**UG DEPARTMENT OF COMPUTER SCIENCE PEOPLES CO-OPERATIVE ARTS AND SCIENCE COLLEGE**

**MUNNAD, KASARAGOD-671541**



**A Project Report On**

**“YOUR CARE”**

### Submitted to Kannur University in the partial fulfillment of the requirement for the Award

**of Bachelor of Computer Science.**

**By**

**,**

**SREERAJ K**

**Sixth Semester BSc Computer Science**

**Under The Guidance of Mrs. Bindu K Nair**

### Assistant Professor in Computer Science UG Department of Computer Science

**Peoples Co-operative Arts and Science College Munnad, Kasaragod-671541**

**PEOPLES CO-OPERATIVE ARTS AND SCIENCE COLLEGE MUNNAD**

### Affiliated to Kannur University

**EMS Aksharagramam-Munnad (PO), Chengala (via), Kasaragod- 671541 (A Unit Of Kasaragod co-operative Educational Society Ltd)** [**www.peoplescollege.edu.in**](http://www.peoplescollege.edu.in/) **Ph:04994207100**

**CERTIFICATE**

This is to Certify that the Project work entitled **“YOUR CARE**” is an authenticated record of the project work carried out by **SREERAJ K (PM20CCSR05)** in partial fulfillment of the requirement for the award Degree in the field of Computer Science for Kannur University under my guidance and supervision during the academic year 2020- 2023.

#### Internal Guide Head of Department

**Submitted for the project valuation held on…………………………………………….**

#### Signature of Examiners: 1………………………………………

**2………………………………………**

**Place: Munnad Date :**

**ACKNOWLEDGEMENT**

I express my humble thanks to almighty, for the kind grace showed me to complete the project successful. I would like to acknowledge the assistance and contribution of those who aided in successful completion of this project.

I have pleasure to acknowledge my deep sense of gratitude to **Dr. C K Lockose** Principal Peoples Co-operative Arts And Science College Munnad, **Mrs.Bindu K Nair** Head of the Department and all other teachers for the whole or some guidance and support in completing our project.

I also thank **Ms. Aparna and Ms. Joshna** their support, advice and timely help for the success of this project. I want to thank my friends and others including my parents of their help and co-operation for the success of this project.

My love and gratitude go once again to all those who supported as for the success of my project.

**ABSTRACT**

Skin diseases are one of the most common medical conditions encountered in the medical field. They are often caused by a combination of genetic, environmental, and lifestyle factors. This project aims to explore the various skin diseases that can affect individuals, their causes, symptoms, and treatments. “YOUR CARE” is an android application developed for patients. If the user is affected by any skin disease, they just need to take a picture of the affected area and upload to the application. This application predict your disease. It implement using artificial intelligence. This application tell the user about the disease that they affected in detail. If the disease will seriously affect your health, it shows an options to make a consultation .The available doctor details will be display through this application .The users need to make an appointment via online at the available time and get treated. This system will notify the appointments that have been made to a doctor and can manage according to the time of consulting by the doctor .This application also provide a mechanism for communicate with the doctor. The users also notify about the appointment with the time and date. Apart from this, a user can give the general symptoms other than skin related and know about the disease. These all activities can be going to implement in this project using Artificial Intelligence Techniques.

The major modules are

* Admin
* Doctor
* Pharmacy
* User

**CONTENTS**

#### INTRODUCTION

[INTRODUCTION 08](#_bookmark0)

[PROJECT OVERVIEW 08](#_bookmark1)

[NEED FOR THE SYSTEM 09](#_bookmark2)

MODEL FOR DEVELOPMENT 09

1. SYSTEM ANALYSIS

[INTRODUCTION 11](#_bookmark3)

METHOD FOR ANALYSIS 11

[EXISTING SYSTEM 12](#_bookmark4)

[PROPOSED SYSTEM 12](#_bookmark5)

[REQUIREMENTS 13](#_bookmark6)

FUNCTIONAL DESCRIPTION 13

[FEASIBILITY STUDY 16](#_bookmark7)

[ACTOR IDENTIFICATION 17](#_bookmark8)

1. SYSTEM DESIGN

[INTRODUCTION 19](#_bookmark9)

[TABLE DESIGN 20](#_bookmark10)

[DFD (DataFlow Diagram). 29](#_bookmark11)

[INTERFACE DESIGN 38](#_bookmark12)

[INPUT INTERFACE 38](#_bookmark13)

OUTPUT INTERFACE 38

SOFTWARE DESCRIPTION 39

#### TESTING

TESTING AND EVALUATION 44

TESTING STATERGIES 47

TESTING TECHNIQUES 48

#### IMPLEMENTATION 50

#### SYSTEM MAINTENANCE 52

#### CONCLUSION 54

#### APPENDIX 56

#### BIBLIOGRAPHY 77

# INTRODUCTION

## INTRODUCTION

Welcome to the “YOUR CARE” application prediction and identification of skin disease using artificial intelligence is a python and android based application. This application is an online platform that allows users to book appointments with dermatologists, buy medicines and get detailed information about various skin diseases. The app also provides users with interactive tools such as a symptom checker and risk calculator to help them identify and manage their skin condition. The app is designed to be user-friendly and easy to use. Through our user-friendly app, we provide an easy-to-use platform that allows users to quickly and easily find information about various skin diseases and their treatments.

The “YOUR CARE” application was created with the goal of providing users with a comprehensive, accurate and up-to-date source of information about skin diseases. The app is constantly being updated with new information and features in order to provide users with the most accurate and up-to-date information available. We are committed to providing the most up-to-date information on the diagnosis, treatment, and prevention of skin diseases. Our project also aims to empower individuals to take an active role in their healthcare. We invite you to explore our application and learn more about skin diseases.

## PROJECT OVERVIEW

“**YOUR CARE**” is a android application. In this system, database is designed using MySQL. Logical coding is done using java script which contains main logic of overall system functionality. To design web pages CSS is used for different styles. The system is covered under all validation checks for authentication. These student data is been analyzed based on various aspects like marks. System will be designed in such a way that it can be later updated to accept different format. The major modules are

* Admin
* Doctor
* Pharmacy
* User

## NEED FOR THE SYSTEM

The major objectives of the computerization in any field are to make man’s job easier. This application tell the user about the disease that they affected in detail. We identify drawback of the existing system and overcome them through the proposed system. The objectives of the software are

* + - **User friendly**
    - **Highly flexible**
    - **Time saving**
    - **convenience**

## MODEL FOR DEVELOPMENT

Waterfall Model is used to develop this application with object oriented concepts. The waterfall model is another iterative model that has been proposed. As the name suggests, the activities in this model can be organized like a waterfall that has many cycles. The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases.

There are five phases in waterfall model, they are:

* Requirement analysis and specification
* Design
* Implementation and unit testing
* Integration and system testing
* Operation and maintenance.

# SYSTEM ANALYSIS

## INTRODUCTION

System design provides an understanding of the procedural details, necessary implementing the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system and has a major impact on the testing and implementing phases. System design consists of three major steps.

* + - Drawing of the expanded system data flow charts to identify processing function required.
    - The allocation of the equipment and the website to be used.
    - The identification of the test requirements for the system.

## METHOD OF ANALYSIS

The first step of designing a system is to identify the drawbacks in the existing systems. The working of the existing system is to examine and study for this purpose. One must know what information is to be gathered, where to find it and how to make it use. Data are gathered and checked for completeness and accuracy. Analysis of data involve identification of the component of the system and their inter relationship and identify the strength and weakness of the system are part of their process. The tools for gathering information are,

* Interviews
* Phone conversation

## EXISTING SYSTEM

In existing system we need to make an appointment with doctor first. First the patient needs to call directly to the hospital for make an appointment. The administrator asks the patient details and tells them about the consulting time, token number and mark the appointment on the register. Doctor Analyses diseases using clinical screening and prescribe medicine in a white paper. There is no automation available. For other diseases patient, patients first share symptoms to doctor. Then doctor checks the patient. Prescriptions are made using doctor’s knowledge. If the doctor can’t diagnose the disease he will write lab notes. After getting lab results only doctor prescribe medicines. The lab tests are much expensive. The results may get late often in some times so the patient have to wait for the correct identification

## PROPOSED SYSTEM

The proposed system is a model based application. If the user is affected by any skin disease, they just need to take a picture of the affected area and upload to the application. The application tell the user about the disease that they affected in detail. If the disease will seriously affect your health, it shows an options to make a consultation .The available doctor’s and their consulting time will be displayed on their profile. The users need to make an appointment via online at the available time and get treated. Apart from this, a user can give the general symptoms other than skin related and know about the disease. If the user wants to consult a doctor for this type of disease, can also make appointments via this application.

