, a database management system that will facilitate easy storage and retrieval of data.

Software Requirements and Specifications

Objectives

To select a robust backend database i.e. a RDBMS that will be able to handle large volumes of data. To select a lightweight framework for the front end that will run on a simple hardware.

Scope

The system implemented will be a Web Application which will be accessible primarily through a Web Browser. The database management system used will be MySQL, and Python will be used as a server-side scripting language and for connectivity between the database and the front-end.

Overall Description

Product perspective

The system will provide functionality to store and retrieve patient data, medical records, booking and canceling of appointments and billing. It will do so by maintaining a database which will be used to store the data. The system will be reliable and provide a high up time.

Overview of functional requirements

The functional requirements will cover all the aspects of the system that affect the usability of the system. It will cover things like the functionality needed and the data required.

Overview of non-functional requirements

The non-functional requirements will cover all the aspects of the system that affect the performance of the system. It will cover things like the response time, up time, accessibility and platform used.

Functional requirements

- Management of doctors and staff details
- Consultation management:
    - Recording patient details
    - Appointment booking
    - Billing and receipt generation
- Patient treatment management:
    - Allocation of room
    - Medical report generation and analysis
- Hospital medical inventory management

Non-functional requirements

- Response time: The system will give responses within 1 second after checking the patient information and other information.
- User interface: User interface screen will respond within 1.5 seconds

- Availability: The system shall be available 99.999% of the time.
- Accessibility: Administrator and other users can access the system but the privileges are assigned differently based on the role of each user.

Experimentation details

Platform used

- Windows 10
- Chrome Browser
- XAMPP software

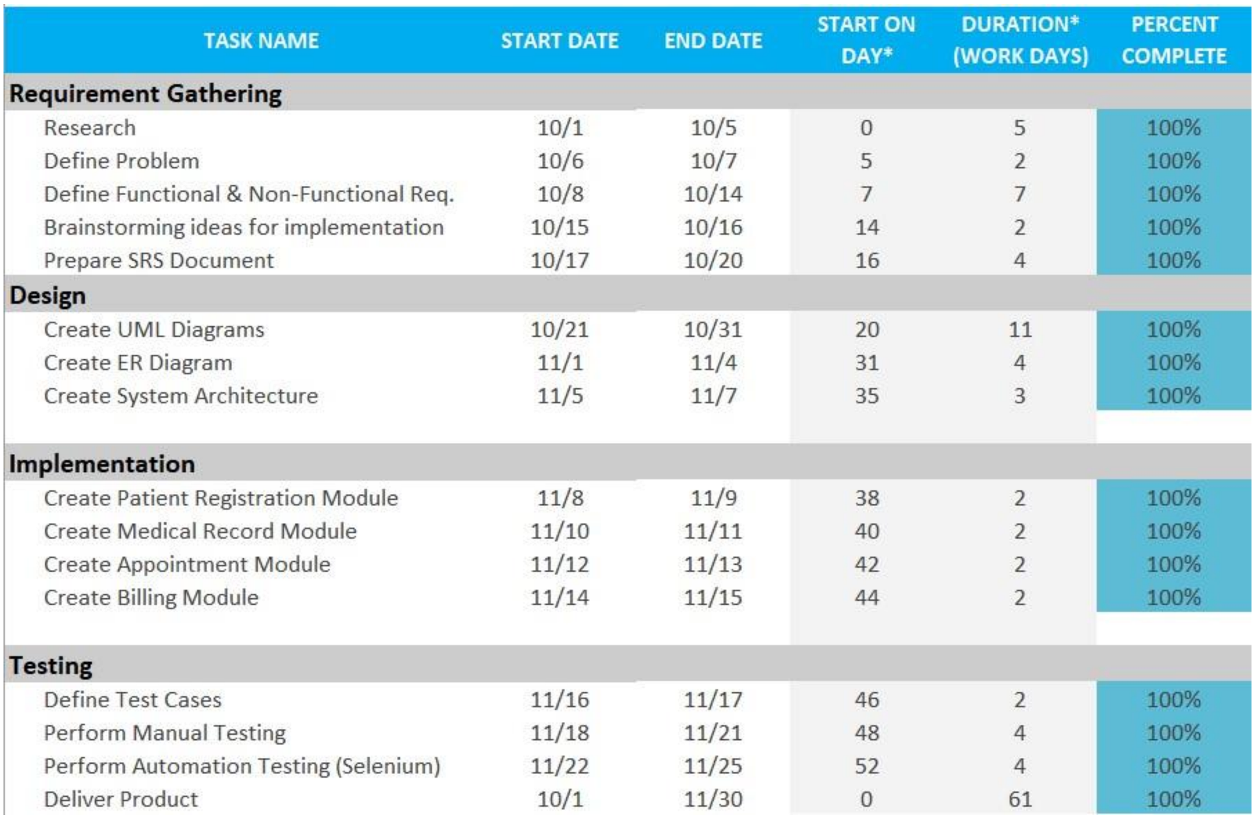
Languages used

- HTML
- CSS
- JS
- PYTHON
- JAVA
- Spring Boot

Project Plan

Project Plan

Gantt Chart



TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)	PERCENT COMPLETE
Requirement Gathering					
Research	10/1	10/5	0	5	100%
Define Problem	10/6	10/7	5	2	100%
Define Functional & Non-Functional Req.	10/8	10/14	7	7	100%
Brainstorming ideas for implementation	10/15	10/16	14	2	100%
Prepare SRS Document	10/17	10/20	16	4	100%
Design					
Create UML Diagrams	10/21	10/31	20	11	100%
Create ER Diagram	11/1	11/4	31	4	100%
Create System Architecture	11/5	11/7	35	3	100%
Implementation					
Create Patient Registration Module	11/8	11/9	38	2	100%
Create Medical Record Module	11/10	11/11	40	2	100%
Create Appointment Module	11/12	11/13	42	2	100%
Create Billing Module	11/14	11/15	44	2	100%
Testing					
Define Test Cases	11/16	11/17	46	2	100%
Perform Manual Testing	11/18	11/21	48	4	100%
Perform Automation Testing (Selenium)	11/22	11/25	52	4	100%
Deliver Product	10/1	11/30	0	61	100%

Table 4.1

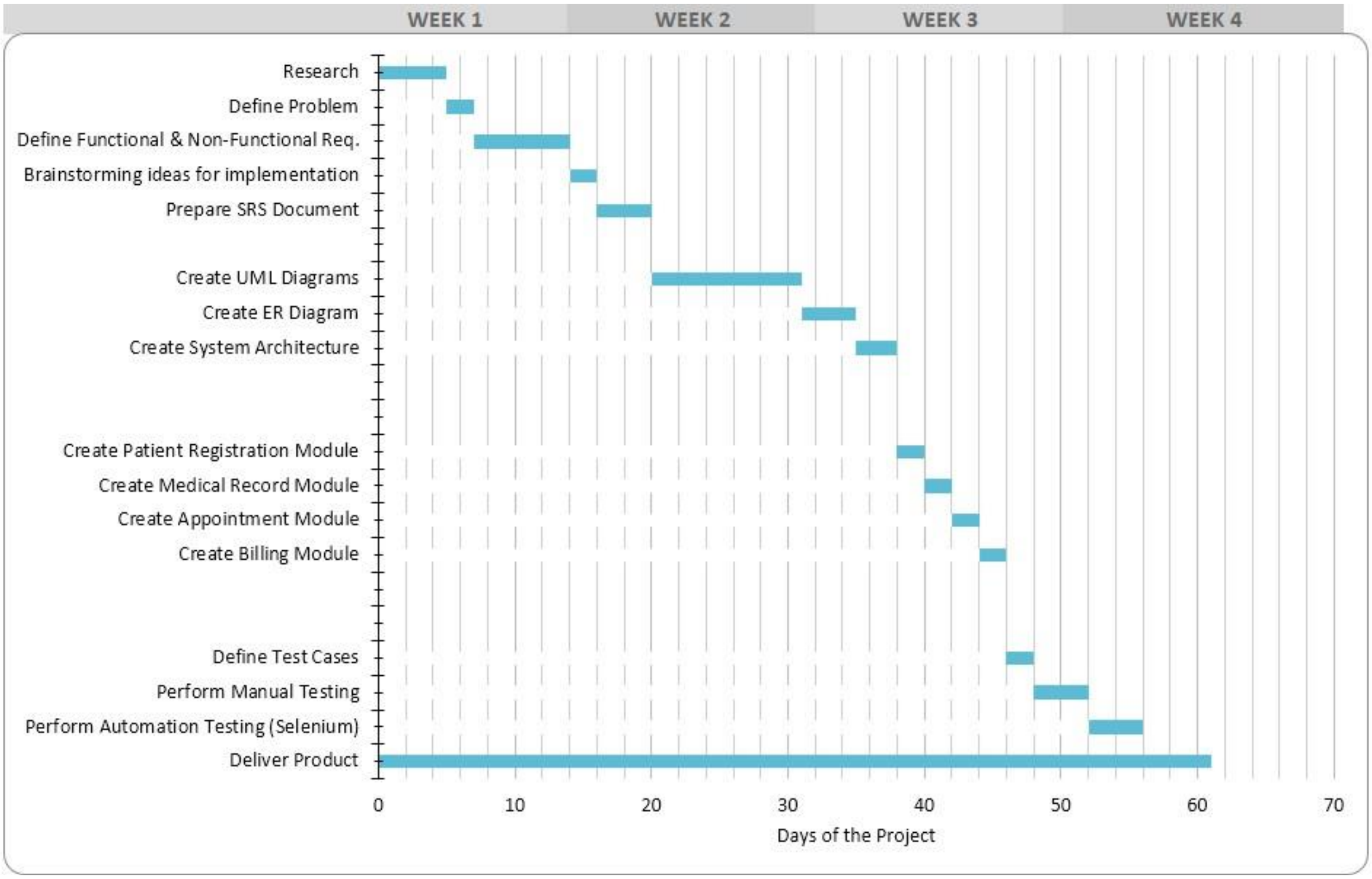


Figure 4.1

Feasibility Study

Technical Feasibility

The system will require basic computational hardware to run. Required hardware containing a mouse, display monitor, keyboard, speaker, CPU should be available at the facility. The system can run on any Operating system namely (MacOS, Windows or Linux). An IT professional needs to be appointed for each facility who will manage all the systems.

Economic Feasibility

Because the system will be reliable, robust and accessible, many hospitals can use it to run their facility. This will reduce the paperwork by more than 80%. This will reduce the consumption of paper which in turn will bring down the cost of maintaining and archiving records. This money which is saved can be used for other projects like improving the facilities provided at the healthcare center.

