

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

## 

B. Tech. CE Semester – VI

Subject: Object Oriented Software Engineering

Project Title:

**HealthCare System**

By:

Maliha Nan - CE073 (17CEUOS109)

Riya Mistry - CE069 (17CEUOG003)

Guided by:

Prof. Prashant M. Jadav



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that Object Oriented Software Engineering Project entitled “**HealthCare System**” is the bona fide report of work carried out by

1. **Maliha Nan (17CEUOS109)**
2. **Riya Mistry (17CEUOG003)**

Guide

**Prof. Prashant M. Jadav**

Associate Professor of Department of Computer Engineering, Dharmsinh Desai University, Nadiad

HOD

**Dr. C. K. Bhensdadia**

Head of Department of Computer Engineering, Dharmsinh Desai

University, Nadiad

**Table of Contents**

[**Abstract**](#_vgyppn5jowdu) **4**

[Purpose](#_2yv7jdowpw1r) 4

[Scope](#_xkak1uk4k7) 4

[Overview](#_7sgnin35u3ew) 5

[**Introduction**](#_ss271sxm123a) **6**

[Brief Introduction](#_4qlqaexrge7k) 6

[Technology/Platform/Tools used](#_dq5gdaaorg5j) 6

[**Software Requirement Specifications**](#_ktdd6iypkgbg) **7**

[**Design**](#_m7qc18al1jnu) **12**

[Use Case Diagram](#_5xulnhhyshdi) 12

[Class Diagram](#_izwu0o4wjq61) 13

[Sequence Diagrams](#_inc1yuvz9zrm) 14

[Activity Diagrams](#_j0frjmiapurj) 16

[State Diagrams](#_rnonpyi7u0gm) 20

[E-R Diagram](#_vxik55fvws1y) 21

[Data Dictionary](#_4ta0rxpo629o) 22

[**Implementation Detail**](#_o5bldak7zuz9) **29**

[Modules created and a brief description of each module.](#_vaib5xv0upxa) 29

[Function prototypes which implements major functionality.](#_n78bpemms24q) 30

[**Testing**](#_khd1dskvhy7n) **31**

[**Screen-shots**](#_eg83tdgh36su) **32**

[**Conclusion**](#_55nnwf7zat62) **35**

[**Limitation and Future Extension**](#_6asg3tj5z3sc) **36**

[Limitation of the project](#_5ftopnp5vuvf) 36

[Functionality which was not implemented](#_k24y5579ow78) 36

[List possible future extension to your project](#_oz8b7vveq06p) 36

[**Bibliography**](#_ielrexk91vqe) **37**

# **Abstract**

Healthcare has become one of India’s largest sectors - both in terms of revenue and employment. Large volumes of personal health data are collected on a daily basis in places like public health centers and hospitals. These rich sources of data can be used to improve healthcare services. Given its importance and the possibility of it being used for surveillance or corporate interests, it is critical to establish a robust and sensitive data governance structure, suggests a study. Our system aims to work just for that. It is for the use of any party included in healthcare. If used, the system will centralize healthcare data of all the citizens in 1 database. It will also help reduce the amount of paperwork in the healthcare industry.

## **Purpose**

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirement with respect to end users. Also, we shall predict and sort out how we hope this system will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the system develops. In short, the purpose of this document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project’s target audience, its user interface and software requirements. It defines how our clients, team and end user see the system and its functionality.

## **Scope**

The scope of the system is centralizing healthcare data and maintaining a single portal for updating this data. It is of use for any user who works for healthcare including doctors, pharmacists, lab technicians and also patients. It does not identify any specific method, nomenclature or tool for preparing an SRS.

## **Overview**

The remaining sections of this document provide a general description, including characteristics of the users of this project, the functional and data requirements of the system

# **Introduction**

## **Brief Introduction**

Care+ is a desktop application whose main aim is to centralize the healthcare data of all patients. There are 4 user types: Doctor, Pharmacist, Lab Technician and Patient. The doctor can add and manage cases for patients, can add prescriptions in these cases and view the history of the patient. The patient can also check his history. The pharmacist can see and mark prescriptions. The lab technician can see and mark reports.

## **Technology/Platform/Tools used**

* + 1. Technology
       - .Net
       - Microsoft SQL Server
    2. Platform
       - Windows
    3. Tools
       - Visual Studio

# **Software Requirement Specifications**

**Pharmacist functionalities**

R1.1 Login

Input: username, password

Output: login successful

R.1.2 Registration

Input: pharmacy-name, password, re-enter password, license-proof, owner-name, email, identity-proof, address, city, state

Output: Generate unique pharmacy Id

Processing: Validations of all the entries

R.1.3 Give prescribed medicine

Input: patient-Id

Output: list of medicines to be given

R.1.3.1 Select required prescription

Input: case-Id, date

Output: list of medicines

R.1.3.2 Mark the given medicine

Input: pharmacist-id, medicine-name

Output: medicine-name will be shown with green-tick indicating given.

**Doctor functionalities**

R2.1 Login

Input: username, password

Output: login successful

Processing: if username and password are valid then he/she will be logged into the system otherwise proper error messages will be shown.

R.2.2 Register

Input: Doctor-name, Date-of-birth, email, license, university, category, subcategory, nationality, identity-proof, hospital-name, address-of-clinic, city, state

Output: generate Doctor-Id

Processing: validations of all entries

R.2.3 View Patient History

Description: doctor will only be able to view the history of the patient and not be able to modify it. Only if the case belongs to the doctor he/she would have the rights to modify it.

R.2.3.1 View individual patient history

Input: Patient-id

Output: list of all diseases that occurred to the patient in the past will be displayed.

R.2.3.2 Check currently taking medicine

Input: Patient-Id

Output: list of medicines taken by patient will be displayed.

R.2.3.3 Check if suffering from

Input: disease-name

Output: case of the patient with that disease will be displayed if exists else no such history will be displayed.

R.2.4 Add New Case

Input: Patient-Id, Disease-name, category-of-disease, disease-name, Date-of-occurrence, Doctor-Id, Date-of-closure, overview, last-modified-date, case-type (normal, severe, moderate), report-needed (true/false)

Output: generate Case-id

R.2.4.1 Give report description

Input: case-id, report-name, description

Output: display added successfully

R.2.4.2 Give case prescription

Description: doctor will enter the required medicine-name along with how many it is to be taken during the day.