Proposed system advantages:-

1. Less time consuming.
2. It is easy to use.
3. It can save time.
4. It executed properly

## 

## SOFTWARE & HARDWARE REQUIREMENTS

#### SOFTWARE REQUIREMENTS

Front End: HTML,CSS, JS

Android: Android Studio

Back End: PYTHON(Flask) Server: WAMP

Database: MYSQL

Operating System: Windows8 or above

#### HARDWARE REQUIREMENTS

Processor : Intelcore2.0Hz Memory Size : 4GB RAM Mouse : Compatible Mouse Keyboard : Normal Keyboard Monitor : Standard Monitor

Internet Connection : Any Internet Connection

## FUNCTIONAL DESCRIPTION

The project is mainly divided into 4 modules. They are

* Admin
* Doctor
* Pharmacy
* User
* **Admin** module consists of the following,

**Login :** Admin needs to login in order to avail their respective service

**Verify Doctor:** Admin can approve or reject doctor.

**Verify Pharmacy:** Admin can approve or reject pharmacy.

**View Doctor:** Admin can view doctor

**View Pharmacy:** Admin can view pharmacy

**View Appointment:** Admin can view appointment

**View User:** Admin can view user

**Add Medicine:** Admin can add medicine.

**View Medicine:** Admin can view medicine.

**Add Disease:** Admin can add disease.

**View Disease:** Admin can view disease.

* **Doctor** module consist of the following,

**Registration:** Doctor needs to register.

**Login:** Doctor needs to login in order to avail their respective service.

**View Disease:** Doctor can view diseases.

**Add Schedule:** To Add schedule.

**View Schedule Date:** To view scheduled date.

**View Appointment :** To view all appointments.

**View Today Appointment:** To view today appointments.

**Chat With User:** Doctor can chat with user.

**View Profile:** To view profile.

**View Rating:** To view rating.

* **Pharmacy** module consist of the following,

**Registration:** Pharmacy needs to register.

**Login:** Pharmacy needs to login with the username and password.

**View Medicine:** To view medicine.

**Request Medicine:** Accept the request medicine.

**History:** To view requested medicine.

* **User** module consist of the following,

**Registration:** User can use the application user must register first.

**Login:** User needs to login with the username and password .

**View Profile:** User can view their profile.

**Upload Disease Affected Area**: User can upload their disease.

**View Doctor:** User can view doctor profile.

**Add Rating:** User can rate the doctor.

**Chat With Doctor:** User can chat with doctor.

**Book Appointment:** User can book appointment.

**Add Rating:** User can rate the doctor.

**View Appointment:** User can view appointment.

**View Disease:** User can view disease.

**View Prescription:** User can view prescription.

**Add Rating:** User can rate the doctor.

**Send Request To Medicine:** User can send request for pharmacy medicines.

**View Requested Medicine:** User can view requested medicine.

## FEASIBILITY STUDY

Here are three feasibility studies needed for the systems. They are

* Behavioral Feasibility Study
* Technical Feasibility study
* Economic Feasibility Study

#### ECONOMICAL FEASIBILITY STUDY

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. It is more commonly known as cost benefit analysis , the procedure to determine the benefits and saving that are expected from a candidate system and compare them with costs. If the benefits out weight cost then a decision is made to design and implement the system. Otherwise make alteration in the proposed system.

#### BEHAVIOURAL FEASIBILITY STUDY

People are inherently resistant to change and computers have been known to facilitate change .An estimate should be made about there action of the user staff towards the development of a computerized system. Computer installation have something to do with turnover, transfers and changes in jobs status. The introduction of a candidate system requires special effort to educate , and train the staff. The system was found to be technically, economically and behaviorally feasible. The system was developed user friendly, needless training and improves efficiency. Disregarding the initial expenses ,the candidates system was assessed to be feasible in all ways.

#### TECHNICAL FEASIBILITY STUDY

The assessments of technical feasibility centers on the existing system and to what extent it can support the proposed addition. This was based on an outline design of system requirements in turns of inputs ,files ,programs, procedures and staff. It involves financial considerations to accommodate technical enhancements.

## ACTOR IDENTIFICATION

#### ACTOR

An actor is someone or something that interact with the system. An actor is he/she who uses the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actor of the system.

#### IDENTIFICATION OF ACTOR

We can identify actors through a list of questionnaires. Who will use the functionality of the system:-Administrator

Who will lead support from the system and do their daily tasks:-clients Who will lead maintain and administrate the system:-Administrator With which other this system need to interact:-database

Which hardware device does the system need to handle:-No Who was interface in the result to produce by the system:-Client The actor of the system is administrator

# SYSTEM DESIGN

## INTRODUCTION

Systemdesignprovideanunderstandingoftheproceduraldetails,necessary Implementing of the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system and has a major impaction the testing and implementation phase.

System design consists of three major steps.

#### Drawing of the expanded system dataflow charts to identify all the processing functions required.

**The allocation of the equipment and the software to be used. The identification of the test requirements for the system.**

Characteristics of design

#### A design should exhibit a hierarchical organization that make intelligent use of control among component of the software.

**A design should be modular that is, the software should be logical.**

**A design should contain distinct and separable representation of data and procedure.**

**A design should lead to interface that reduce the complexity of the connection between modules and with the external environment.**

## TABLE DESIGN

DB design is required to manage large bodies of information. The management of data involves both the definition of the structure of storage of information and provisions of mechanism for the manipulation of information. For developing an efficient database certain condition shave to be fulfilled such as:

#### Control redundancy Ease of use

* + - **Data independence Accuracy and integrity**

There are five major steps in design process:

#### Identify table and relationship

* + - **Identify the data that is needed for each table and relationship**
    - **Resolve the relationship**
    - **Verify the design**
    - **Implement the design**

**Table name: login Purpose: Login process Primary key: login\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| Login\_id | int | Primary key | Login id |
| username | varchar(100) | Not null | Username |
| password | varchar(100) | Not null | Password |
| usertype | varchar(30) | Not null | Type of user |

**Table name: pharmacymedicine**

**Purpose: To store pharmacy medicne**

**Primary key: pmid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| pmid | int | Primary key | Pharmacy medicine id |
| pid | int | Foreign key | Prescription id |
| medicine\_id | int | Foreign key | Medicine id |
| stock | int | Not null | Stock |

**Table name: doctor**

**Purpose: Doctor Registration**

**Primary key: doctor\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| doctor**\_**id | int | Primary key | Doctor id |
| doctor\_name | varchar(100) | Not null | Doctor name |
| place | varchar(100) | Not null | Place |
| post | varchar(100) | Not null | Post |
| district | varchar(100) | Not null | District |
| email | varchar(100) | Not null | Email |
| mobilenumber | varchar(100) | Not null | Mobile Number |
| gender | varchar(100) | Not null | Gender |
| specialization | varchar(100) | Not null | Specialization |
| qualification | varchar(100) | Not null | Qualification |
| upload\_certificate | varchar(100) | Not null | Upload certificate |
| photo | varchar(100) | Not null | Photo |

**Table name: pharmacy**

**Purpose :To Pharmacy details**

**Primary key: pid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Descripti on** |
| pid | int | Primary key | pharmacy id |
| pharmacy\_name | varchar(200) | Not null | Pharmacy Name |
| email | varchar(200) | Not null | Email |
| phonenumber | bigint(20) | Not null | Phone number |
| latitude | varchar(200) | Not null | latitude |
| longitude | varchar(200) | Not null | longitude |

**Table name: chat**

**Purpose: To store chat details Primary key: chat\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| chat\_id | int | Primary key | Chat id |
| from\_id | int | Not null | From id |
| to\_id | int | Not null | To id |
| date | varchar(45) | Not null | Date |
| message | varchar(100) | Not null | Message |