Performance Feasibility

The system will be robust, reliable, and accessible. The up time for the system will be 99.99%. The data will be backed up locally and on a cloud which will ensure safety of the records in case the local database gets corrupted. The retrieval time for any data will be less than 1sec.

Time Feasibility

The project will take at least 1 month to implement. Getting the system set up in a hospital will take around a week and training the staff will take around 2 weeks.

Risk Analysis

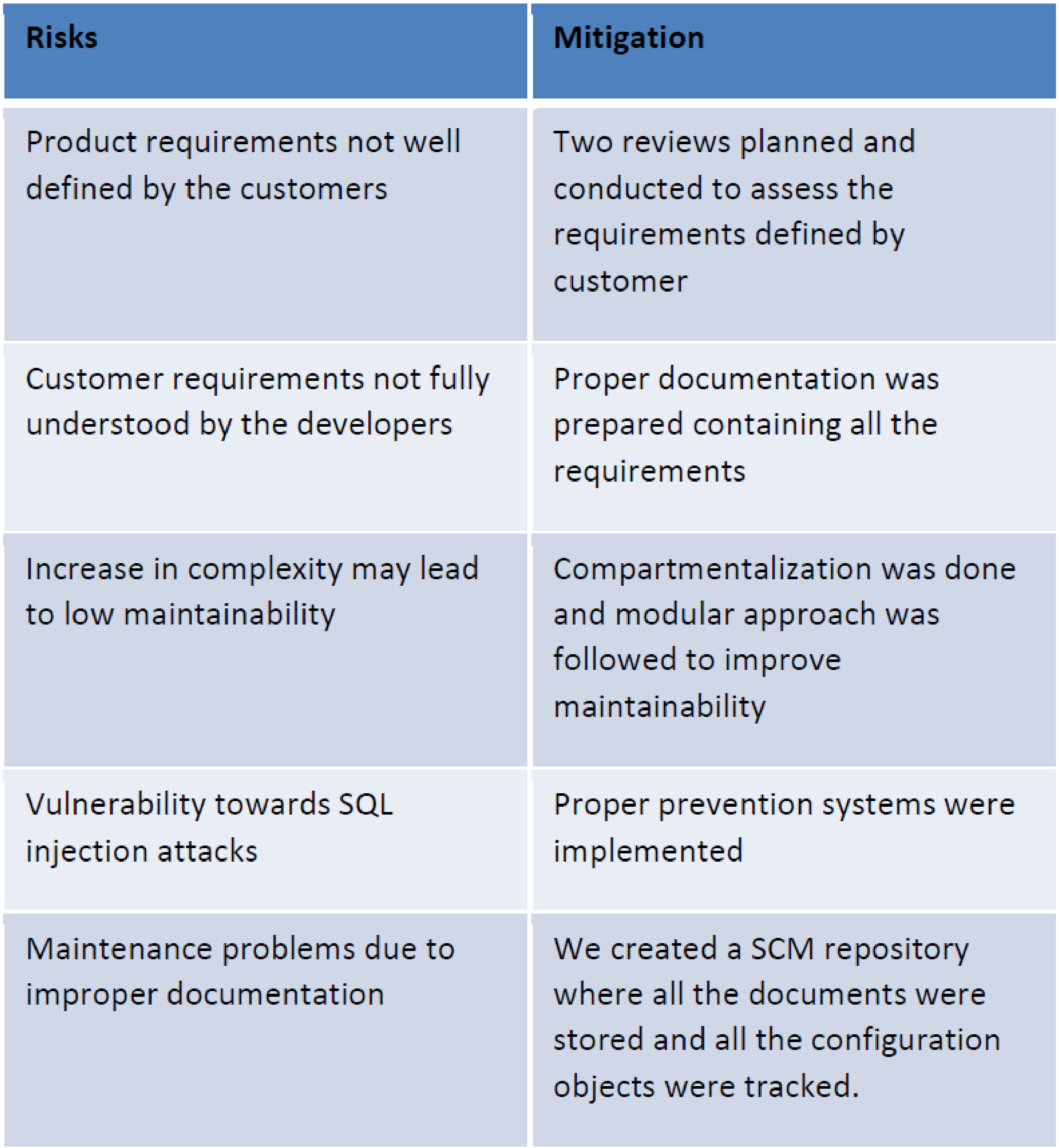
Project Risk

- Product requirements not well defined by the customers
- Customer requirements not fully understood by the developers.
- Increase in complexity may lead to low maintainability.

Technical Risks

- Vulnerability towards SQL injection attacks.
- Maintenance problems due to improper documentation.
- Server downtime due to lack of availability

Risk Mitigation



Risks	Mitigation
Product requirements not well defined by the customers	Two reviews planned and conducted to assess the requirements defined by customer
Customer requirements not fully understood by the developers	Proper documentation was prepared containing all the requirements
Increase in complexity may lead to low maintainability	Compartmentalization was done and modular approach was followed to improve maintainability
Vulnerability towards SQL injection attacks	Proper prevention systems were implemented
Maintenance problems due to improper documentation	We created a SCM repository where all the documents were stored and all the configuration objects were tracked.

Table 4.2

Effort and Cost Estimation

Lines of Code (LOC)

Function	**LOC**
User Interface	437
Patient Registration	211
Medical Record	435
Billing	245
Appointment	360
Total Lines of Code	1688