Input: medicine-name, date, morning, afternoon, evening

Output: Prescription-id is generated

R.2.5 Update case

State: doctor must be logged in

R.2.5.1 Find case

Input: Case-Id

Output: All case details till now

R.2.5.2 Add new prescription

Input: patient-id, case-id, medicine-details, date

Output: new entry added successfully

R.2.5.3 Add new report-description

Input: patient-id, case-id, report-description, date

Output: add new entry successfully

R.2.6 View Report

Input: report-id, case-id

Output: display the report

**Patient functionalities**

R3.1 Login

Input: username, password

Output: login successful

R.3.2 Register

Input: name, email, sex, password, re-enter password, identity-proof, city, address, state, country, contact-number

Output: Generate Patient-Id

R.3.3 Update Details

Input: name, email, sex, password, re-enter password, identity-proof, city, address, state, country, contact-number

Output: modified the entries successfully will be displayed

**Lab-Technician functionalities**

R4.1 Login

Input: username, password

Output: login successful

R.4.2 Register

Input: name, license-proof, category, university, works-at, address

Output: Generate lab-Id

R.4.3 Add New Report

Input: Patient-Id, Case-Id, report-name, Report-description, lab-id, test-type, date, disease-name, report-doc, is-disease-prevalent (true/false)

Output: Generate Report-id

**Admin Functionalities**

R.5.1 Login

Input: username, password

Output: logged in successfully

R.5.2 Validate User

Input: user-Id

Output: display license certificate

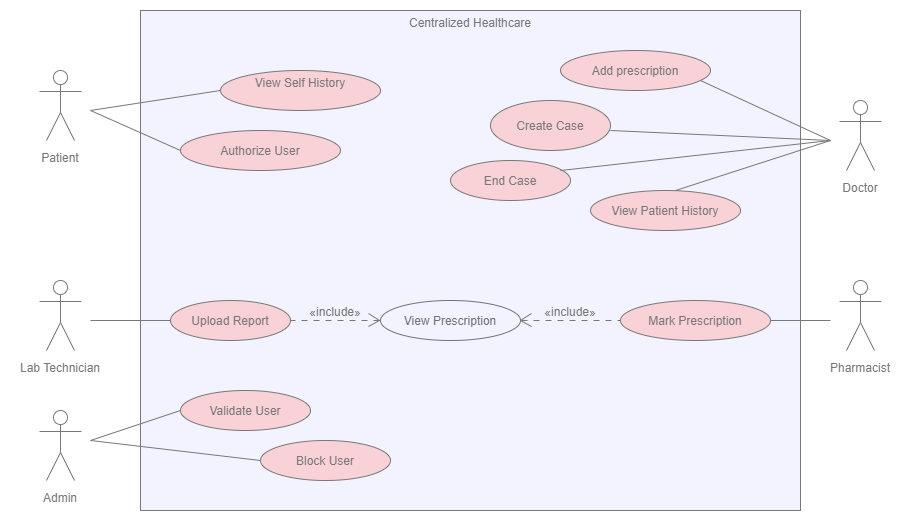
R.5.3 Block User

Input: user-id