**Table name: disease**

**Purpose: To store disease details**

**Primary key: disease\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| disease\_id | int | Primary key | Disease id |
| disease\_name | varchar(100) | Not null | Disease name |
| symptoms | varchar(100) | Not null | Symptoms |
| photo | varchar(100) | Not null | Photo |
| details | varchar(100) | Not null | Details |

**Table name: medicine**

**Purpose: To store medicine details Primary key: mid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| mid | int | Primary key | Medicine id |
| medicine\_name | varchar(200) | Not null | Medicine Name |
| medicine\_image | varchar(200) | Not null | Medicine image |
| medicine\_details | mediumtext | Not null | Medicine details |
| price | varchar(200) | Not null | Price |

**Table name: medicinebook**

**Purpose :To store booked medicine details**

**Primary key: mbid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| mbid | int | Primary key | Medicine book id |
| pharmacy\_id | int | Foreign key | Pharmacy id |
| date | varchar(20) | Not null | Date |
| status | varchar(200) | Not null | Status |
| prescription\_id | int | Foreign key | Prescription id |

**Table name: appointment Purpose: To store appointment Primary key: appointment\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Descripti on** |
| appointment\_id | int | Primary key | Appointment id |
| doctor\_id | int | Foreign key | Doctor id |
| patient\_id | int | Foreign key | Patient id |
| date | varchar(20) | Not null | Date |
| status | varchar(100) | Not null | Status |

**Table name: photo\_upload**

**Purpose: To store photo details**

**Primary key: photo\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| photo\_id | int | Primary key | Photo id |
| user\_id | int | Foreign key | User id |
| date | varchar(50) | Not null | Date |
| photo | varchar(100) | Not null | Photo |
| details | varchar(100) | Not null | Details |

**Table name: prescription**

**Purpose: To store prescription details**

**Primary key: pid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| pid | int | Primary key | prescription id |
| appointment\_id | int | Foreign key | Appointment id |
| medicine\_id | int | Foreign key | Medicine id |
| quantity | int | Not null | Quantity |
| details | varchar(200) | Not null | Details |

**Table name: rating**

**Purpose: To store rating details**

**Primary key: rating\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| rating\_id | int | Primary key | Rating id |
| user\_id | int | Foreign key | User id |
| doctor\_id | int | Foreign key | Doctor id |
| date | varchar(50) | Not null | Date |
| rate | int | Not null | Rate |

**Table name: schedule**

**Purpose: To store schedule details**

**Primary key: schedule\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| schedule\_id | int | Primary key | Schedule id |
| from\_time | time | Not null | From time |
| to\_time | time | Not null | To time |
| date | varchar(20) | Not null | Date |
| doctor\_id | int | Foreign key | Doctor id |

**Table name: user**

**Purpose: User Registration**

**Primary key: user\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** | **Description** |
| user**\_**id | int | Primary key | User id |
| name | varchar(100) | Not null | User name |
| place | varchar(100) | Not null | Place |
| age | int | Not null | Age |
| post | varchar(100) | Not null | Post |
| district | varchar(100) | Not null | District |
| email | varchar(100) | Not null | Email |
| mobile\_number | varchar(100) | Not null | Mobile Number |

## DFD-DATA FLOW DIAGRAM

A graphical representation is used to describe and analysis the movement of data through a system manual or automated including the processes, storing of data and delays in the system. Data flow diagrams are the central tool and the basis from which other components are developed.

The transformation data, from input to output through process may be described logically and independently of the physical components associated with the system.

Theyaretermedlogicaldataflowdiagramsshowingtheactualimplementationandthemovement of data between people, department and workstation. DFD is one of the most important modeling tool is used in system design. DFD shows the flow of data through different process in the system.

#### PURPOSE

The purpose of the system design is to create architecture for the evolving implementation and to establishing the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have reasonably completed model the behaviour of the system. It is important to avoid premature designs, where in develop designs for analysis reaches closer. It is important to avoid delayed where in the organization crashes while trying to complete an un achievable analysis model.

Throughout our project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In my opinion "efficient design of dataflow and context flow diagram helps to design the system successfully without much major flows within the scheduled time ".This is the most complicated part in the project .In the designing process , my project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as a way of expressing system requirements in graphical form. A data flow diagram is also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformations thatwillbecomeprogramsinsystemdesign.Itfunctionalitydecomposestherequirementspecificatio n down to the lowest level. Data Flow Diagram depicts the information flow, transformation flow and the transformations that are applied as data move from input to output. Thus DFD describes what data flows rather than how they are processed.

Data Flow Diagram is quite effective: especially when the required design is unclear and the user and analyst need a notational language for communication. It is one of the most important tools used during system analysis. It is used to model the system components such as the system process, any external entities that interact with the system and information flows in the system.

Data Flow Diagrams are made up of a number of symbols, which represents system components .Data Flow modeling method users four kind so of symbols, which used to represent four kinds of system components. There are

1. Process.
2. Data stores.
3. Data flows.
4. External entity.

#### PROCESS

Process how the work of the system. Each process has one or more data inputs and procedure one or more data outputs. Processes are represented by rounded rectangles in Data Flow Diagram. Each process has a unique name and number. This name and number appears inside the rectangle that represents the process in a Data Flow Diagram.

#### DATA STORES

A data source is are repository of data. Processes can enter data, into as to re or retrieve the data from the data store. Each data has a unique name.

#### DATA FLOWS

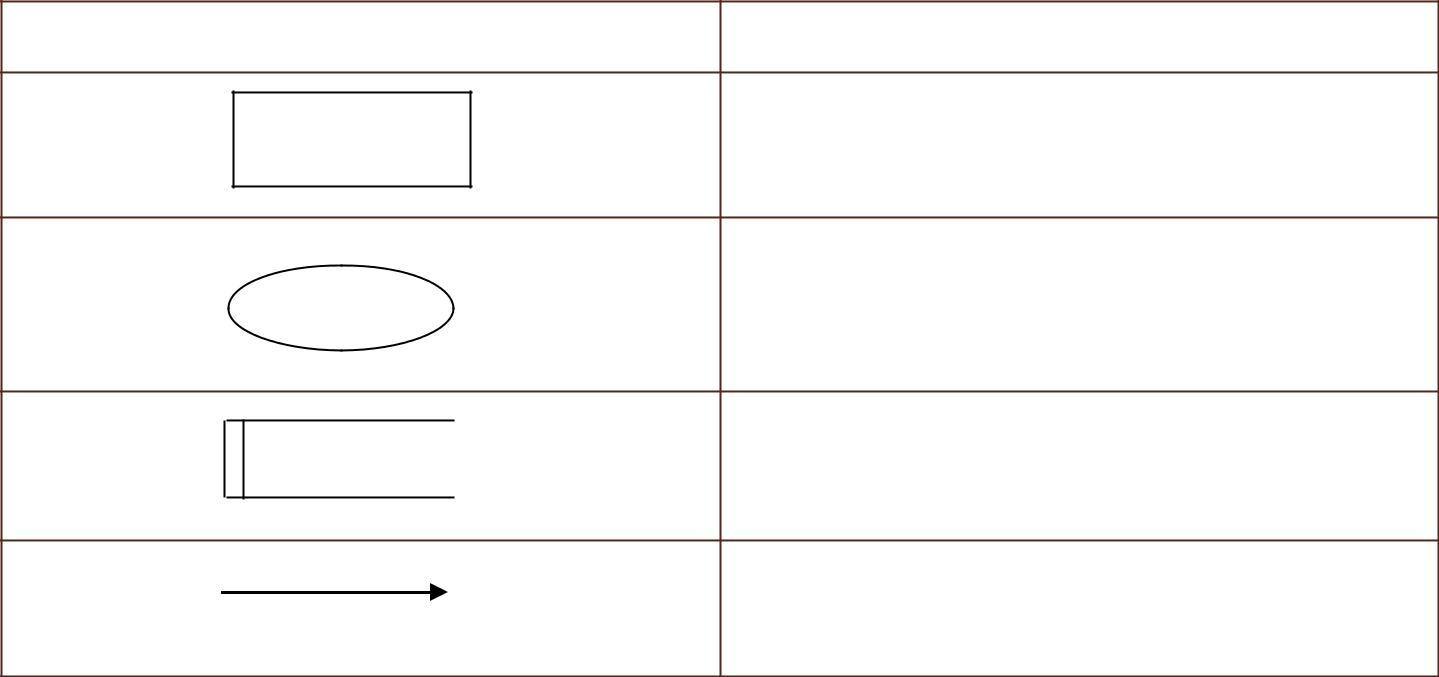
A data flows how the passage of data in the system and is represented by lines joining system components. An arrow indicates the direction of flow and the line is labeled by name of the data flow.