Table 4.3

Development Time

It took 2 months to develop this project

Number of People

4 people

SYSTEM DESIGN AND IMPLEMENTATION

System Architecture

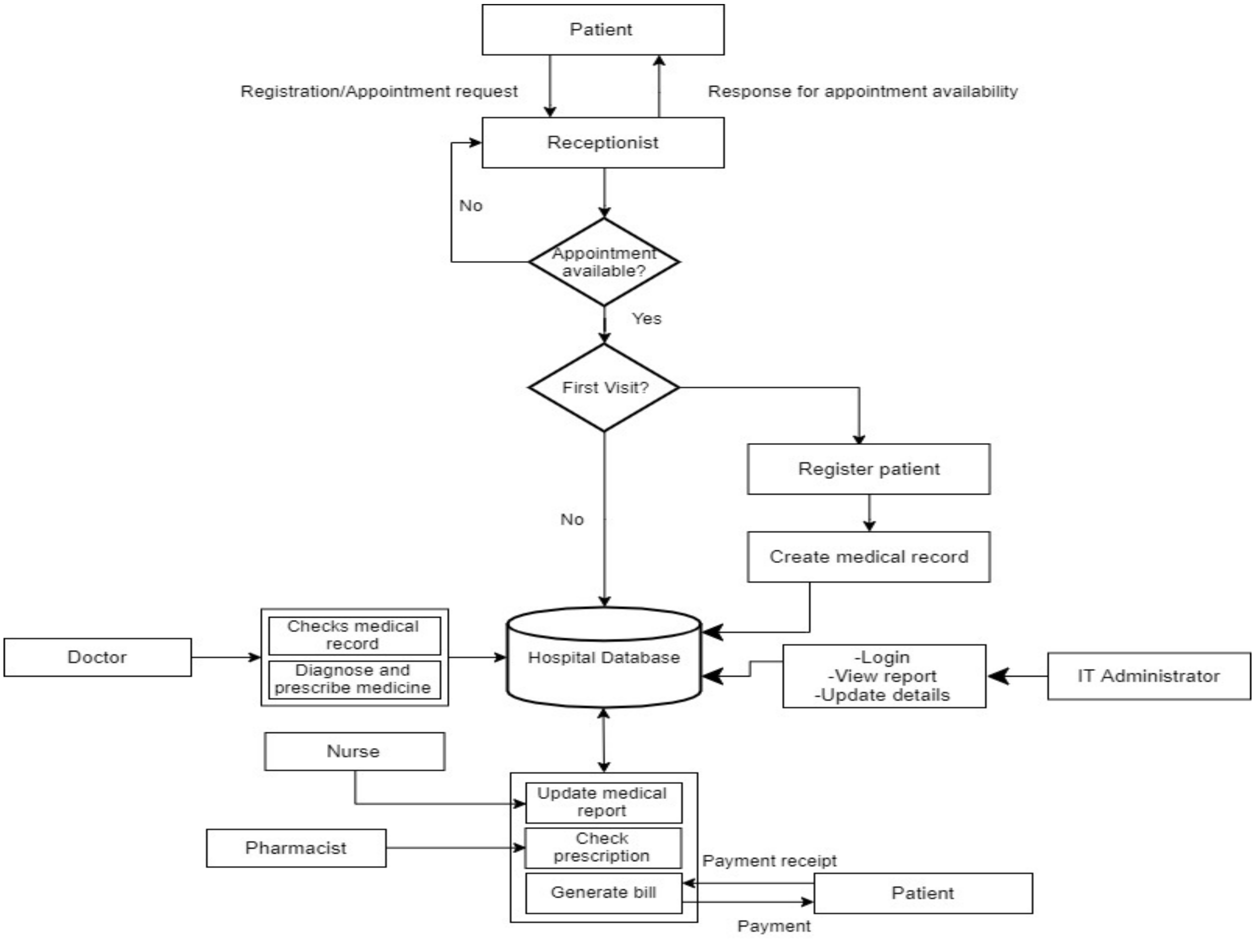


Figure 5.1

Database Design

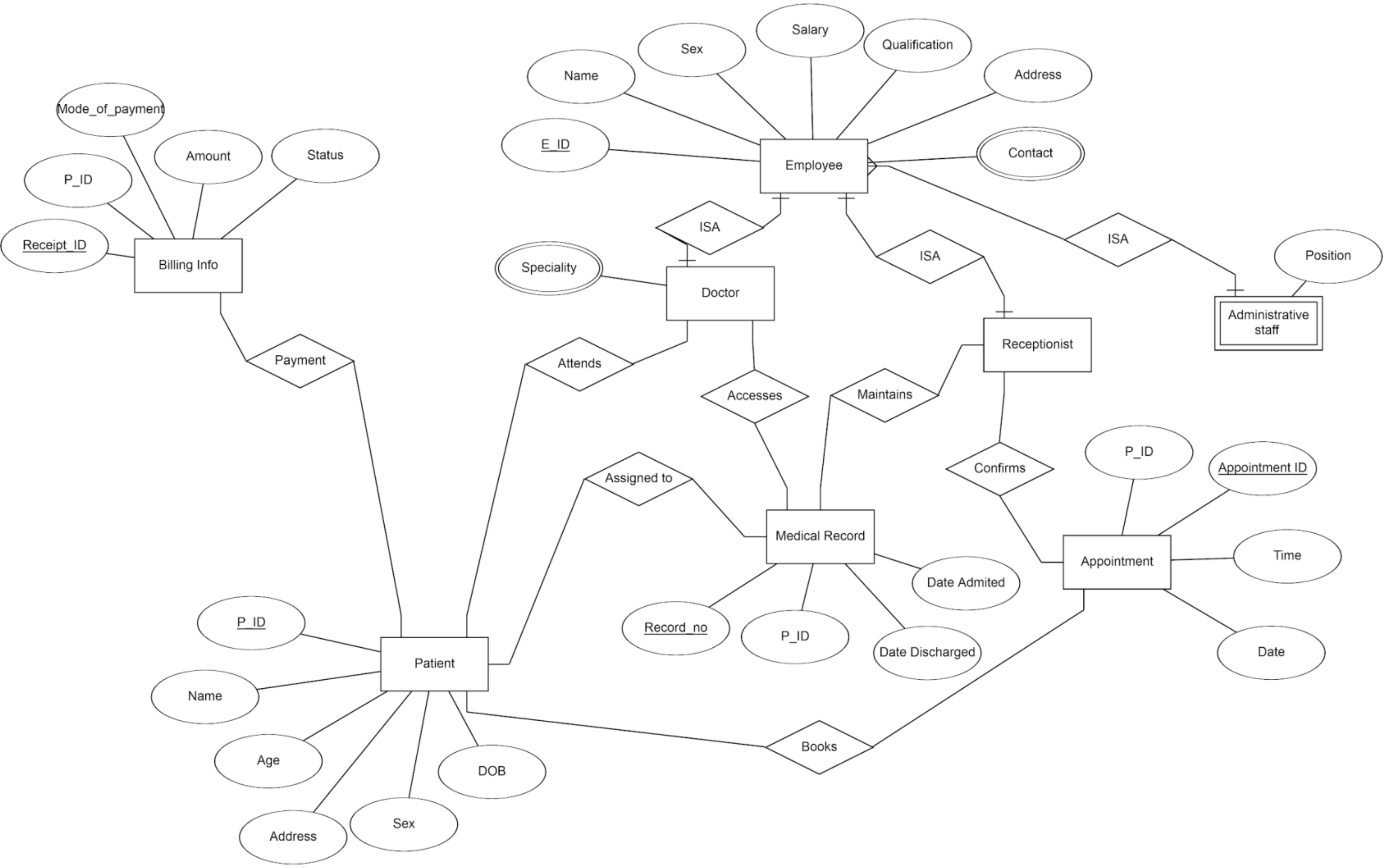


Figure 5.2

UML Diagrams

Use Case Diagrams

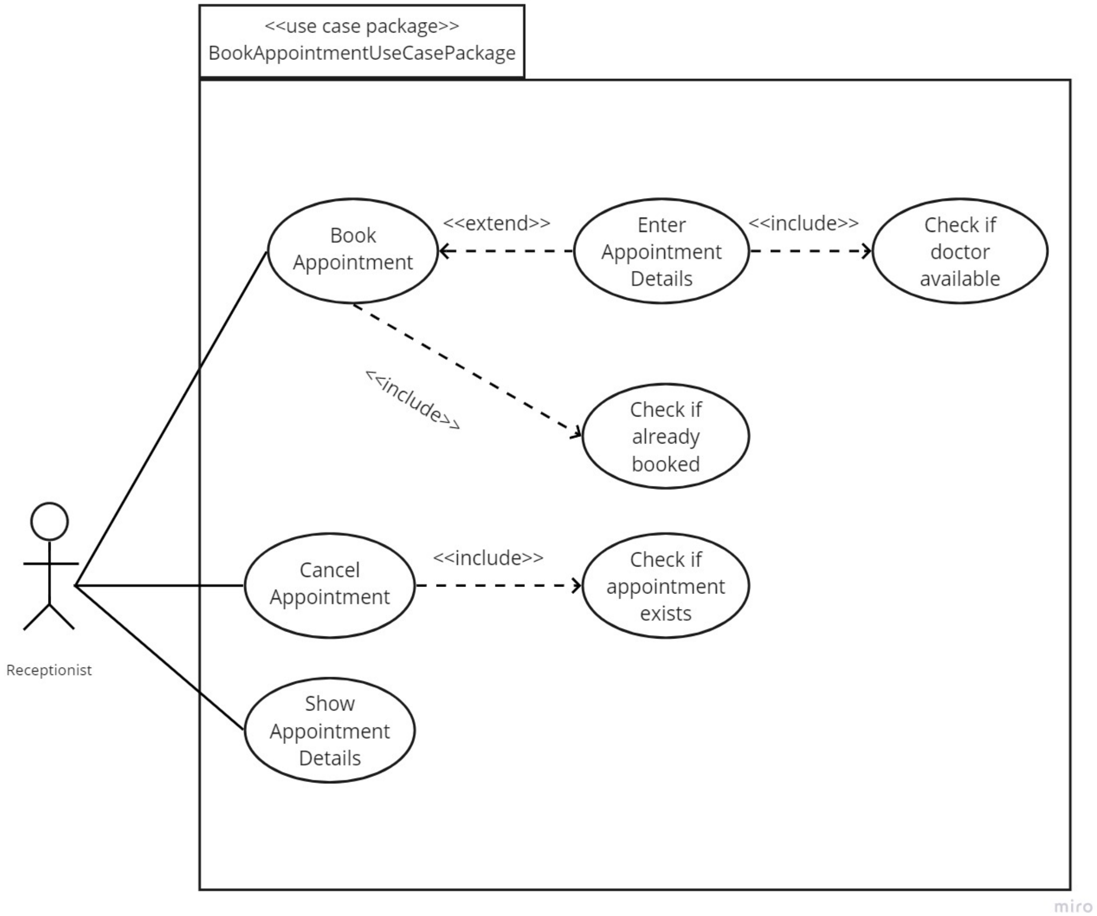


Figure 5.3

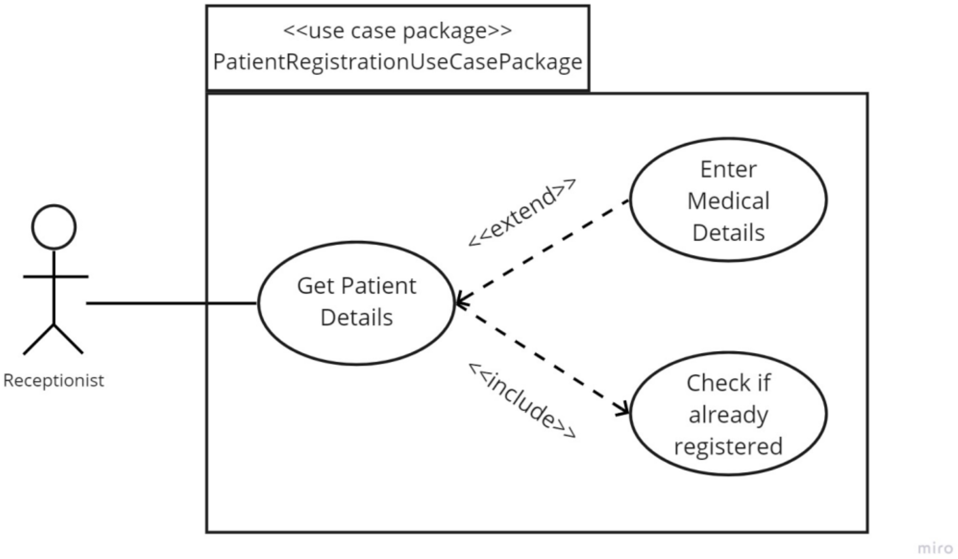