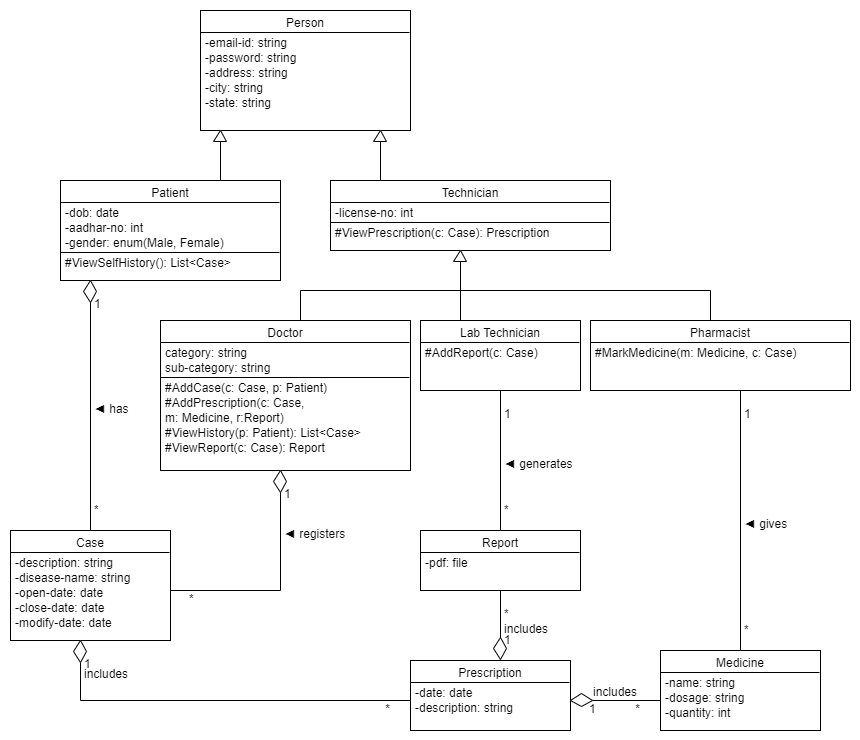
Output: block the user

# **Design**

## **Use Case Diagram**

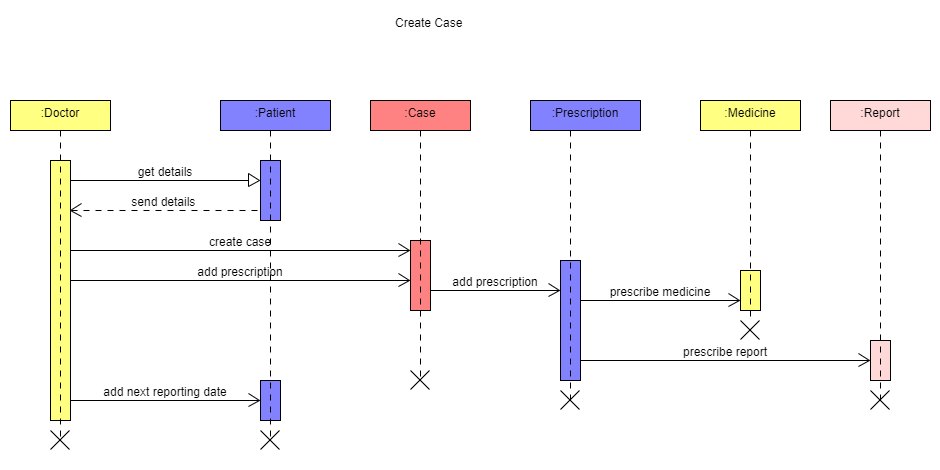


## **Class Diagram**

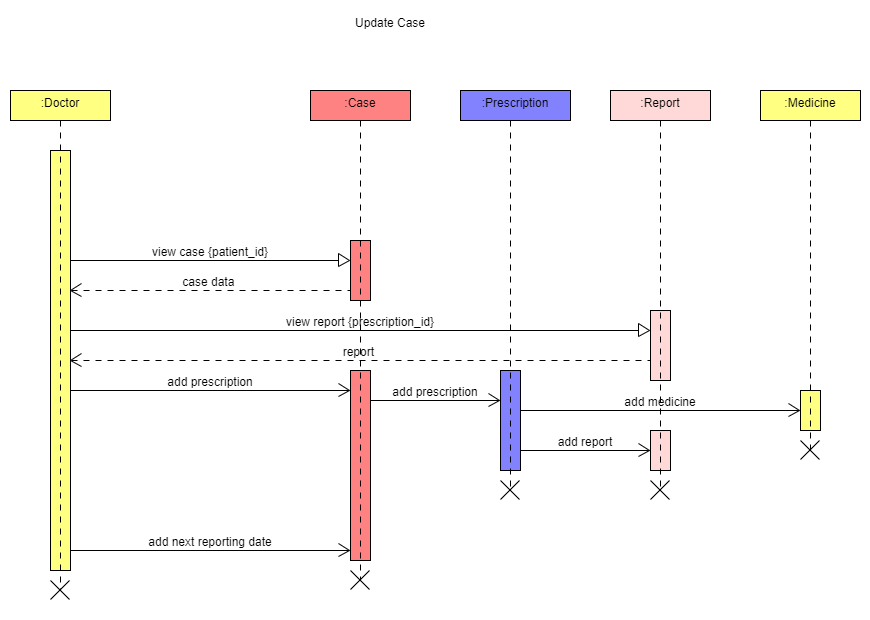


## **Sequence Diagrams**

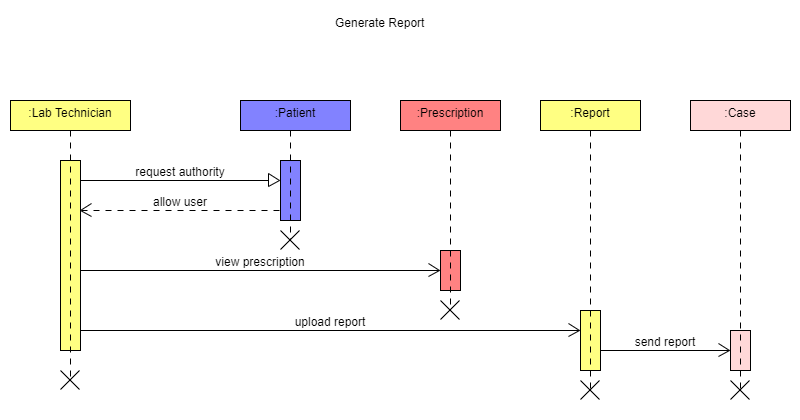
* + 1. Create Case



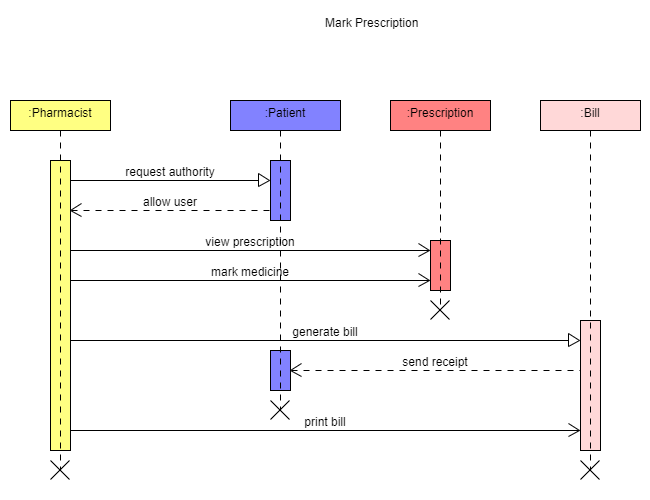
* + 1. Update Case



* + 1. Generate Report

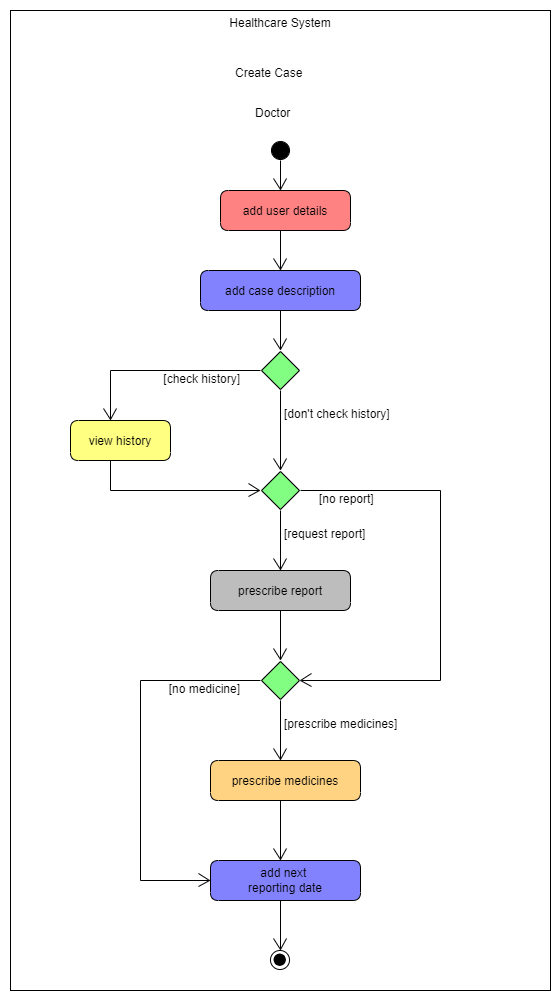


* + 1. Mark Prescription

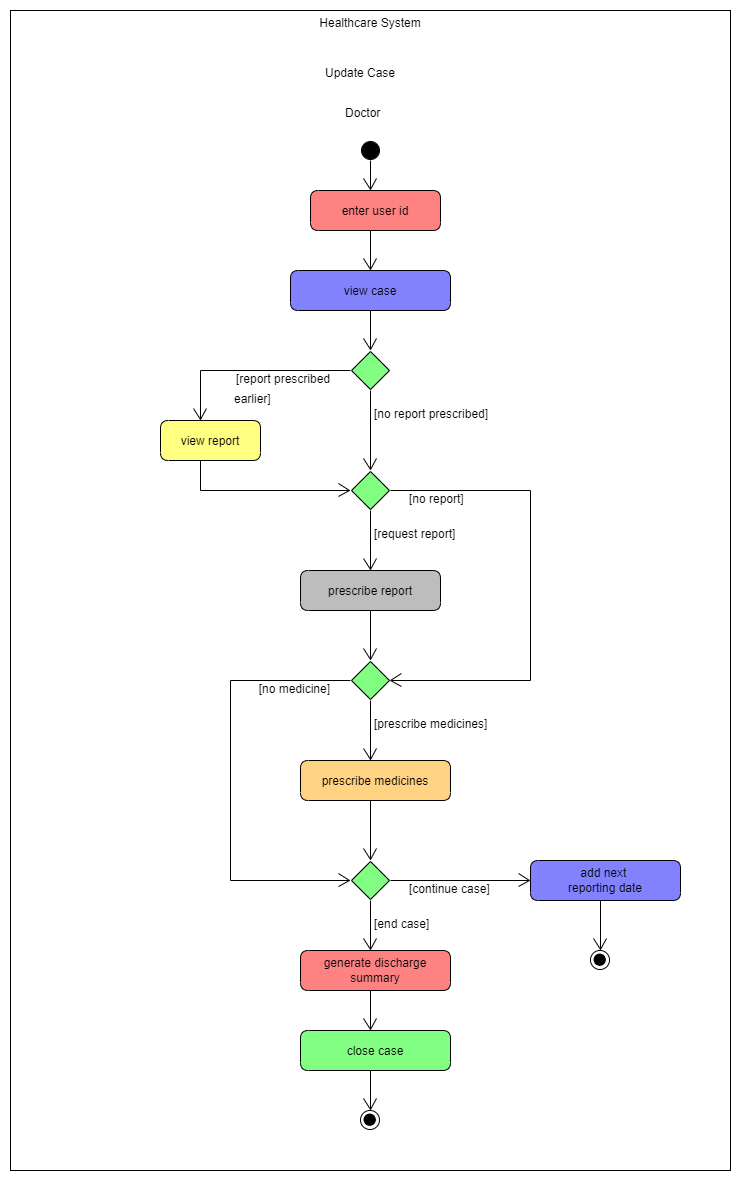


## **Activity Diagrams**

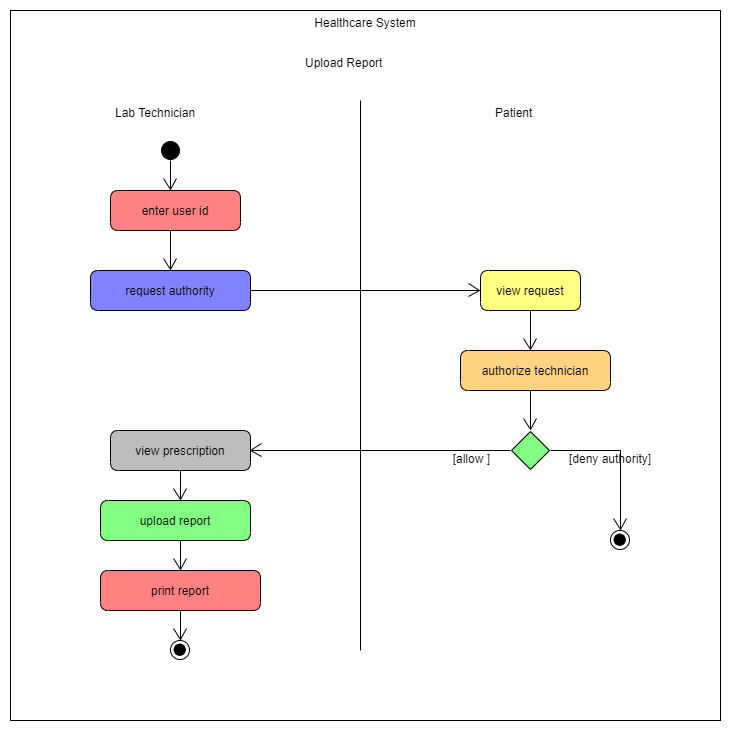
* + 1. Create Case



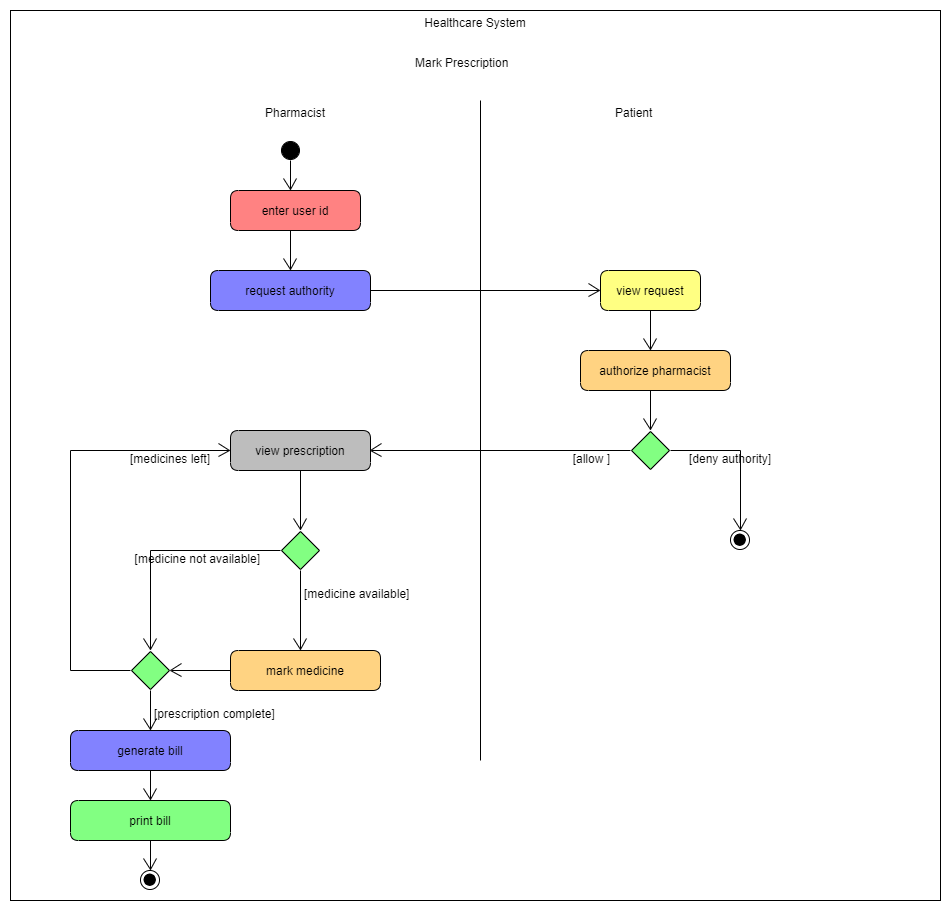
* + 1. Update Case



* + 1. Upload Report

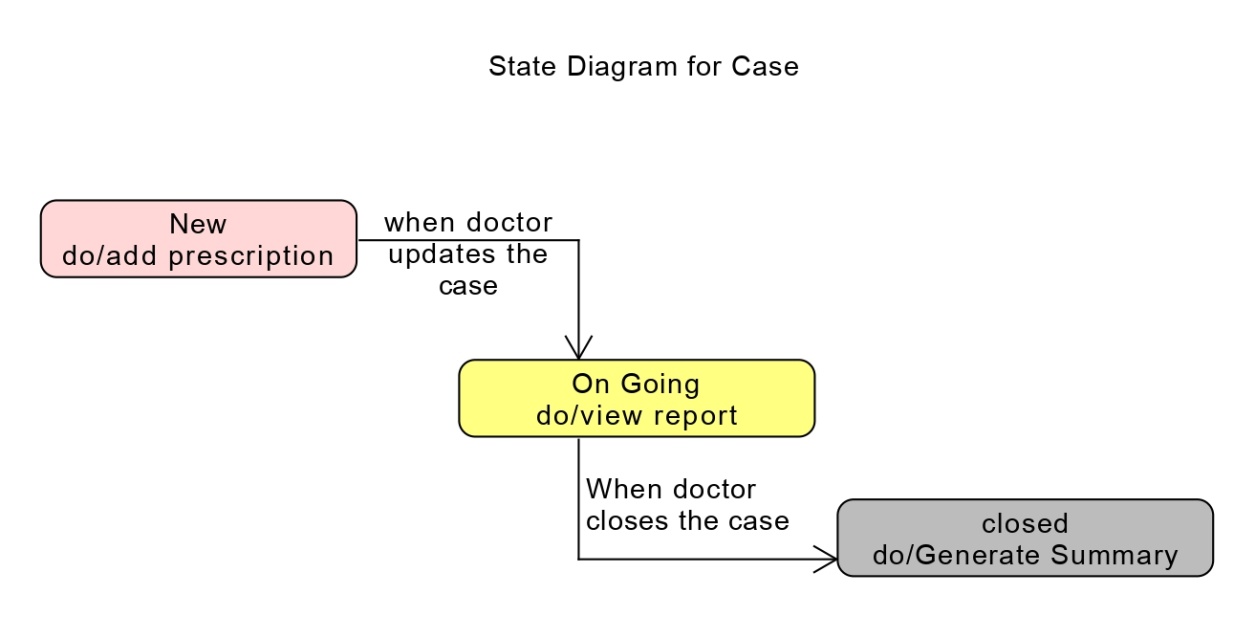


* + 1. Mark Prescription

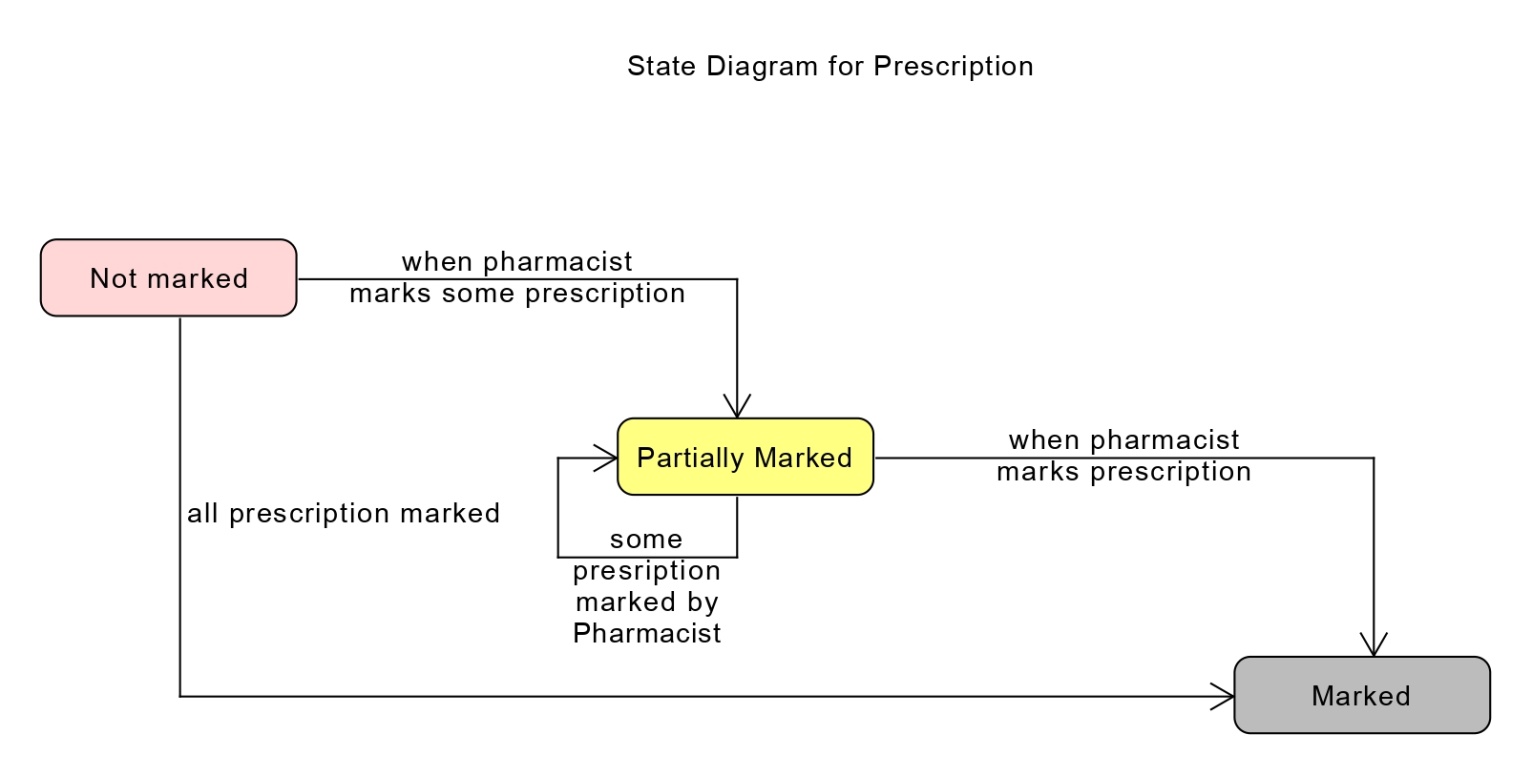


## **State Diagrams**

* + 1. Case State Diagram



* + 1. Prescription State Diagram



## **E-R Diagram**

## 

## **Data Dictionary**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doctor | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | DoctorId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | password | Varchar2 | 15 | Yes |  |  |  |  |
| 3. | licenseNo | Varchar2 |  | Yes | Yes |  |  |  |
| 4. | Dob | Date |  | Yes |  |  |  |  |
| 5. | emailId | Email |  | Yes | Yes |  |  |  |
| 6. | Address | Varchar2 |  | Yes |  |  |  |  |
| 7. | City | Varchar2 |  | Yes |  |  |  |  |
| 8. | State | Varchar2 |  | Yes |  |  |  |  |
| 9. | category | Varchar2 |  | Yes |  |  |  |  |
| 10. | Sub-category | Varchar2 |  | Yes |  |  |  |  |
| 11. | Name | Varchar2 | 15 | Yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Patient | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | PatientId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | password | Varchar2 | 15 | Yes |  |  |  |  |