#### EXTERNAL ENTITY

External entities are outside the system but they either supply input data into the system or use other systems output. They are entities on which the designer has control. They may bean organizations customer or other bodies with which the system interacts. External entities that use the system data are sometimes called sinks. These are represented by rectangles in the Data Flow Diagram. Four basics symbols are used to construct data flow diagram .They are symbols that represented by rectangles in the Data Flow Diagram

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, data transformations and data storage. The points at which data are transformed by enclosed figures, usually circles, which are called nodes.

Basic data flow diagram symbols are



Source or destination of system data

Processes

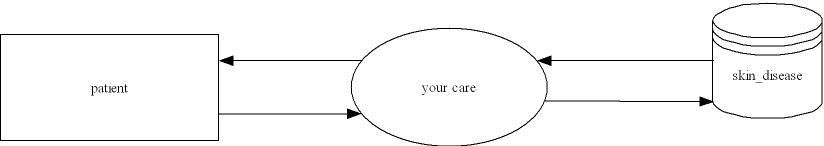
A Database/table

Flow of data

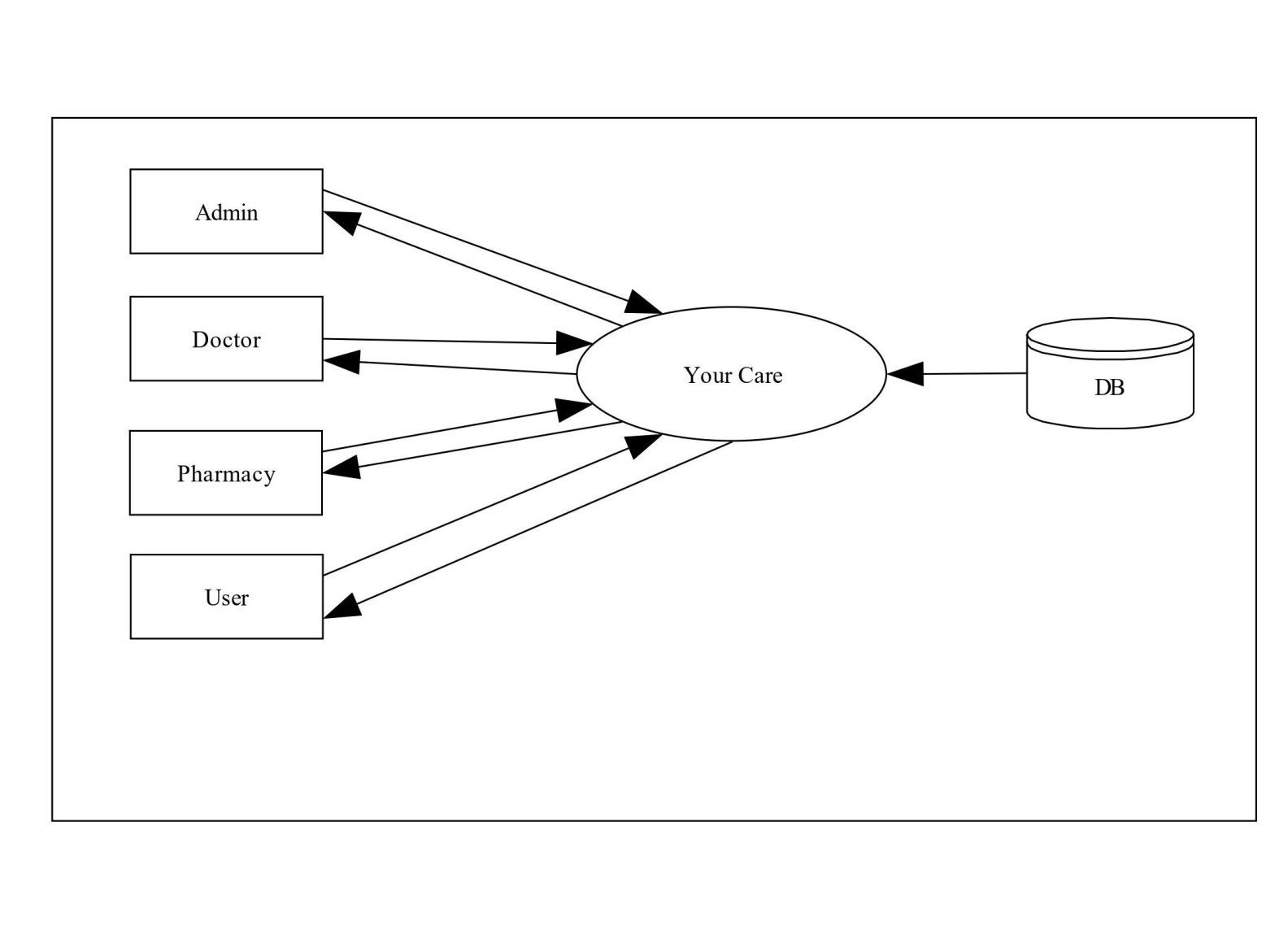
Four steps are commonly used to construct at DFD.

1. Process should be named and numbered for easy reference. Each name should be representative of the process.
2. The direction of flow is from to up to bottom and left to right.
3. When a process is exploded into lower level details they are numbered.
4. The names of data stores, sources and destinations are written in capital letters.