Figure 5.4

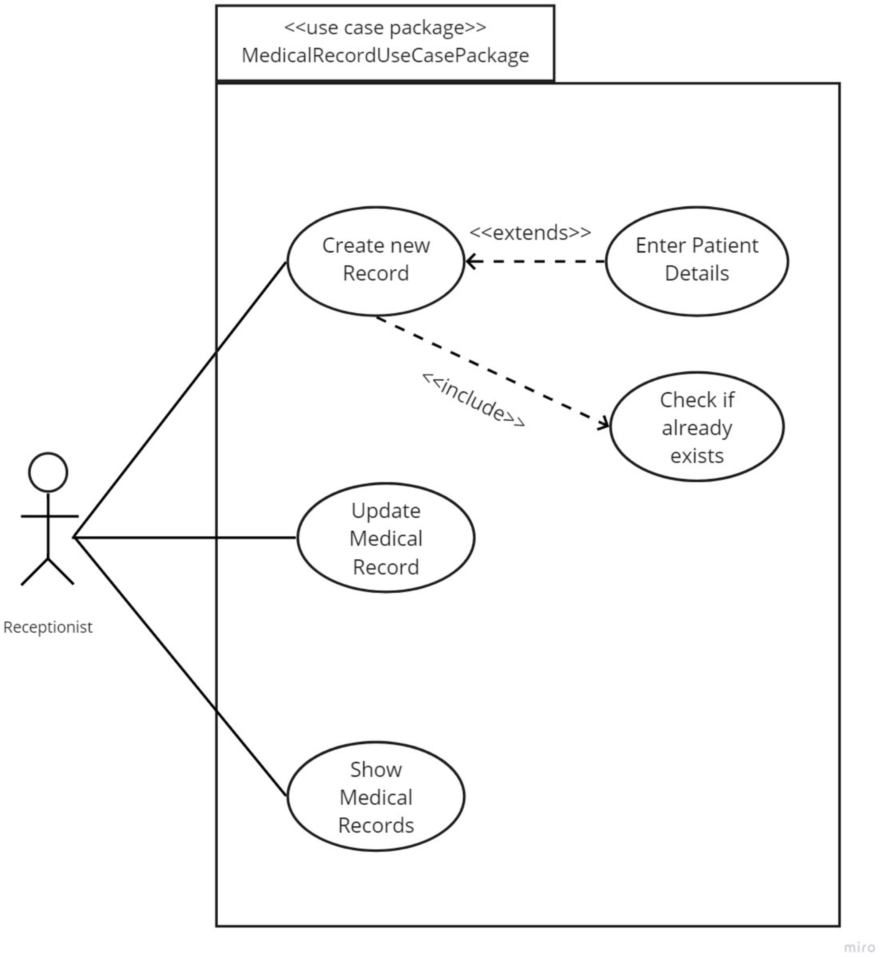


Figure 5.5

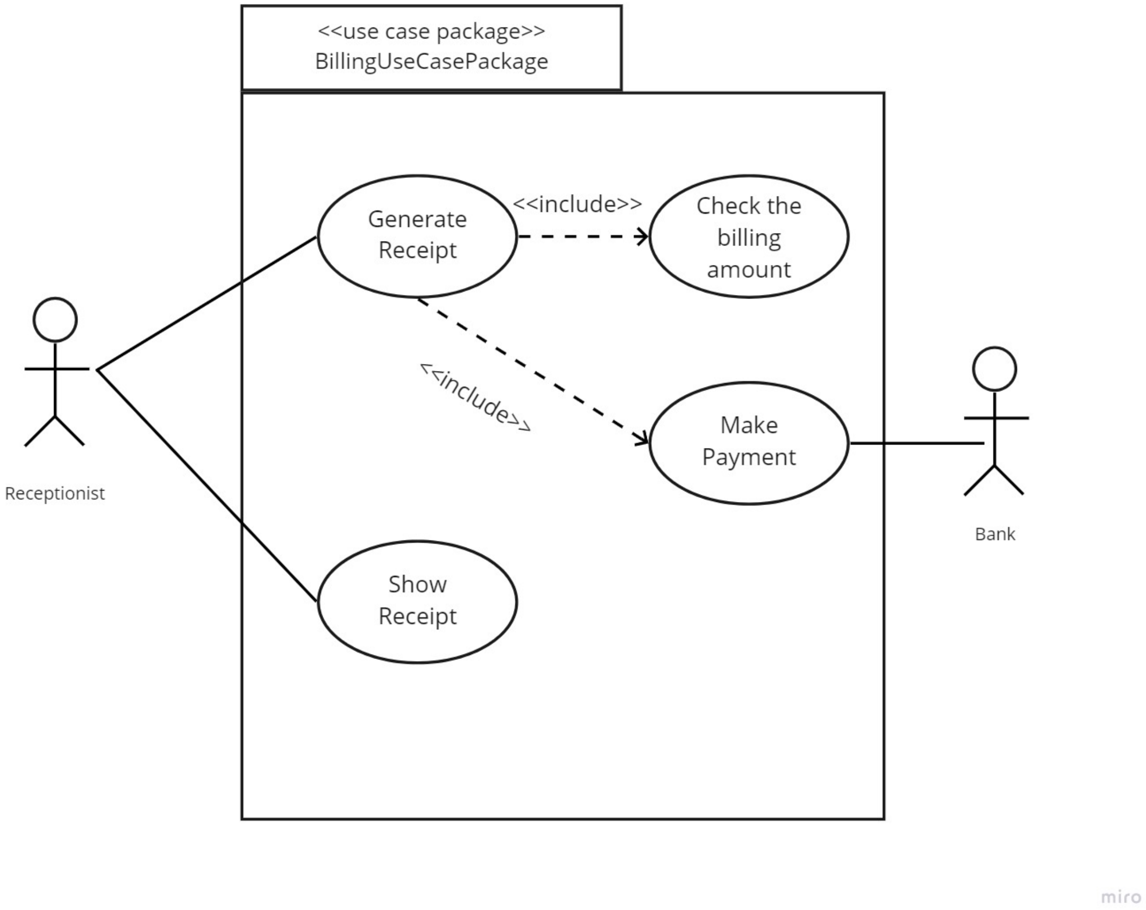

<<use case package>>
BillingUseCasePackage
Generate Receipt
<<include>>
Check the billing amount
<<include>>
Make Payment
Show Receipt
Receptionist
Bank
miro

Figure 5.6

Activity Diagrams

Book appointment:

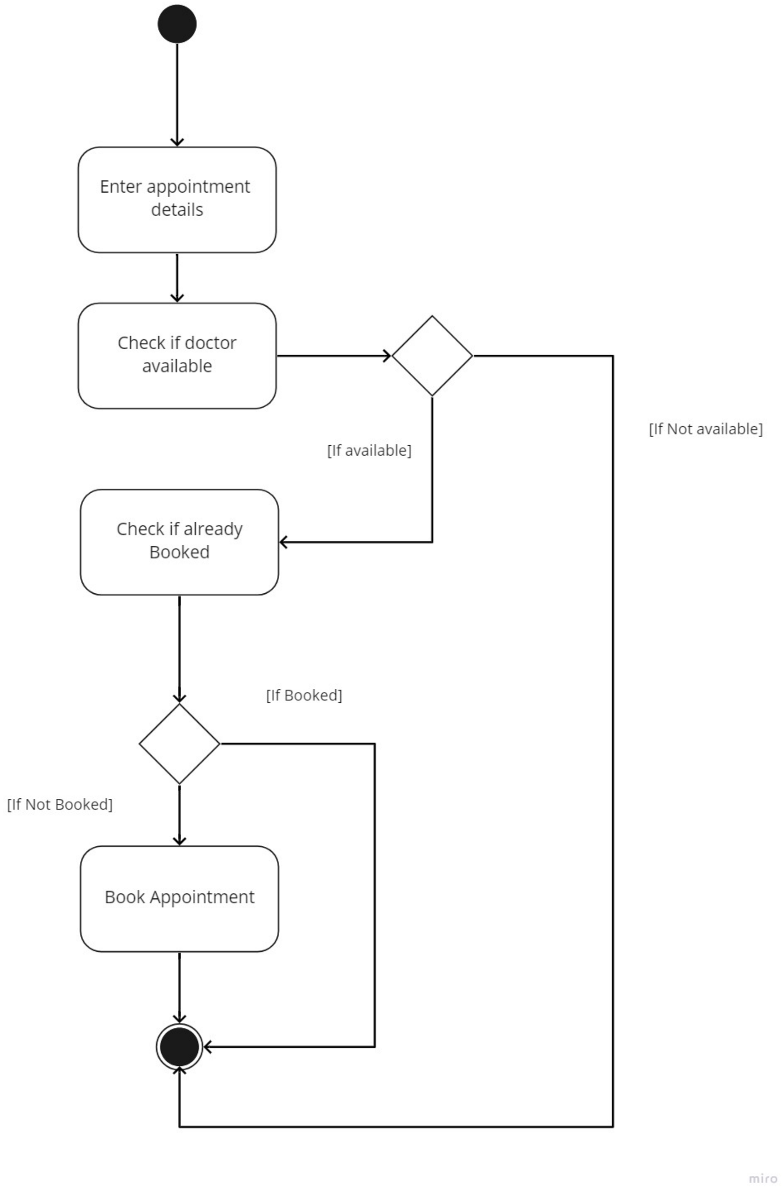


Figure 5.7

Register Patient:

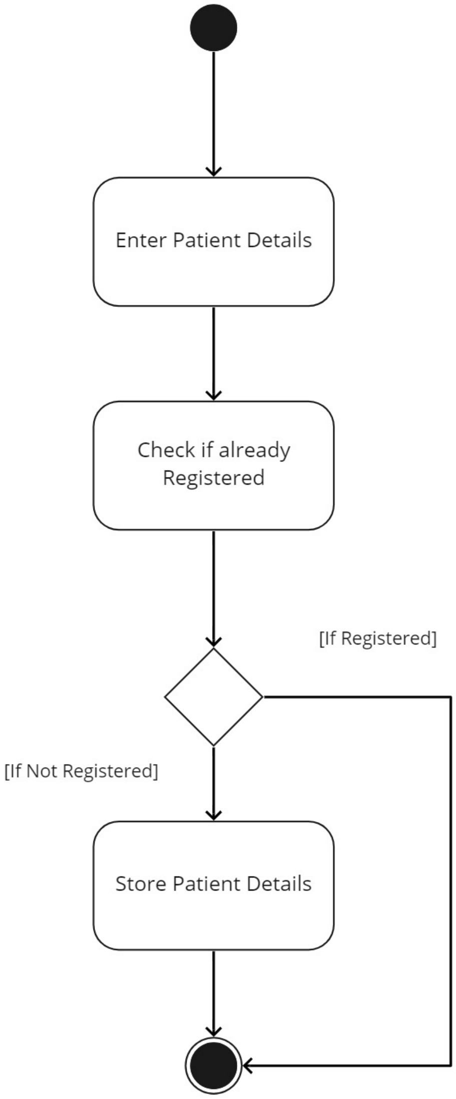


Figure 5.8

Pay Bill

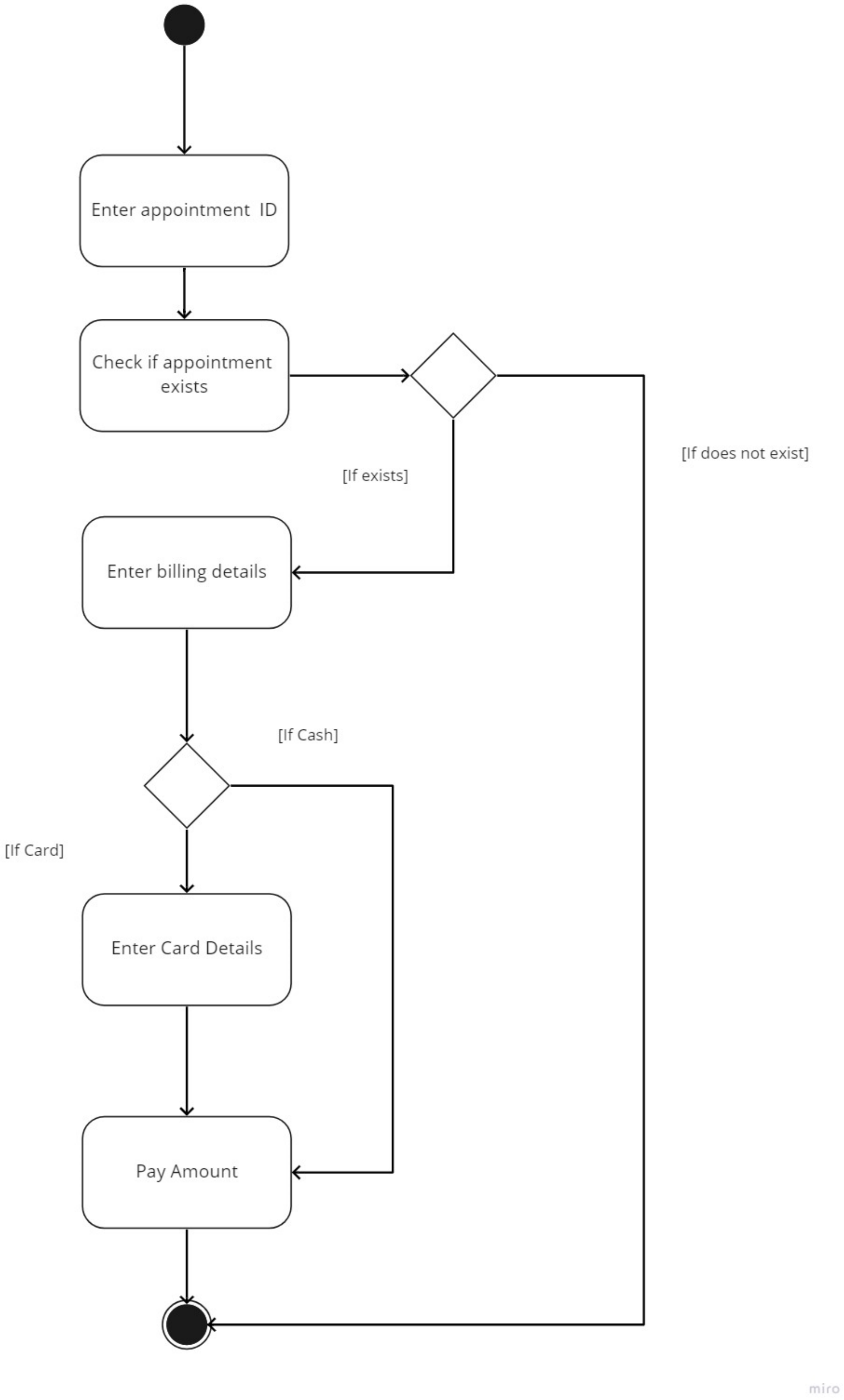


Figure 5.9

Patient Record

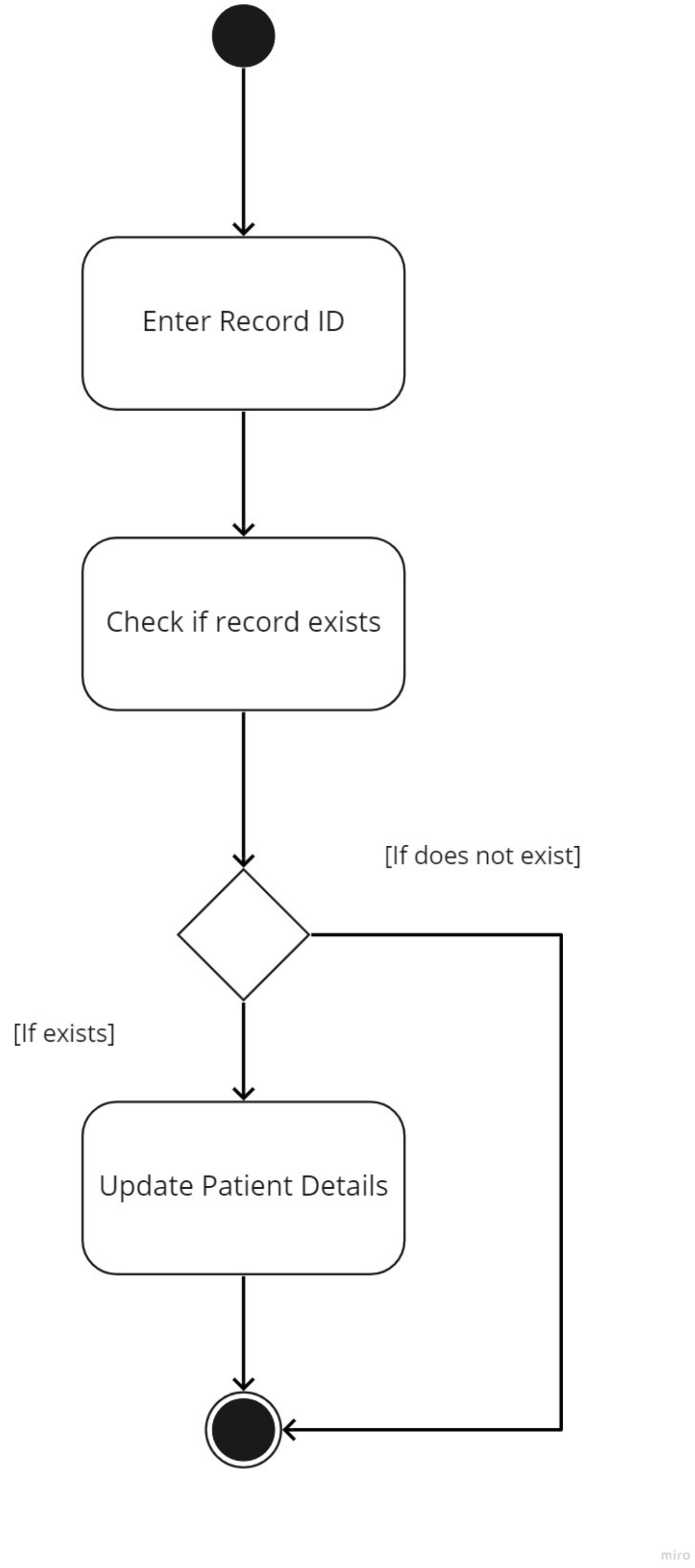
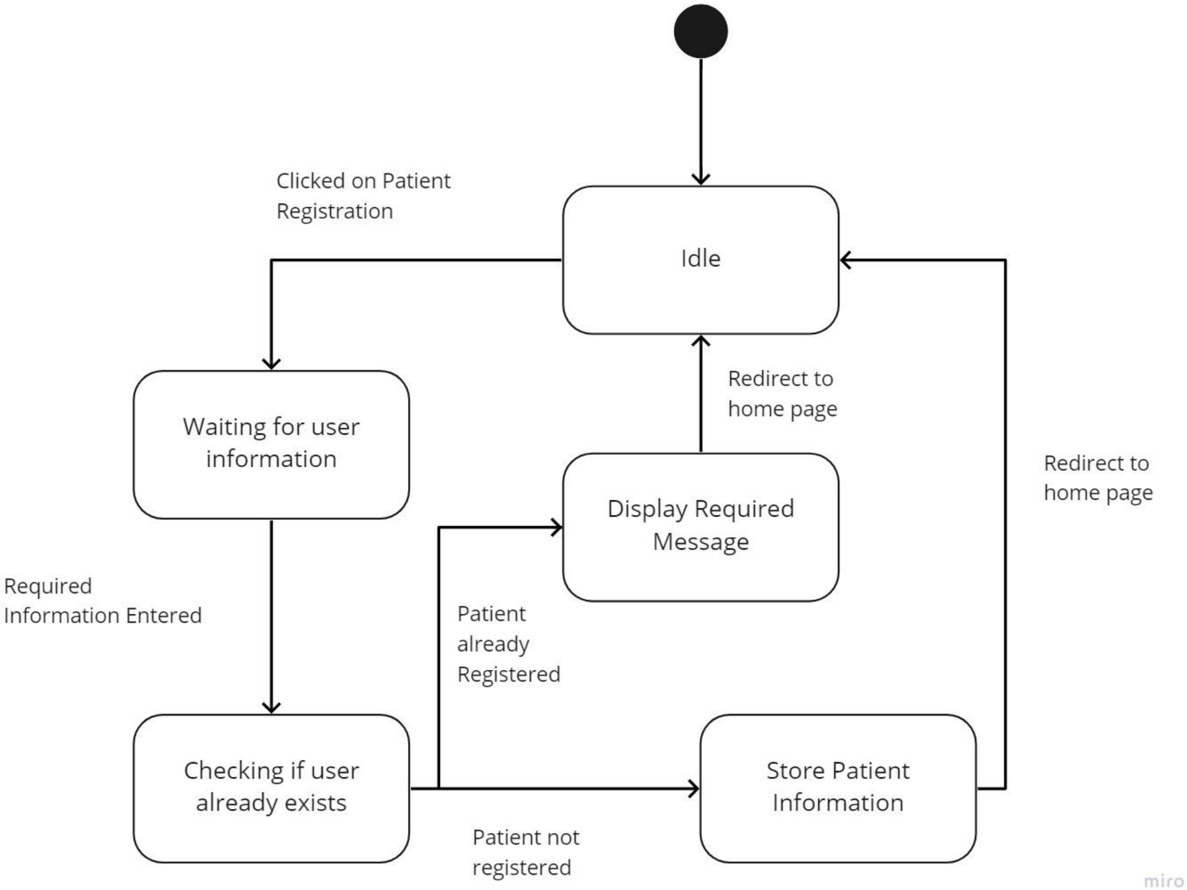


Figure 5.10

State chart Diagrams

Register Patient



Idle
Clicked on Patient Registration
Waiting for user information
Redirect to home page
Redirect to home page
Display Required Message
Required Information Entered
Patient already Registered
Checking if user already exists
Store Patient Information
Patient not registered
miro

Figure 5.11

Appointment Booking

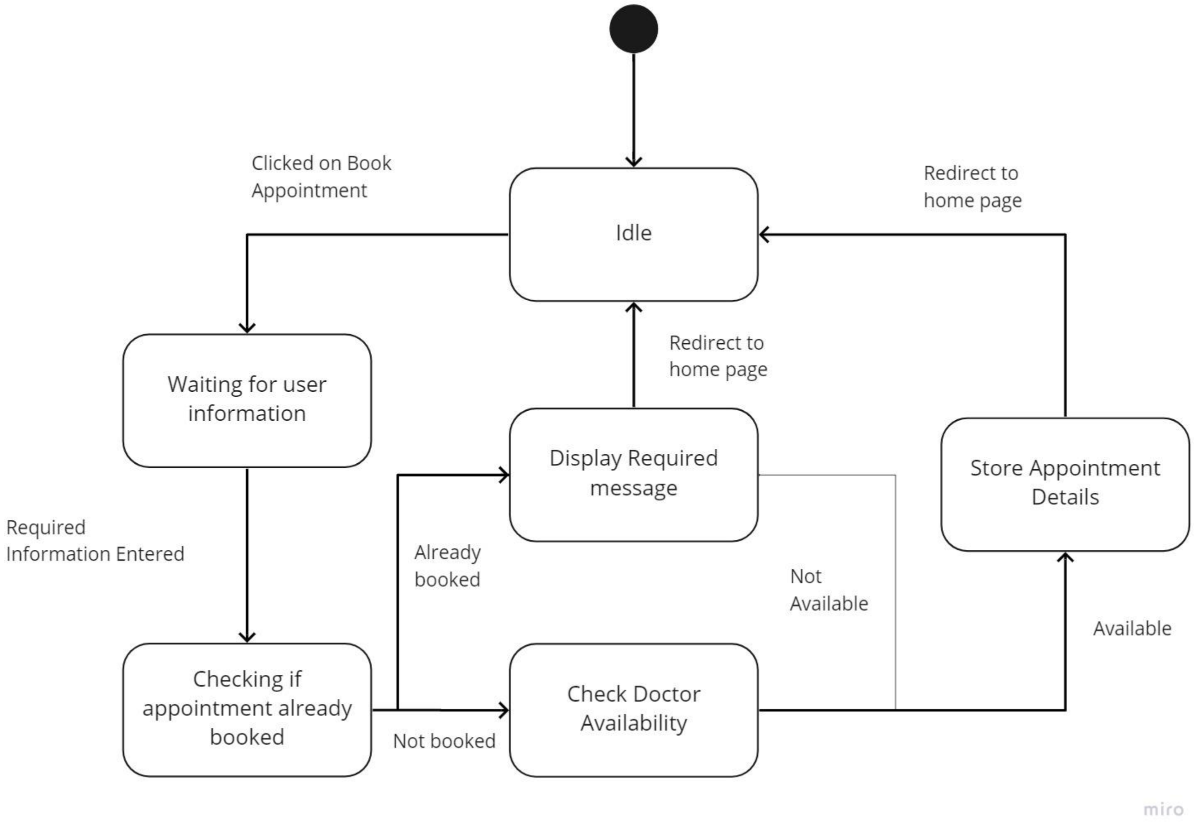

Idle
Clicked on Book Appointment
Redirect to home page
Waiting for user information
Redirect to home page
Display Required message
Store Appointment Details
Required Information Entered
Already booked
Not Available
Available
Checking if appointment already booked
Not booked
Check Doctor Availability
miro