| 3. | AadharCardNo | Varchar2 |  | Yes | Yes |  |  |  |
| 4. | Dob | Date |  | Yes |  |  |  |  |
| 5. | emailed | Email |  | Yes | Yes |  |  |  |
| 6. | Address | Varchar2 |  | Yes |  |  |  |  |
| 7. | City | Varchar2 |  | Yes |  |  |  |  |
| 8. | State | Varchar2 |  | Yes |  |  |  |  |
| 9. | Name | Varchar2 | 15 | yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lab-Technician | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | technicianId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | password | Varchar2 | 15 | Yes |  |  |  |  |
| 3. | licenseNo | Varchar2 |  | Yes | Yes |  |  |  |
| 4. | Dob | Date |  | Yes |  |  |  |  |
| 5. | emailId | Email |  | Yes | Yes |  |  |  |
| 6. | Address | Varchar2 |  | Yes |  |  |  |  |
| 7. | City | Varchar2 |  | Yes |  |  |  |  |
| 8. | State | Varchar2 |  | Yes |  |  |  |  |
| 9. | Name | Varchar2 | 15 | yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pharmacist | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | PharmacistId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | Name | Varchar2 | 15 | yes |  |  |  |  |
| 3. | password | Varchar2 | 15 | Yes |  |  |  |  |
| 4. | licenseNo | Varchar2 |  | Yes | Yes |  |  |  |
| 5. | Dob | Date |  | Yes |  |  |  |  |
| 6. | emailId | Email |  | Yes | Yes |  |  |  |
| 7. | Address | Varchar2 |  | Yes |  |  |  |  |
| 7. | City | Varchar2 |  | Yes |  |  |  |  |
| 8. | State | Varchar2 |  | Yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Case | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | Caseid | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 1. | PatientId | Varchar2 | 10 | Yes | Yes | FK | Patient |  |