#### Level-0

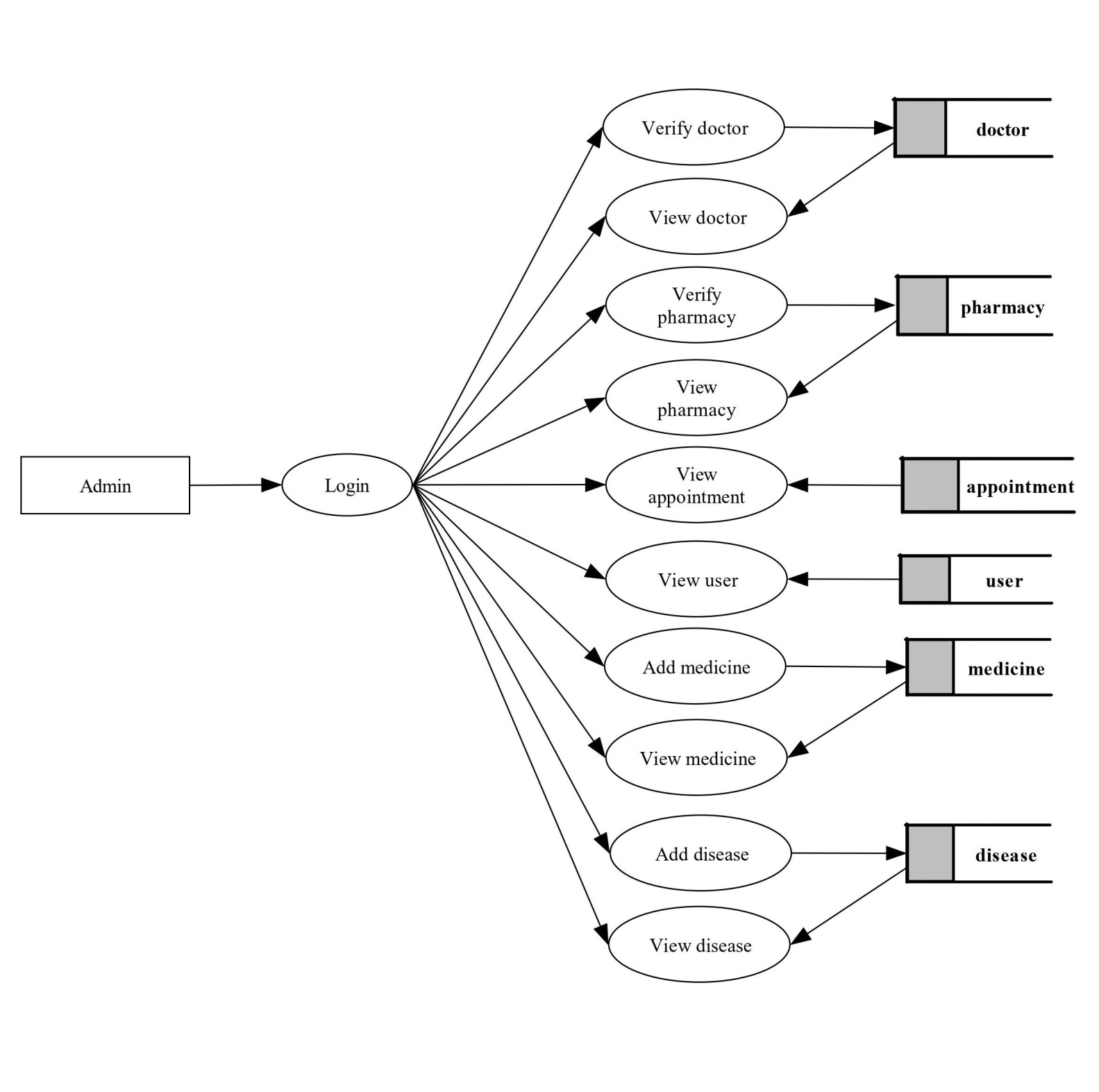
****

#### Level-1

****

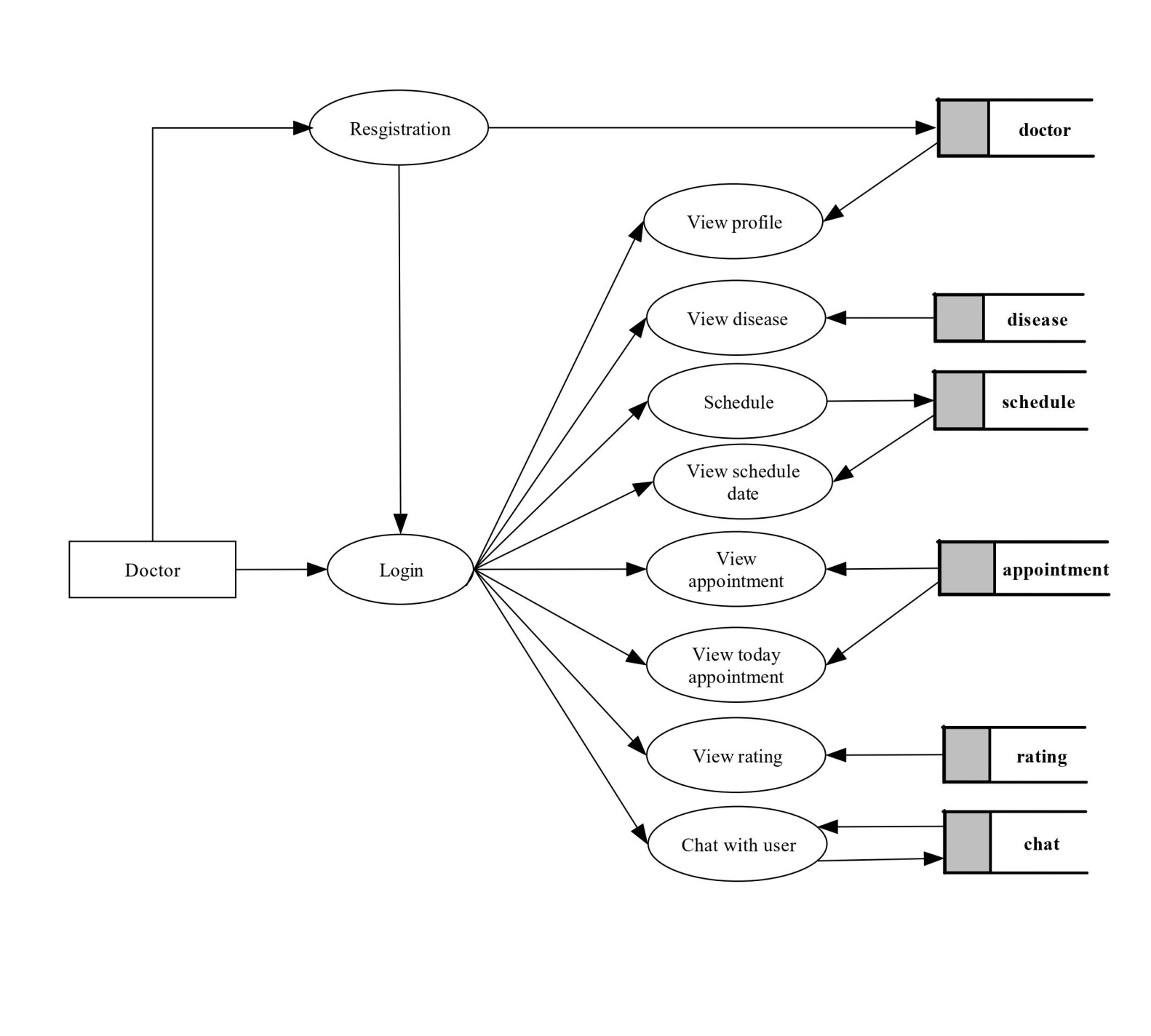
**Level-2.1**

**Admin:**

****

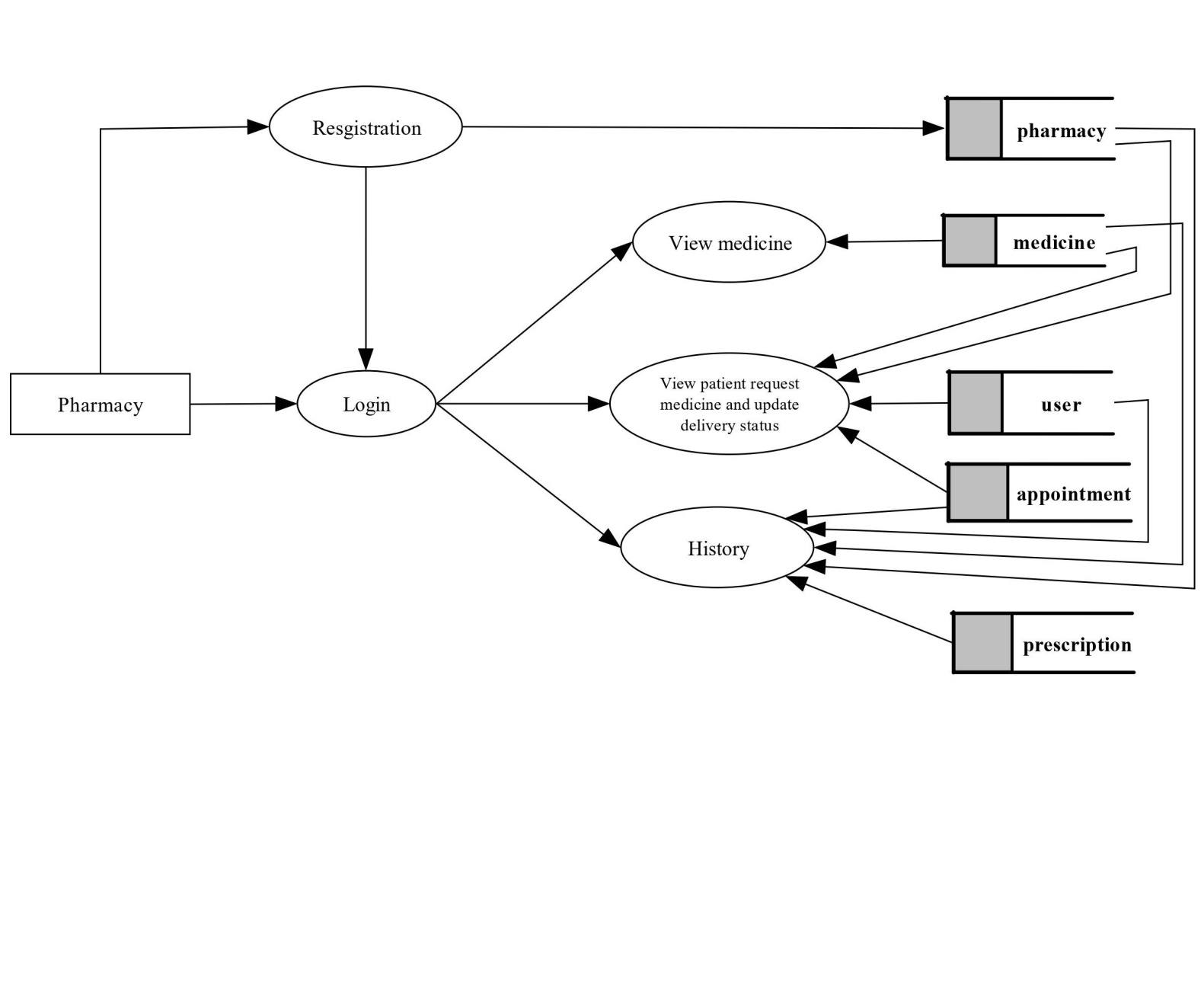
**Level-2.1**

**Doctor:**



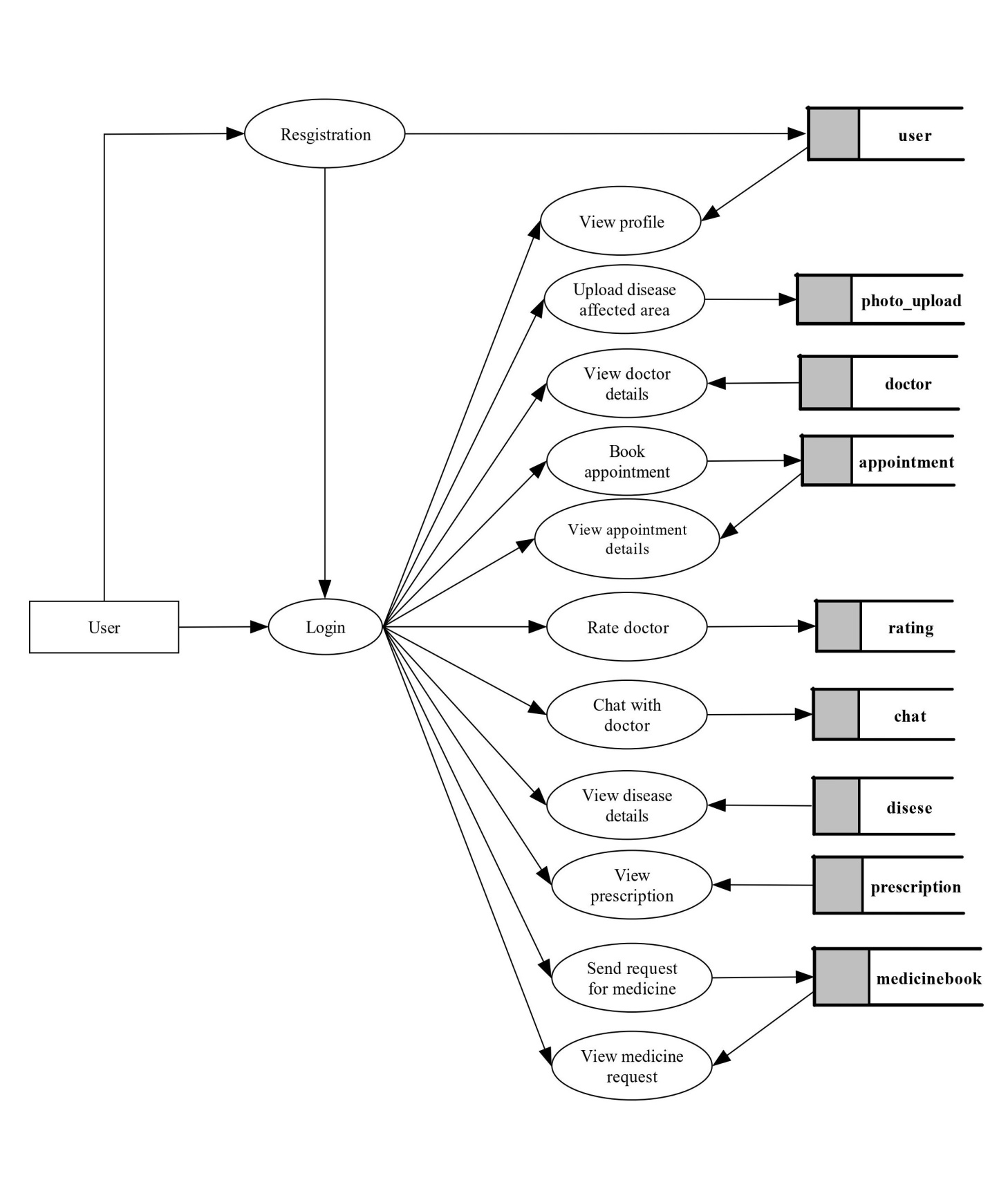
**Level-2.1**

**Pharmacy:**

****

**Level-2.1**

**User:**



## INTERFACE DESIGN

**INPUT INTERFACE**

Input design is a part of overall system design, which requires very careful attention. If data going into the system is correct, then the processing and output will magnify these errors. Thus the designer has a number of clear objectives in the different stages of input design.

* + - * To produce a cost effective method of input.
      * To achieve the highest possible level of accuracy.
      * To ensure that input is acceptable to and understand by the user.

#### OUTPUT INTERFACE

At the beginning of the output design various types of outputs such as external, internal, operational and interactive and turn around are defined. Then the format, content, location, frequency, volume and sequence of the outputs are specified. The content of the output must be defined in detail. The system analysis has two specific at this stage.

To interpret and communicate the results of the computer part of a system to the users in a form, which they can understand, and which meets their requirements.

To communicate the output design specifications to programmers in a way in which it is unambiguous, comprehensive and capable of being translated into a programming language.

## SOFTWARE DESCRIPTION

#### WATERFALL MODEL

WaterfallModelisusedtodevelopthisapplicationwithobjectorientedconcepts.Thewaterfall model is another iterative model that has been proposed. As the name suggests, the activities n this model can be organized like a waterfall that has many cycle.

The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases.

There are five phases in waterfall model, they are:

* Requirement analysis and specification
* Design
* Implementation and unit testing
* Integration and system testing
* Operation and maintenance

#### BOOTSTRAP 4

Bootstrap is a free and open-source CSS framework directed at responsive, mobile first frontend web development. It contains CSS and (optionally) java Script –based design templates of or typography, forms, buttons, navigation and other interface components.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to Twitter developer Mark Otto:

A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company.