Figure 5.12

Patient Medical Record

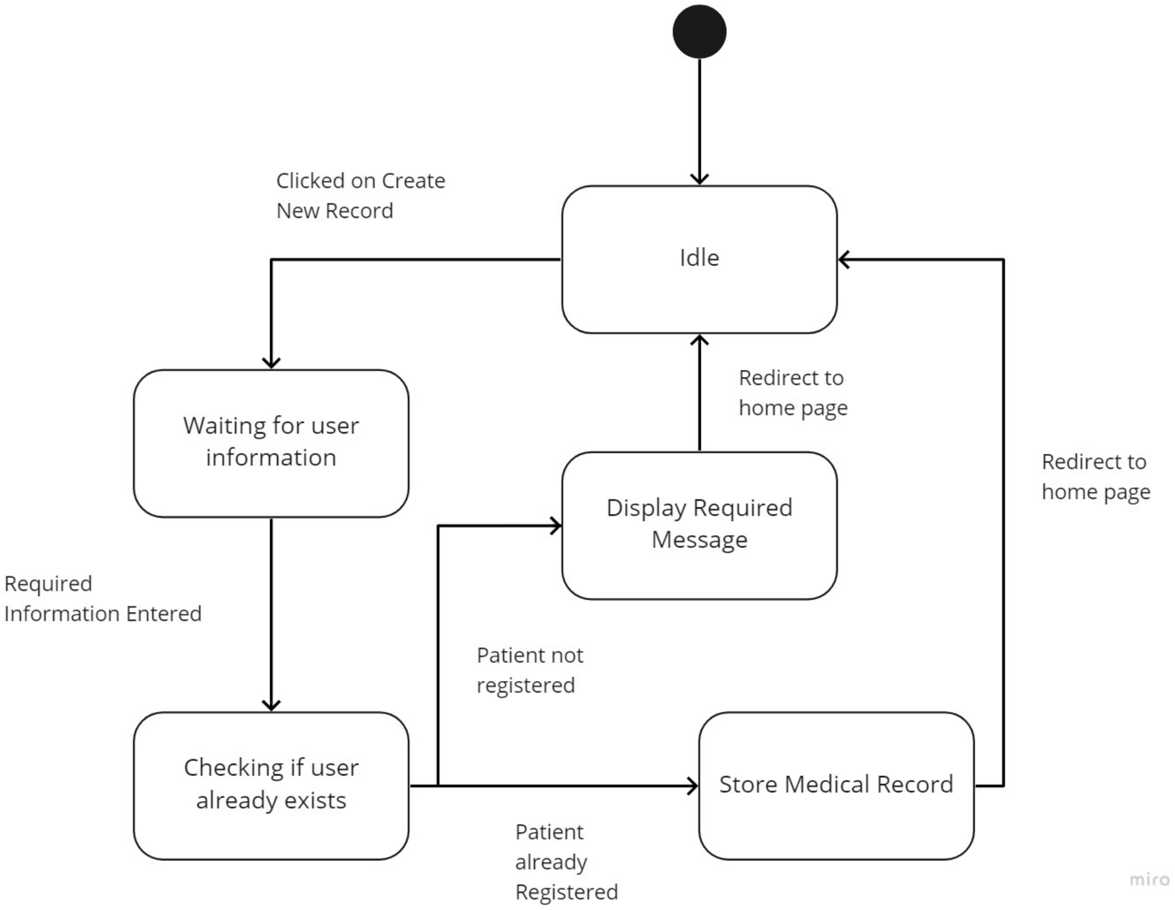

Clicked on Create New Record
Idle
Redirect to home page
Redirect to home page
Waiting for user information
Display Required Message
Required Information Entered
Patient not registered
Checking if user already exists
Store Medical Record
Patient already Registered
miro