| 2. | DoctorId | Varchar2 | 10 | Yes | Yes | FK | Doctor |  |
| 3. | Case description | Varchar2 | 1000 | Yes |  |  |  |  |
| 4. | Disease name | Varchar2 | 15 | yes |  |  |  |  |
| 5. | openDate | date |  | yes |  |  |  |  |
| 6. | modifyDate | date |  | yes |  |  |  |  |
| 7. | closedate | date |  | yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prescription | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | PrescriptionId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | CaseID | Varchar2 | 10 | Yes |  | FK | Case |  |
| 3. | Date | Varchar2 | 20 | Yes | No |  |  |  |
| 4. | description | Varchar2 | 1000 | Yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Medicine | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | MedicineId | Varchar2 | 10 | Yes | Yes | PK |  |  |
| 2. | PrescriptionId | varchar2 | 4 | Yes |  | FK | Prescription |  |
| 3. | Medicine name | Varchar2 | 20 | yes | No |  |  |  |
| 4. | Dosage | Varchar2 | 50 | No | yes |  |  |  |
| 5. | quantity | number |  | yes |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Report | | | | | | | | |
| SR.no | Field name | Data type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | ReportId |  |  |  |  | PK |  |  |
| 2. | CaseId | varchar2 | 10 | Yes |  | FK | Case |  |
| 3. | TechnicianId | Varchar2 | 10 | Yes |  | FK | Lab-Technician |  |
| 4. | pdf | file |  | Yes |  |  |  |  |

# **Implementation Detail**

The system is implemented as Windows Forms Desktop Application. It follows MVC pattern. It implements Factory Method Design Pattern for creating users.