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011.[5] It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

#### PYTHON

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often describe data "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0,released in2000,introduced features like list comprehension sand a garbage collection system capable of collecting reference cycles.Python3.0,released in 2008,was a major

Revision of the language that is not completely backward-compatible, and much Python 2 code does not run un modified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[30][31]With Python2'send-of-life,onlyPython3.5.x[32] and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains C Python, an open source [33] reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C Python development.

#### FLASK

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries.[3] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.[4] Applications that use the Flask framework include Interest and LinkedIn.

#### MYSQL

MYSQL is the most popular Open Source database, is developed, distributed and supported by MYSQL LAB. MYSQL is a database management system. MYSQL is a relational database management system. The SQL part of “MYSQL” stands for “Structured Query Language” the most common standardized language used to access databases. MYSQL software is Open Source. Open source means that it is possible for anyone to use and modify it. Anybody can download the MYSQL software from the Internet and use it without paying anything. The MYSQL database server is very fast, reliable and easy to use. It was originally developed to handle large database much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Though under constant development, MYSQL Server today offers a rich and useful set of functions. Its connectivity, speed and security make MYSQL Server highly suited for accessing databases on the Internet.

# TESTING

## TESTING AND EVALUATION

Testing is a process of executing program with the intent of finding an error. Software testing is a critical element of software quality assurance and the ultimate review or specification , design and coding. Testing includes verification of the basic logic of each program and verification that the entire system works properly. Testing demonstrates that software functions appear to be working according to specification. In addition, data collected as testing conducted provide a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process.