Figure 5.13

Billing

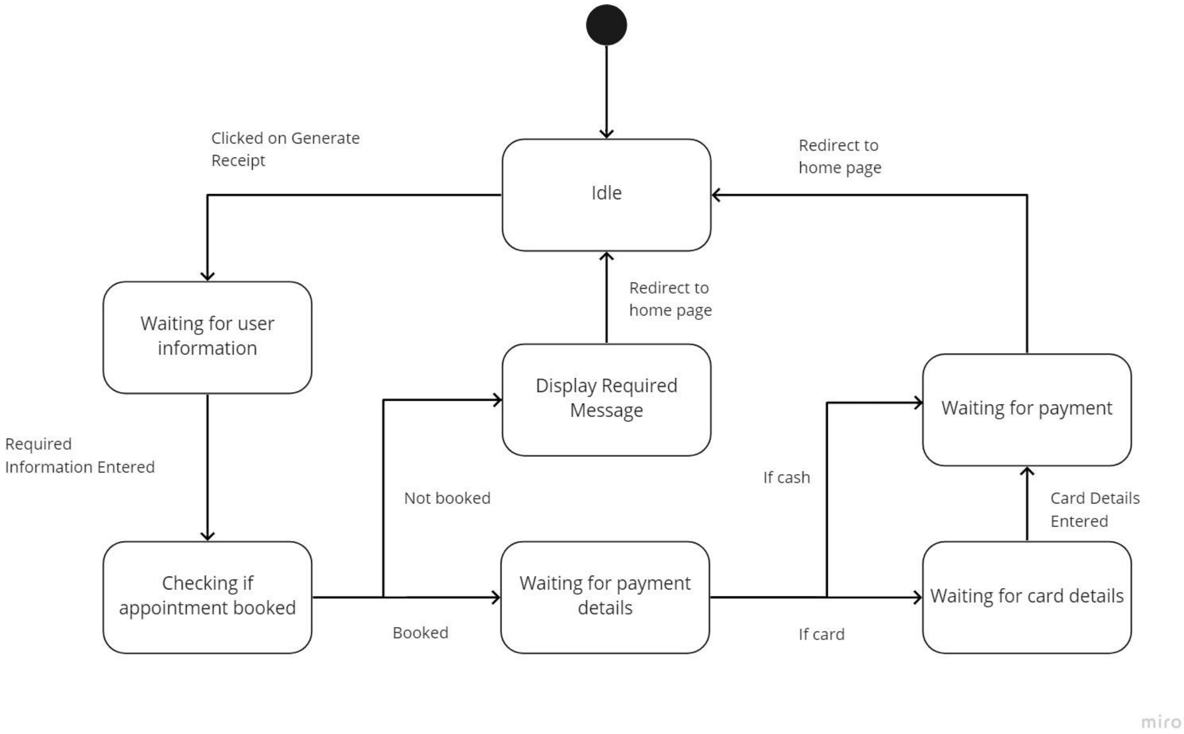


Figure 5.14

Communication Diagrams

Patient Registration

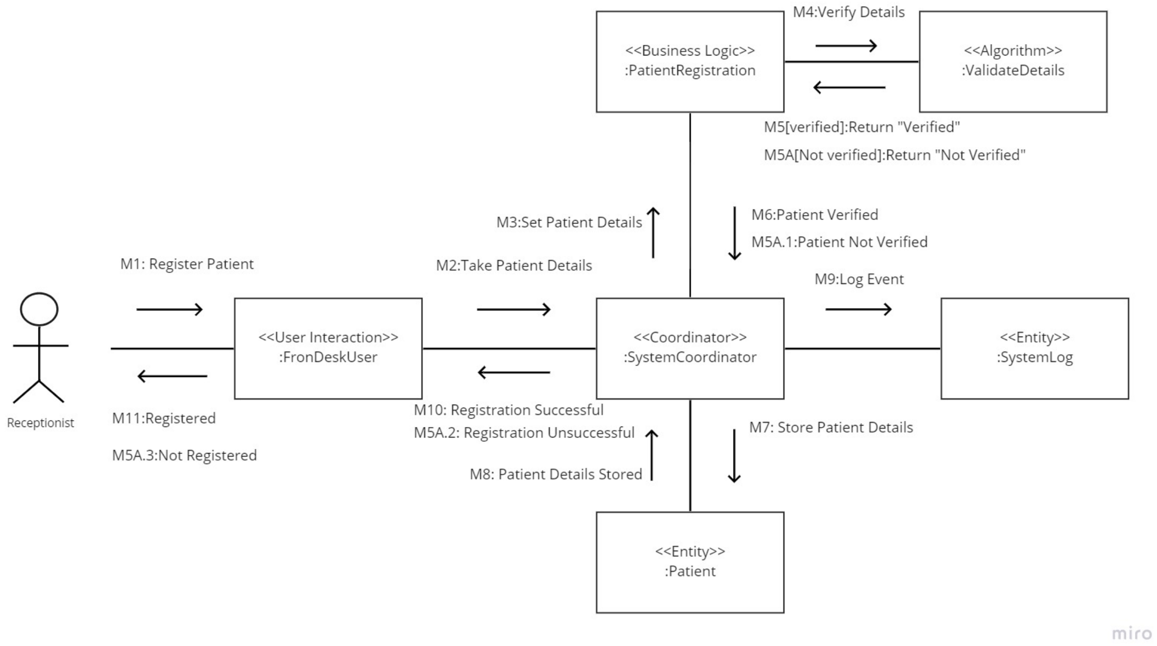


Figure 5.15

Create Medical Record

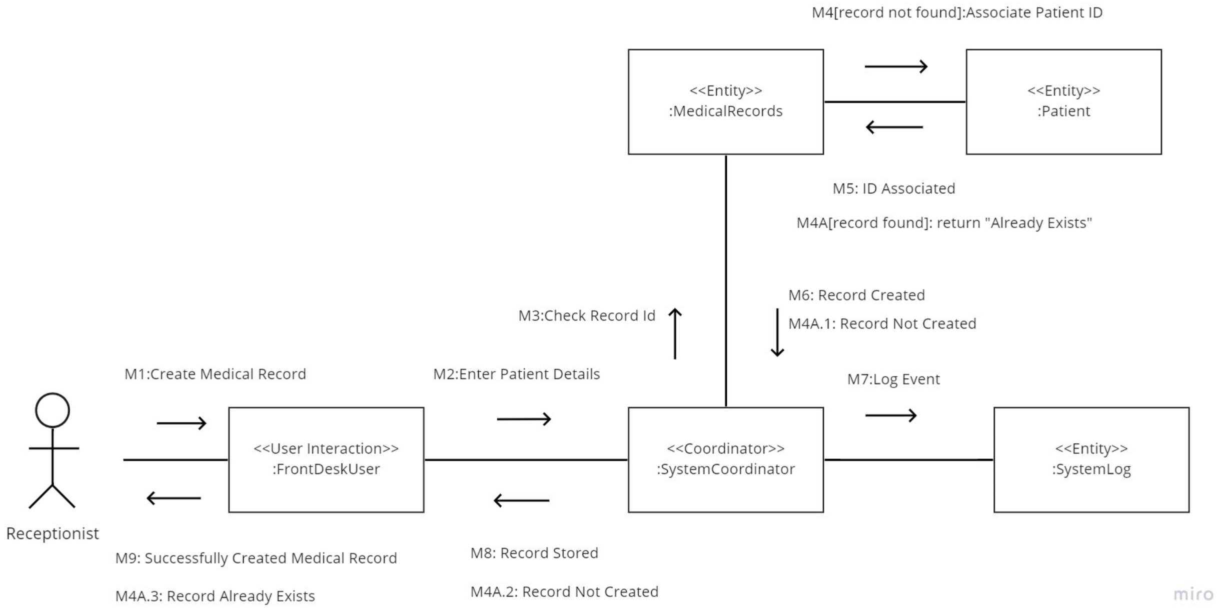


Figure 5.16

Class Diagram

Overview

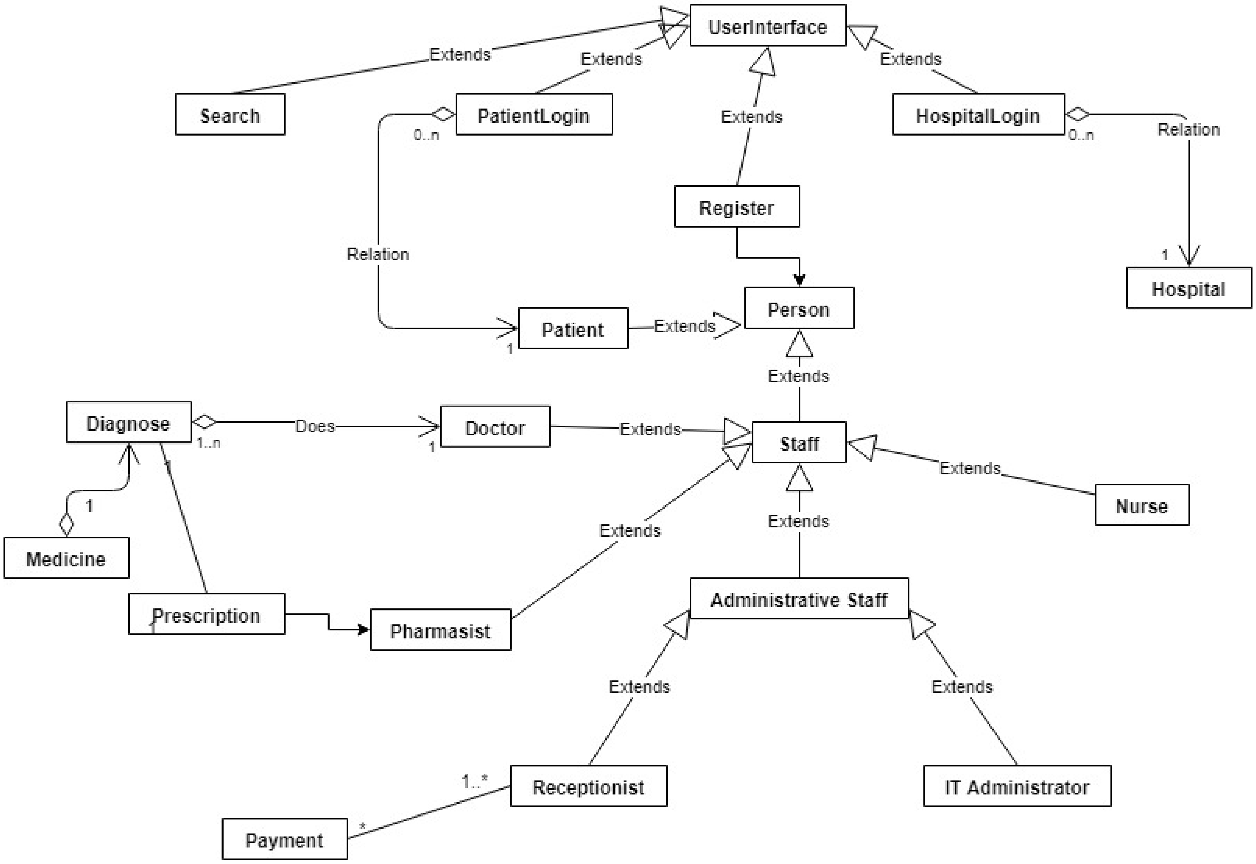


Figure 5.17

Elaborated

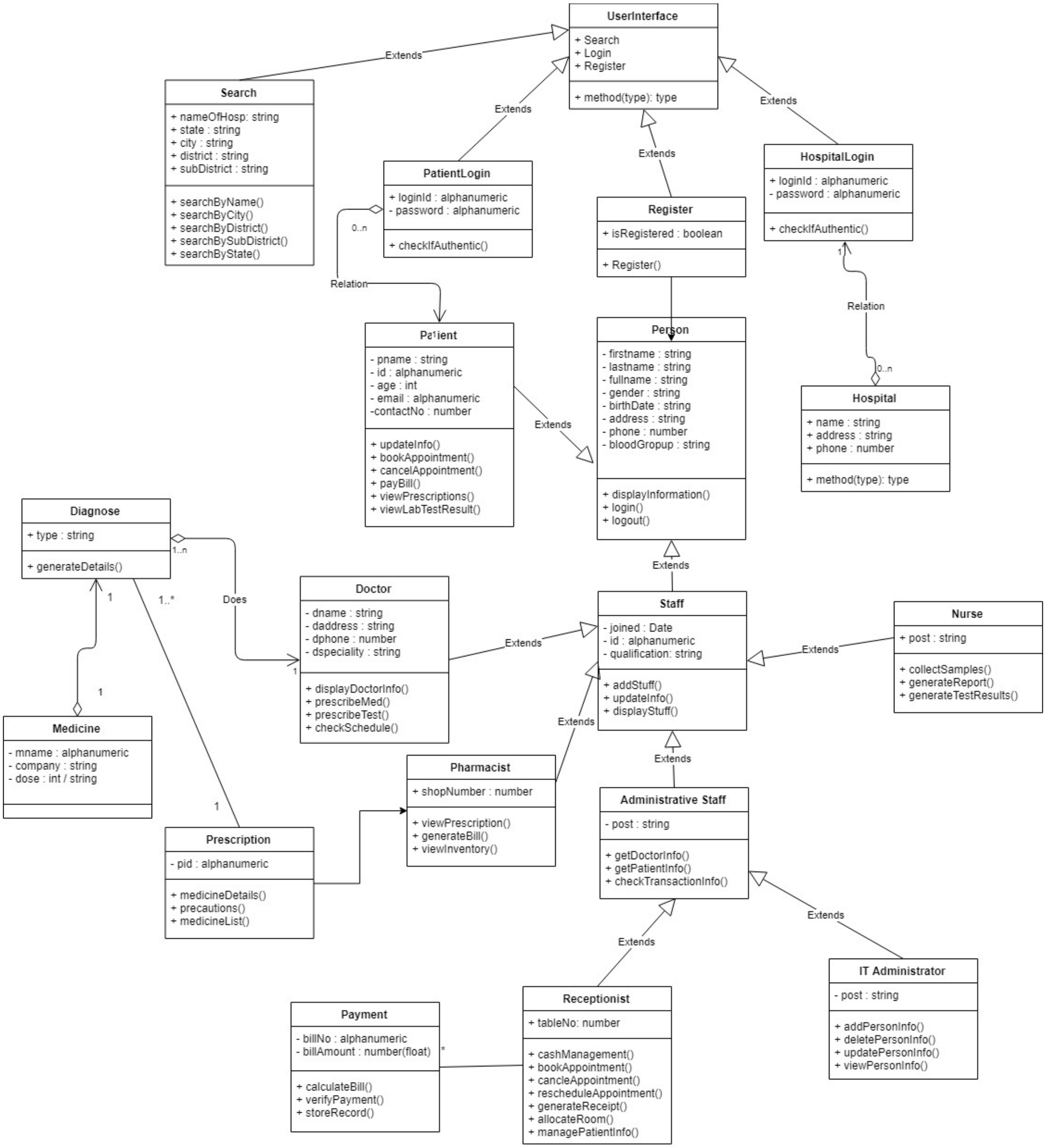


Figure 5.18

Test Specifications

Introduction

This project aims to develop a Healthcare Management System which will manage the medical, financial and administrative aspects related to a hospital. The system is expected to automate the existing manual processes of patients, doctors and appointment management.

Intended Audiences

This test plan is made for the system testing of the application. This plan will be referred by

1. Development and management team
2. Testing Team
3. Customers
4. Senior Managers

Intended Users

This application will be used by the receptionist and the admin who are responsible to handle the records.

Test Scope

System Testing will include the following testings -

1. User Interface Testing
2. Functionality Testing
3. Database Testing

Out of Scope Testing

1. Security Testing

2. Performance Testing
3. Usability Testing

Test Objectives

- Targeted Number of Defects: 20
- Targeted Number of Test Case Scenario: 7
- Targeted Number of Test Cases: 12
- Iterations Required: 3
- Test Cases Scenario Writing per day: NA
- Test Cases Writing per day: NA
- Test Cases Execution per day: NA

Assumptions of Testing

- User understands English and they are capable of using a computer.
- Users understand basic data entry on a computer.
- Applications will be working on Windows 10 and other operating systems, if any, used will be compatible.
- Application is tested on intel i7, 8GB/16 Ram, 3.7GHz systems. Other configurations used, if any, will be compatible.