## **Modules created and a brief description of each module.**

* + 1. View Cases

This module allows viewing cases of a particular patient or a particular doctor. It will show the details of a case and can also view a single case in more detail

* + 1. Manage Case

This module allows for editing the case by a doctor. The doctor can change details of a case and can also add or edit prescriptions.

* + 1. View History

This module views the history of a patient. It allows the doctor to check the cases that the patient had in the past and details about them.

* + 1. View Prescription

This module views the current prescription of the patient to the pharmacist and details about reports to the lab technician.

* + 1. Mark Prescription

This module is for the pharmacist or the lab technician to mark the prescribed medicine/report as bought.

## **Function prototypes which implements major functionality.**

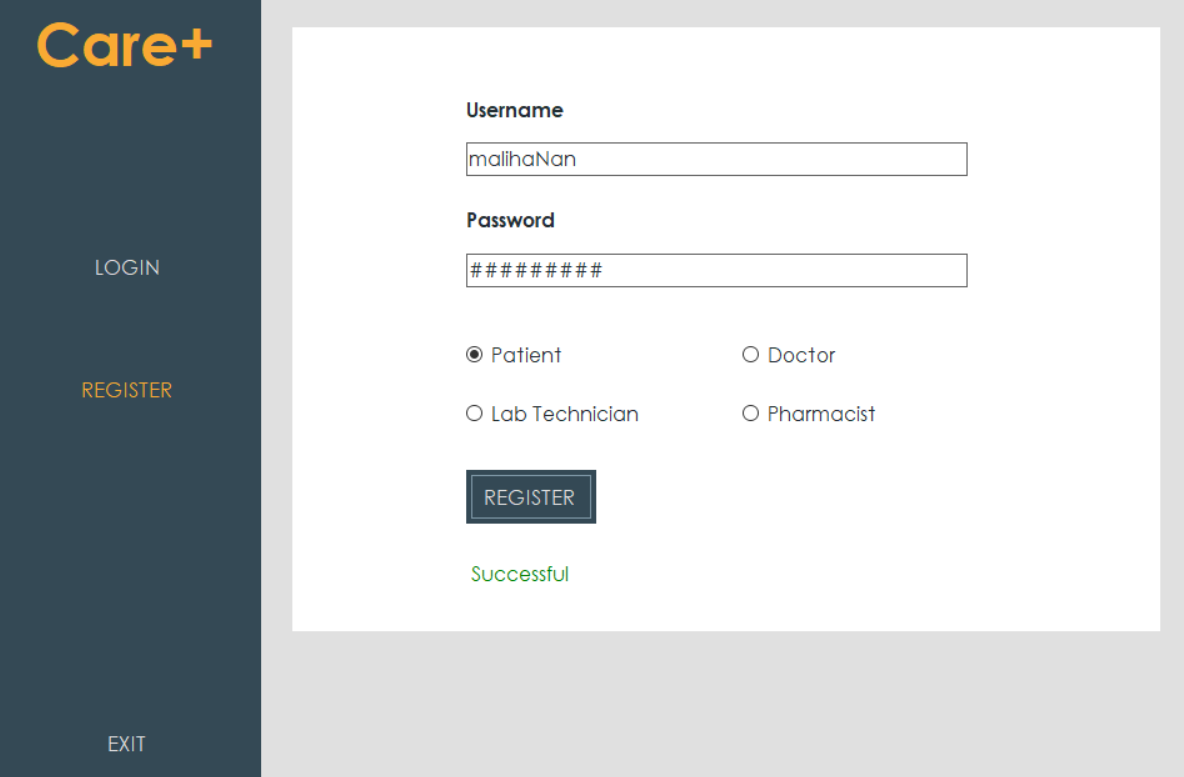
* + - * public bool AddCase(Case c)
      * public bool EditCase(Case c)
      * public bool CloseCase(int id, string closingSummary)
      * public bool AddPrescription(Prescription p)
      * public bool giveMedicine(int medicineId, int patientId)
      * public bool AddUser(User user)
      * public User LoginUser(User user)

# **Testing**

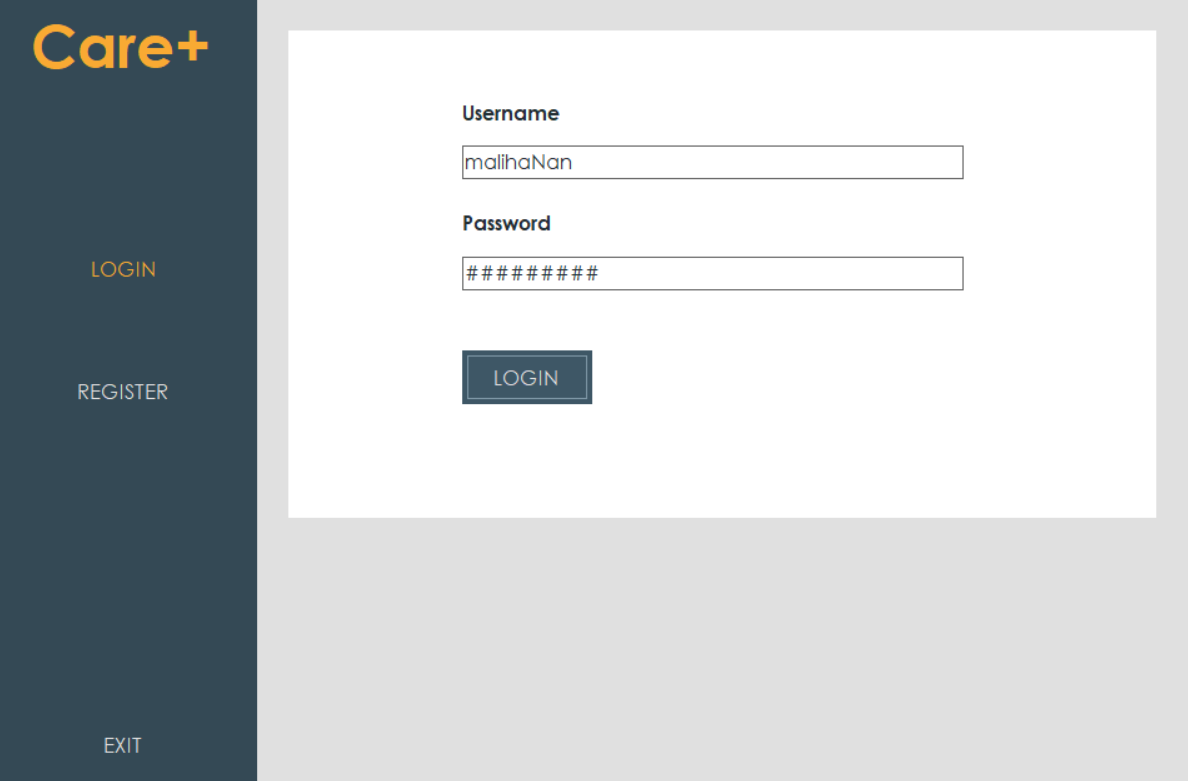
Unit Testing was done as each module was created and then it was integrated with the whole system. Then integration testing was done to check if the modules worked fine together. After the implementation was complete, a final testing was done to ensure the system works correctly.