Testing begins at the module level and works towards the integration of the entire computer based system testing and debugging are different activities, during the development and modification of the software. There are two types of verifications but any testing includes debugging strategy for software testing must accommodate low level tests that are necessary to verify that a small source code segment has been currently implemented as well as high level tests that validate major system function, against customer requirements. No testing is complete with our verification and validation part.

The goals of verification and validation activities are to access and improve the quality of work products generated, they are lifecycle verification and formal verification. Life Cycle verification is the process of determining the degree to which the products of the given phase of the development cycle fulfill the specification established during the prior process. Formal verification is the rigorous mathematical demonstration that source code confirm to its specifications. Validation is a process of evaluating software at the end of the software development process to determine completion with the requirements

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The primary objectives, when we test software are the following:

Testing is a process of exceeding with the intent of finding an error.

A good test is one that has a high probability of finding an un discovered error.

A successful test is on un cover un discovered errors.

Automated testing is per formed at four different levels:

* Unit testing.
* Integration testing.
* System testing.
* Acceptance testing.

#### UNIT TESTING

Unit testing comprises the set of tests performed by an individual programmer prior to the integration of the system. Testing removes residual bugs and improves the reliability of the system. Testing allows the developer to find out the design faults f any, and enable correction if needed. Exhaustive unit testing has to be carried out to ensure the validity of the data. In order to successfully test the entire package, unit test is carried out. Each module was tested as when it was developed. Thus it proved easier to conduct minute testing operation and correct them then and there.

#### INTEGRATING TESTING

Bottom-up integration is the traditional strategy used to integrate the component of software system into a functional whole. Bottom - Up integration consists of unit testing, followed by subsystem testing and followed by testing of entire system. Unit testing has the goal of discovering the errors in the individual parts of the system

Parts are tested in isolation from one another in an artificial environment known as “Test Harness”, which consists of driver of programs and data necessary to exercise the modules unit testing should be as exhaustive as possible to entire that each representative case handled

by each module has been tested. Unit testing is eased by a system structure that is composed of small loosely coupled modules.

A subsystem consists of several modules that communicate with each other through well- defined interfaces. Normally, subsystem implements a major segment of the total system the primary purpose of the subsystem is to verify operation of the interfaces between two modules in the subsystem. Both control and data interfaces must be rested. Large software system may require several levels of subsystems. Lower level subsystems are successively combined to form higher level subsystems. In most software systems, exhaustive testing of subsystem capabilities is not feasible due to the combination complexity of the module interface. Therefore, test cases must be carefully chosen to exercise the interface in the desired, manner.

#### ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance tests andstresstestsinordertodemonstratethattheimplementedsystemsatisfiesitsrequirements.Itis not to unusual for two sets of acceptance test to be run, those developed by the quality group and those developed by the customer. In addition to the functional and performance tests, stress tests are performed to determine the limitation of the system. For example, a compiler might be tested to determine the effect of the symbol table overview, or real-time system might be tested to determine the effect of simultaneous arrival of numerous high priorities interrupts.

#### SYSTEM TESTING

System testing is similar to integration testing, but instead of integrating modules into programs for testing, programs in a system. System testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably excepted by the customer.

## TESTING STRATEGIES

A strategy for software testing integrated software test case design method in to a well- planned series of steps that results in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when the series steps are planned and then undertaken, and how much effort, time and resources will be required. Therefore any testing strategy must incorporate test planning, test case, design, test execution, resultant data collection and evaluation. A software testing strategy should be flexible enough to promote reasonable planning and management tracking as the project progress. The general characteristics of software testing strategies are

* + - Testing begins at the component level and works “outward” toward the integration of the entire computer system.
    - Different testing techniques are appropriate a different point in time.

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and set of mile stones for the manager. Because the step on the test strategy occurs at a time when deadline pressure begins to rise, progress must be measurable and problem must surface as early as possible.

The software teams approach to testing is defining a plan that describes an overall strategy and a procedure that defines specific testing steps and test that will be conducted. In the Proposed system, if the administrator makes any attempt to login to the application without entering his password, then the system will not allow the user to login to the application.

## TESTING TECHNIQUES

The various testing techniques are given below.

#### WHITEBOX TESTING

White-box testing also called a glass box testing, is a test case design method that goes to the control structure of a procedural design to derive test case. Using white-box testing methods, the software engineer can derive test cases that

* Guarantee that all independent paths within a module have been exercised atleast once.
* Exercised all logical decisions on their true and falses ideas.
* Execute all loops at their boundaries and within their optional bounds.
* Exercise internal data structure to ensure their validity.

White-box testing was successfully conducted on our system. All independent paths with in a module have been exercised on their true and false sides.

#### BLACK BOX TESTING

Black - box testing also called behavioural testing, focuses on the functional requirements of the software. It is complementary approach that is likely to uncover different classes of errors than white-box methods.

Black – Box testing attempts to find errors in the following categories.

* + Incorrect or missing functions.
  + Interface errors.
  + Errors on the data structure or external database access.
  + Behaviour or performance errors.
  + Initialization and termination errors.

# IMPLEMENTATION

## IMPLEMENTATION

The implementation phase is less reactive. It is primarily concerned with user training, site preparation and file conversion .It is also tests the users acceptance.

After the completion of project it is aimed to implement or install the system in the company. Prior to the implementation site preparation are made. It includes the assurance of hard and software’s that have already met. After the new system is implemented then checks for the readiness and accuracy of the system to access, update and retrieve data from new files. Once the programs become available, test data are read into the comport and processed against the files provided for testing. If successful, the programs are run with„ live data ‟.Thus the data‟ that the company want to keep in the new system are entered, processed and verified the output. Otherwise, a diagnostic procedure is used to locate and corrects errors in the program. Major activities in implementation stages are:

#### User training

While computer programs are being developed and tested, the analyst will start familiarizing users with procedures to be used in the new system. The data entry operators must be trained to enter data from the source documents and the correct errors.

#### Implementation procedure

Implementation phase is to last phase regarding any system development, before any system is implemented. Various tests are performed the system is reviewed thoroughly. The changeover selects is used to change the existing system to the new proposed system. In this phase, the system enters the operation and routine maintenance stage.

#### Operational documentation

This system is developed in such a way that the existing system facilities are enough for implementation. The hardware facilities are sufficient enough to implement the developed system.

# SYSTEM MAINTENANCE

## SYSTEM MAINTENANCE

Software Development has many phases. These phases include requirements Engineering, Architecting, design, implementation, testing, software deployment and maintenance. Maintenance is the last stage of software life cycle. After the product has been released, the maintenance phase keeps the software up to date with environment changes and changing user requirements. The earlier phases should be done so that the product is easily maintainable. The design phase should plan the structure in a way that can be easily altered. Similarly, the software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment implementation phase should create code that can easily read, understood, and changed. Maintenance can only happen efficiently if the earlier phases are done properly.

There are four major problems that can slow down the maintenance process, un structured code, maintenance programmers having insufficient knowledge of the system, documentation being absent, out of date or at best in sufficient and software maintenance phase relies on these problems being fixed earlier in the lifecycle.

Maintenance consists of four parts. Corrective maintenance with fixing bugs in the code. Adaptive maintenance deals with adapting the software to new environments. Perfective maintenance deals with updating the software according to changes in user requirements. Finally preventive maintenance deals with updating documentation and making the software more maintainable. All changes to the system can be characterized by these four types of maintenance. Corrective maintenance is „traditional maintenance‟ while the other types are considered as„ software evolution‟.

# CONCLUSION

**CONCLUSION**

“YOUR CARE” is an android application The application tell the user about the disease that they affected in detail. If the disease will seriously affect your health, it shows an options to make a consultation. The patients can login in this application by giving the attributes like name and password. Doctor can able to see the patients and make a consultation with her. Patients can view their disease details, doctors and appointments etc. YOUR CARE application is offering a maximum of stability, cost effectiveness and availability. it is important to be aware of skin diseases and the warning signs associated with them. Early detection is key to successful treatment and reducing the risk of complications. It is also important to practice good hygiene and skin care to reduce your risk of developing skin diseases. Regular visits to a dermatologist are also recommended to help monitor your skin health and catch any changes in your skin early on. Our goal is to provide users with the resources they need to better understand skin diseases and make informed health decisions. Thanks for joining us in this exciting project.