Test Scenarios

Common for All Test Scenarios (TS00—Priority P1) :

The application can be launched by visiting the url where the webapp is hosted locally. It displays a home page with four call to action buttons named ‘Billing Section’, ‘Appointment Section’, ‘Medical Record’ and ‘Register Patient’. After clicking each button, you will be redirected to a different page intended for a specific function. There are several forms to enter the details which are connected to a mysql database. For each text input you have to write something there. At the bottom left of each page there is a ‘Back’ button which takes you to the previous page. After successfully filling any form, Pop-up will appear which will say ‘Success’. These pages also contain buttons which when pressed display the details of the ‘Appointments’, ‘Doctors’, ‘Patients’ and ‘Medical Records’. Size of the text boxes vary based on the page.Tab sequence is as follows, When the application is launched, the home page is opened. For each press of the button ‘Tab’ on the keyboard, it moves to the navigation buttons on the page. Similar results are seen on each successive page.

Scenario 1 (TS01)

To check if the receipt is generated or not. If the details are entered correctly in the form, then a window will pop-up showing ‘Successful’. Else if the form is not filled properly then the user will not be able to submit the form by clicking on the ‘Generate’ button.

Scenario 2 (TS02)

To check if the appointment is booked or not. If the details are entered correctly in the form, then a window will pop-up showing ‘Successful’. Else if the form is not filled properly then the user will not be able to submit the form by clicking on the ‘Register’ button.

Scenario 3 (TS03)

To check if the appointment is canceled or not. If the details are entered correctly in the form which contains a single text box then a new pop-up showing ‘Successful’ will appear. Else if the form is not filled properly then a pop will appear which indicates that correct details were not entered and asks to reenter the details.

Scenario 4 (TS04)

To check if the record is created or not. If the details are entered correctly in the form, then a window will pop-up showing ‘Successful’. Else if the form is not filled properly then the user will not be able to submit the form by clicking on the ‘Register’ button.

Scenario 5 (TS05)

To check if the record is updated or not. If the details are entered correctly in the form which contains a single text box then a new page will appear in which we will be asked to enter the new details of the patient’s medical records. Else if the form is not filled properly then a pop will appear which indicates that correct details were not entered and asks to reenter the details.

Scenario 6 (TS06)

To check if the patient is registered or not. If the details are entered correctly in the form, then a window will pop-up showing ‘Successful’. Else if the form is not filled properly then the user will not be able to submit the form by clicking on the ‘Register’ button.

Test Cases

Test case 1

Test Precondition:
The computer should be connected to the local network
The computer must be working
Test Sequence: Not Applicable
Test Scenario Traceability: TS00
Test Case Name and Number: UI/TEST1/0001
Type of Testing: User Interface Testing
Objectives: To check if the form fields are working as intended and if the form is loading
Valid/invalid Conditions: Valid condition

Priority: P1

Test Description: Clicking on the 'Generate Receipt' button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce:

- Go to the webapp
- Click on 'Billing Section'
- Click on 'Generate Receipt'

Expected Results: The page is loaded. The form fields are initiated as empty

Test case 2

Test Precondition:

Form is loaded
The fields are empty

Test Sequence: UI/TEST1/0001

Test Scenario Traceability: TS01

Test Case Name and Number: FUN/TEST1/0002

Type of Testing: Functionality testing

Objectives: To check whether correct data is entered into the relevant fields/text boxes

Valid/invalid Conditions: Valid condition

Priority: P1

Test Description: Clicking on the 'Generate Receipt' button brings up the form to be filled. After filling the form with the relevant details, the same is reflected in the fields.

Steps to Reproduce:

- Go to the webapp
- Click on 'Billing Section'
- Click on 'Generate Receipt'
- Enter the relevant details
- Click on 'Generate'

Expected Results: There is a pop-up message which reads 'Successful' and the data is added to the database.

Test case 3

Test Precondition :

The computer should be connected to the local network
The computer must be working

Test Sequence : Not Applicable

Test Scenario Traceability : TS02

Test Case Name and Number : UI/TEST2/0001

Type of Testing : User Interface Testing

Objectives : To check if the form fields are working as intended and if the form is loading

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Book Appointment’ button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Appointment Section’
- Click on Book Appointment’

Expected Results : The page is loaded. The form fields are initiated as empty

Test case 4

Test Precondition :

Form is loaded
The fields are empty

Test Sequence : UI/TEST2/0001

Test Scenario Traceability : TS02

Test Case Name and Number : FUN/TEST2/0002

Type of Testing : Functionality testing

Objectives : To check whether correct data is entered into the relevant fields/text boxes

Valid/ invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the 'Book Appointment' button brings up the form to be filled. After filling the form with the relevant details the same is reflected in the fields.

Steps to Reproduce :

- Go to the webapp
- Click on 'Appointment Section'
- Click on 'Book Appointment'
- Enter the relevant details
- Click on 'Register'

Expected Results : There is a pop up message which reads 'Successful' and the data is added to the database.

Test case 5

Test Precondition :

The computer should be connected to the local network
The computer must be working

Test Sequence : Not Applicable

Test Scenario Traceability : TS02

Test Case Name and Number : UI/TEST3/0001

Type of Testing : User Interface Testing

Objectives : To check if the form fields are working as intended and if the form is loading

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the 'Cancel Appointment' button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce :

- Go to the webapp
- Click on 'Appointment Section'

- Click on ‘Cancel Appointment’

Expected Results : The page is loaded. The form fields are initiated as empty

Test case 6

Test Precondition :

Form is loaded

The fields are empty

Test Sequence : UI/TEST3/0001

Test Scenario Traceability : TS02

Test Case Name and Number : FUN/TEST3/0002

Type of Testing : Functionality testing

Objectives : To check whether correct data is entered into the relevant fields/text boxes

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Cancel Appointment’ button brings up the form to be filled. After filling the form with the relevant details the same is reflected in the fields.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Appointment Section’
- Click on ‘Cancel Appointment’
- Enter the relevant details
- Click on ‘GO’

Expected Results : There is a pop up message which reads ‘Successful’ and the data is removed from the database.

Test case 7

Test Precondition :

The computer should be connected to the local network
The computer must be working

Test Sequence : Not Applicable

Test Scenario Traceability : TS03

Test Case Name and Number : UI/TEST4/0001

Type of Testing : User Interface Testing

Objectives : To check if the form fields are working as intended and if the form is loading

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Create New Record’ button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Patient Medical Record’
- Click on ‘Create New Record’

Expected Results : The page is loaded. The form fields are initiated as empty

Test case 8

Test Precondition :

Form is loaded
The fields are empty

Test Sequence : UI/TEST4/0001

Test Scenario Traceability : TS03

Test Case Name and Number : FUN/TEST4/0002

Type of Testing : Functionality testing

Objectives : To check whether correct data is entered into the relevant fields/text boxes

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Create New Record’ button brings up the form to be filled. After filling the form with the relevant details the same is reflected in the fields.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Patient Medical Record’
- Click on ‘Create New Record’
- Enter the relevant details
- Click on ‘Register’

Expected Results : There is a pop up message which reads ‘Successful’ and the data is added to the database.

Test case 9

Test Precondition :

The computer should be connected to the local network
The computer must be working

Test Sequence : Not Applicable

Test Scenario Traceability : TS03

Test Case Name and Number : UI/TEST5/0001

Type of Testing : User Interface Testing

Objectives : To check if the form fields are working as intended and if the form is loading

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Update medical Record’ button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Patient Medical Record’
- Click on ‘Update medical Record’

Expected Results : The page is loaded. The form fields are initiated as empty

Test case 10

Test Precondition :

Form is loaded
The fields are empty

Test Sequence : UI/TEST5/0001

Test Scenario Traceability : TS03

Test Case Name and Number : FUN/TEST5/0002

Type of Testing : Functionality testing

Objectives : To check whether correct data is entered into the relevant fields/text boxes

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the ‘Update Medical Record’ button brings up the form to be filled. After filling the form with the relevant details the same is reflected in the fields.

Steps to Reproduce :

- Go to the webapp
- Click on ‘Patient Medical Record’
- Click on ‘Update medical Record’
- Enter the relevant details
- Click on ‘Register’

Expected Results : There is a pop up message which reads ‘Successful’ and the data is added to the database.

Test case 11

Test Precondition :

The computer should be connected to the local network
The computer must be working

Test Sequence : Not Applicable

Test Scenario Traceability : TS04

Test Case Name and Number : UI/TEST6/0001

Type of Testing : User Interface Testing

Objectives : To check if the form fields are working as intended and if the form is loading

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the 'Register Patient' button brings up the form to be filled. The form fields appear empty when the page is loaded.

Steps to Reproduce :

- Go to the webapp
- Click on 'Register Patient'

Expected Results : The page is loaded. The form fields are initiated as empty

Test case 12

Test Precondition :

Form is loaded

The fields are empty

Test Sequence : UI/TEST6/0001

Test Scenario Traceability : TS04

Test Case Name and Number : FUN/TEST6/0002

Type of Testing : Functionality testing

Objectives : To check whether correct data is entered into the relevant fields/text boxes

Valid/invalid Conditions : Valid condition

Priority : P1

Test Description : Clicking on the 'Register Patient' button brings up the form to be filled. After filling the form with the relevant details the same is reflected in the fields.

Steps to Reproduce :

- Go to the webapp
- Click on 'Register Patient'

- Enter the relevant details
- Click on ‘Register’

Expected Results : There is a pop up message which reads ‘Successful’ and the data is added to the database.

Advantages and Disadvantages

Advantages

- The system will be able to manage and process the patient records electronically
- Digital data will be easy to analyze
- The system will be lightweight and can be installed on any modern computer or phone

Disadvantages

- Training will be necessary to be able to use the system effectively
- It will take time to shift to the digital system as there is huge amounts of data that has to be transferred manually from the written records

Future Scope

The project can be expanded to include many more functions. It could be used to maintain records like availability of medicines, availability of doctors and nurses. It could also be used to keep a track of the resources available in the hospital.

Conclusion

Thus, we have successfully implemented a Healthcare Management System and performed the required manual and automation testing.

References

- Software-Engineering 7th ED by Roger S. Pressman
- Ian Sommerville Software Engineering, 9th Edition 2011
- Software Testing, Principles, Techniques and Tools, MG Limaye