# **Screen-shots**

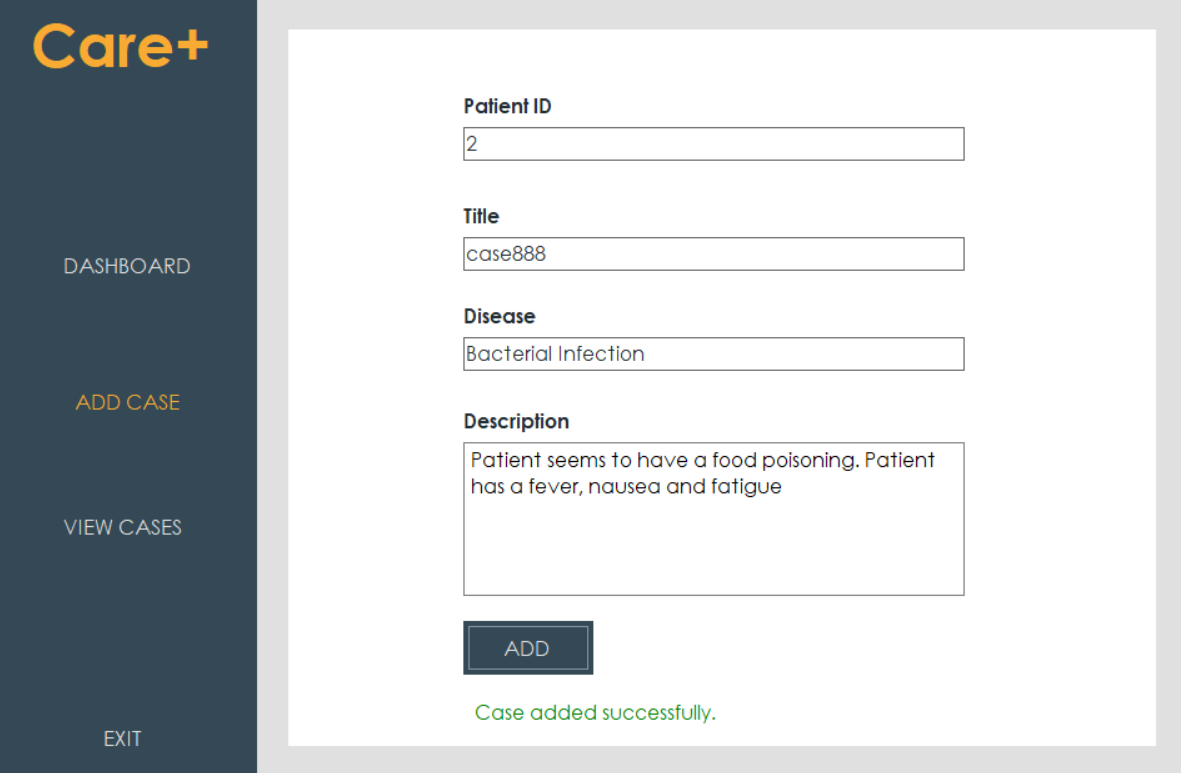
* 1. User Registration



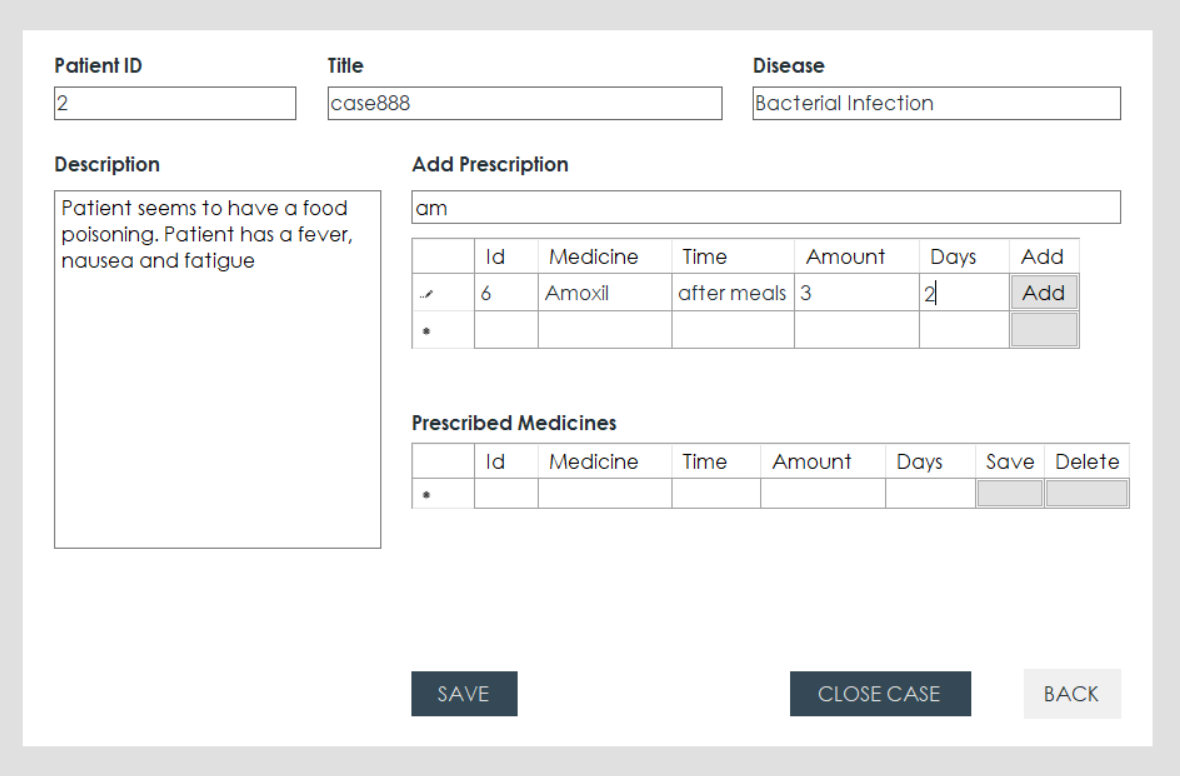
* 1. User Login



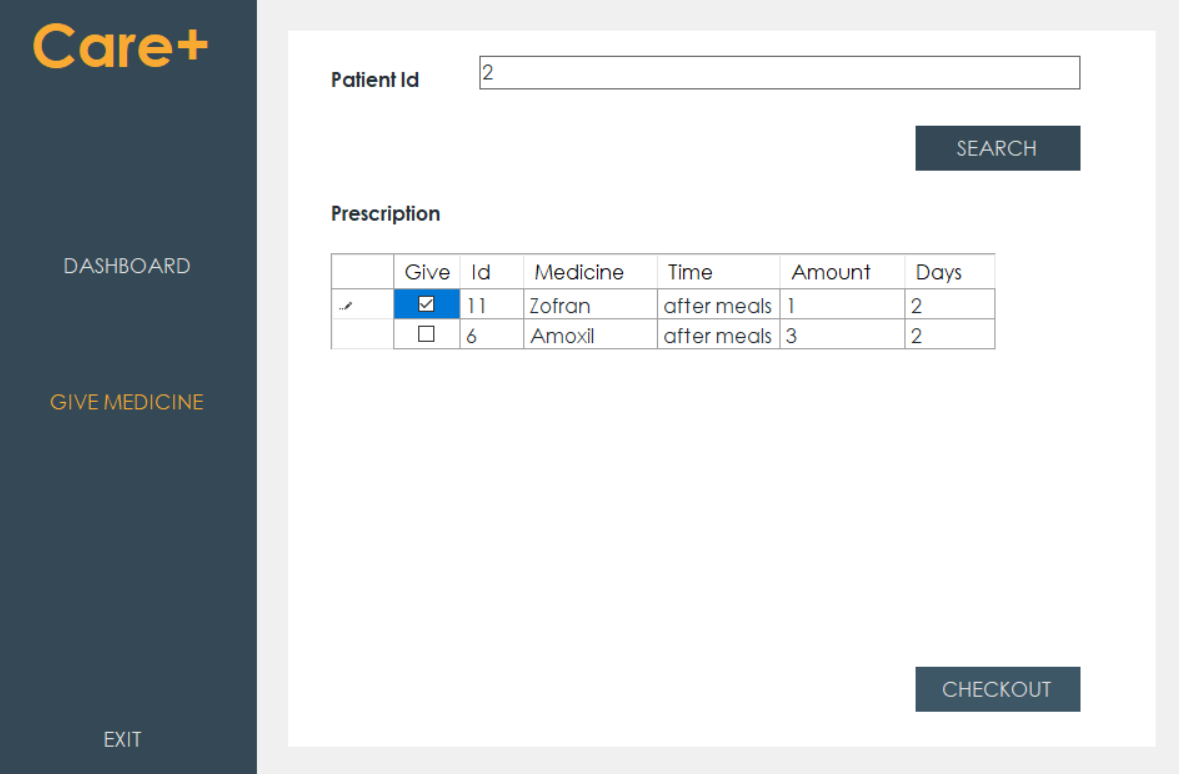
* 1. Doctor: Add Case



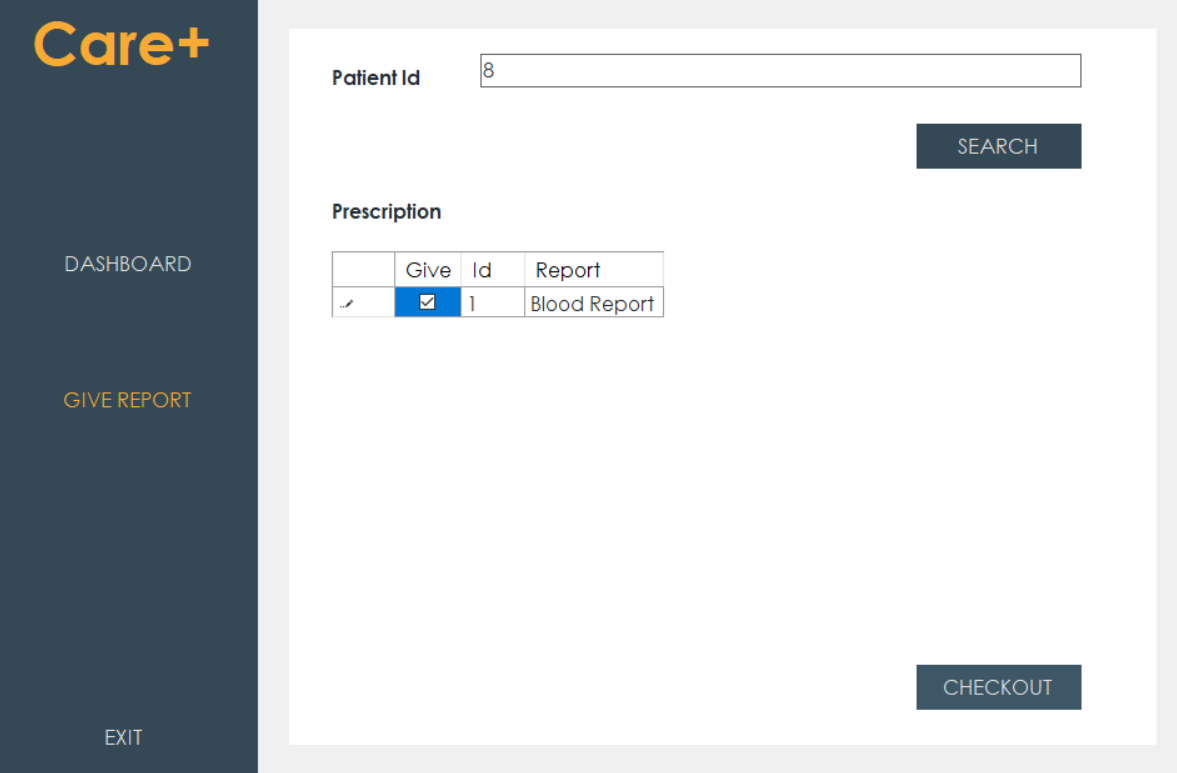
* 1. Doctor: Edit Case & search and add prescription



* 1. Pharmacist: Mark Prescription



* 1. Lab Technician: Mark Report



# **Conclusion**

The functionality implemented in the system was done after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are:

* User Registration
* User Login
* View Dashboard
* User Logout

Doctor:

* View Open Cases
* View Closed Cases
* View Patient History
* Edit Case
* Add Prescription
* Delete Prescription
* Update Prescription

Pharmacist:

* View Patient Prescription
* Mark Medicines

Lab Technician:

* View Report Prescription
* Mark Reports

Patient:

* View self Cases

# **Limitation and Future Extension**

## **Limitation of the project**

The limitation of the project is the inability to identify users uniquely in a way that is easy to access instead of using ids.

## **Functionality which was not implemented**

* + - * Admin functionality to check authenticity of professionals.
      * Uploading pdf of report by lab technician.
      * Generation of bill by pharmacist.
      * Uploading documents for verification by professionals.

## **List possible future extension to your project**

* + - * Can add an admin to check the authenticity of professionals.
      * Bill generation by pharmacist and lab technician.
      * Uploading of reports by lab technicians.

# **Bibliography**

Books:

* Head First Design Patterns by Elisabeth Freeman and Kathy Sierra

Useful Links:

* For info on C#
  + [https://www.c-sharpcorner.com](https://www.c-sharpcorner.com/UploadFile/1e050f/search-record-in-datagridview-C-Sharp/)
* For debugging support
  + <https://www.codeproject.com>
* For medicinal details:
  + <https://www.beckershospitalreview.com/supply-chain/10-most-popular-prescription-drugs-for-2017.html>
  + <https://www.livemint.com/news/india/the-muddled-health-data-system-in-india-11578590179610.html>