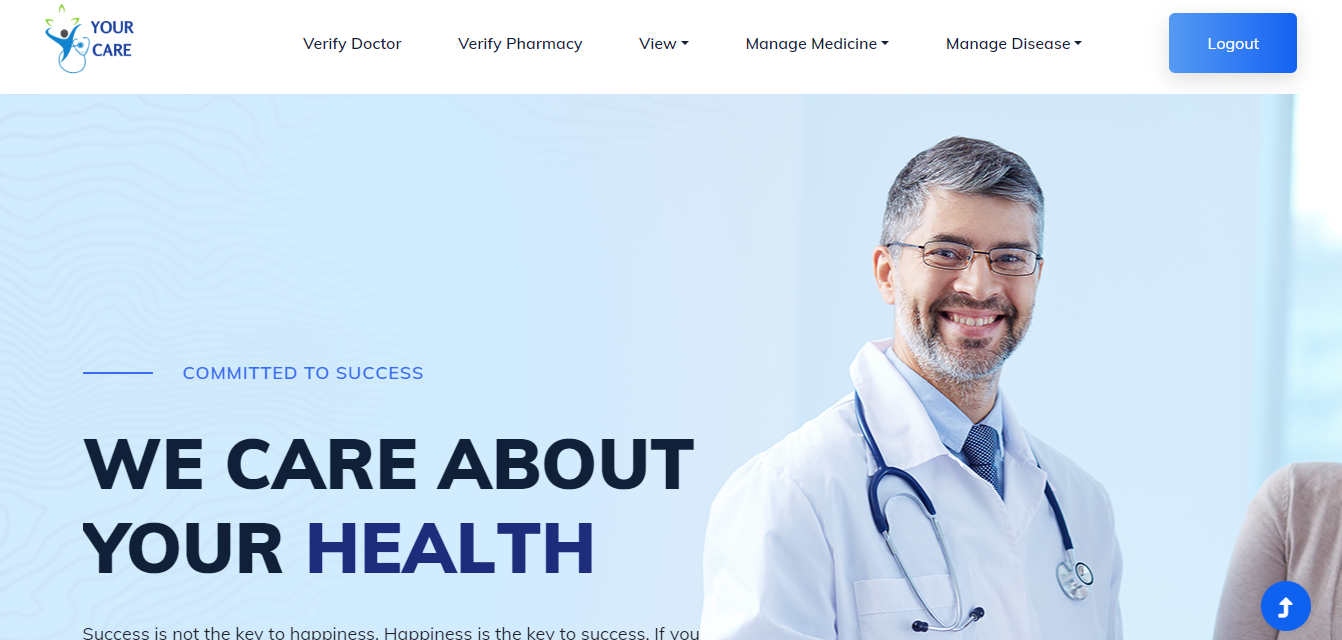
.

# APPPENDEX

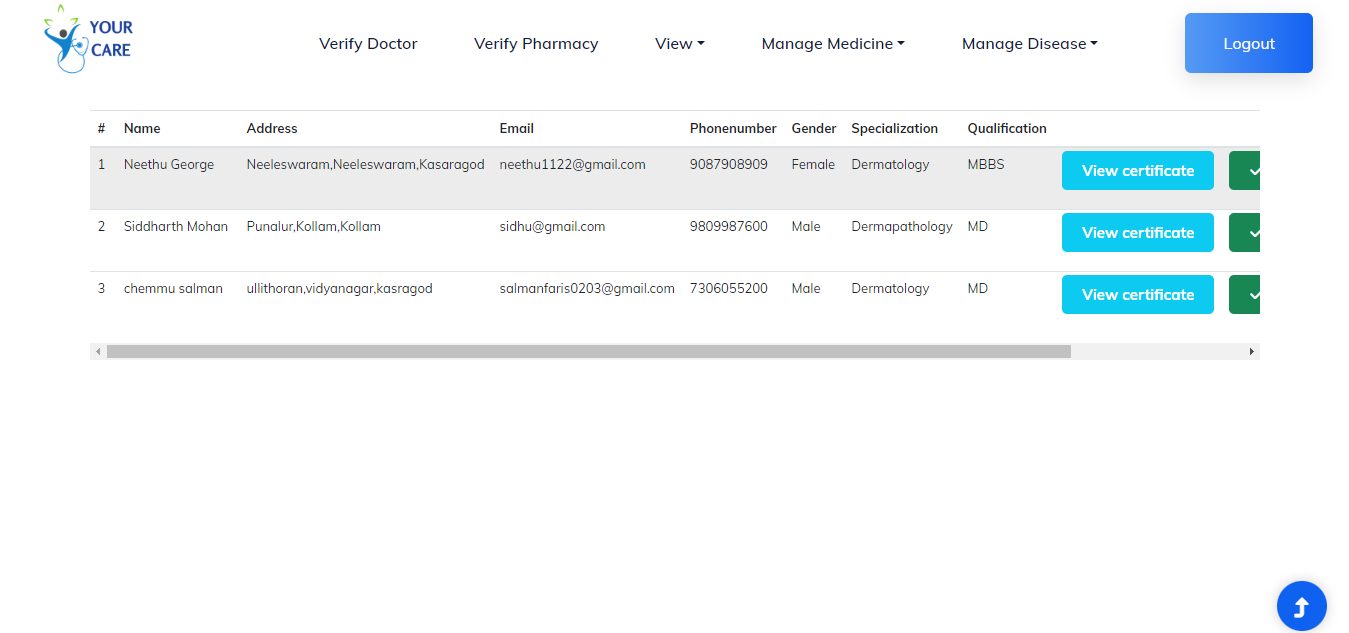
**Main Login Page**

### 

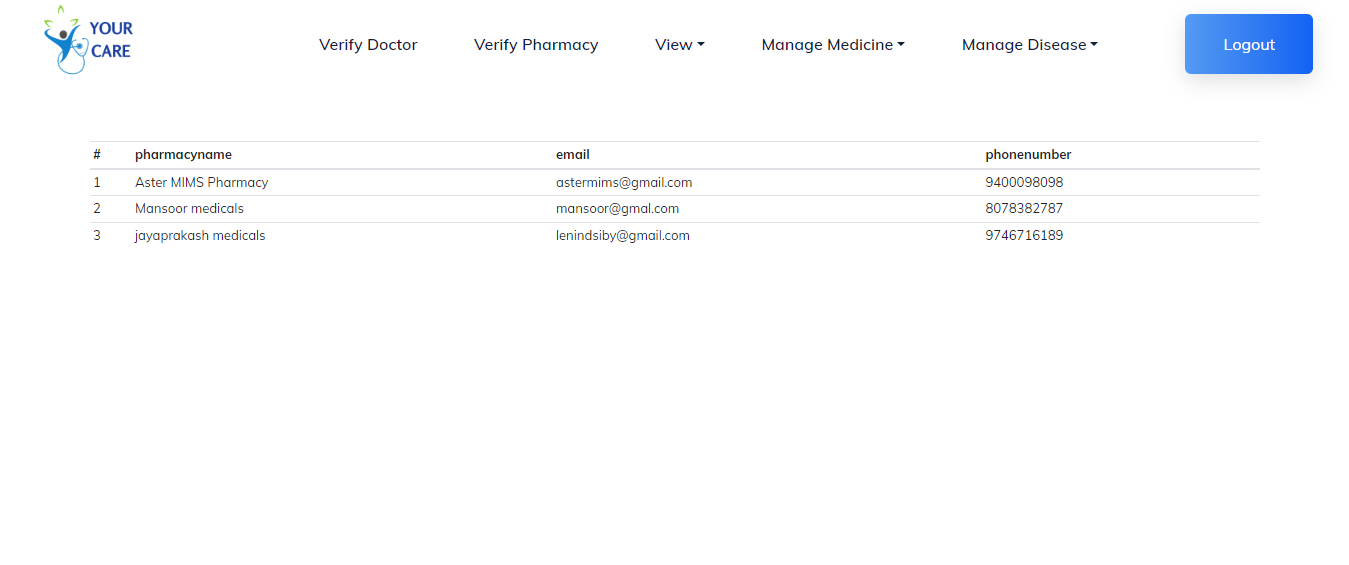
### Admin Home

****

**Admin Verify Doctor**

****

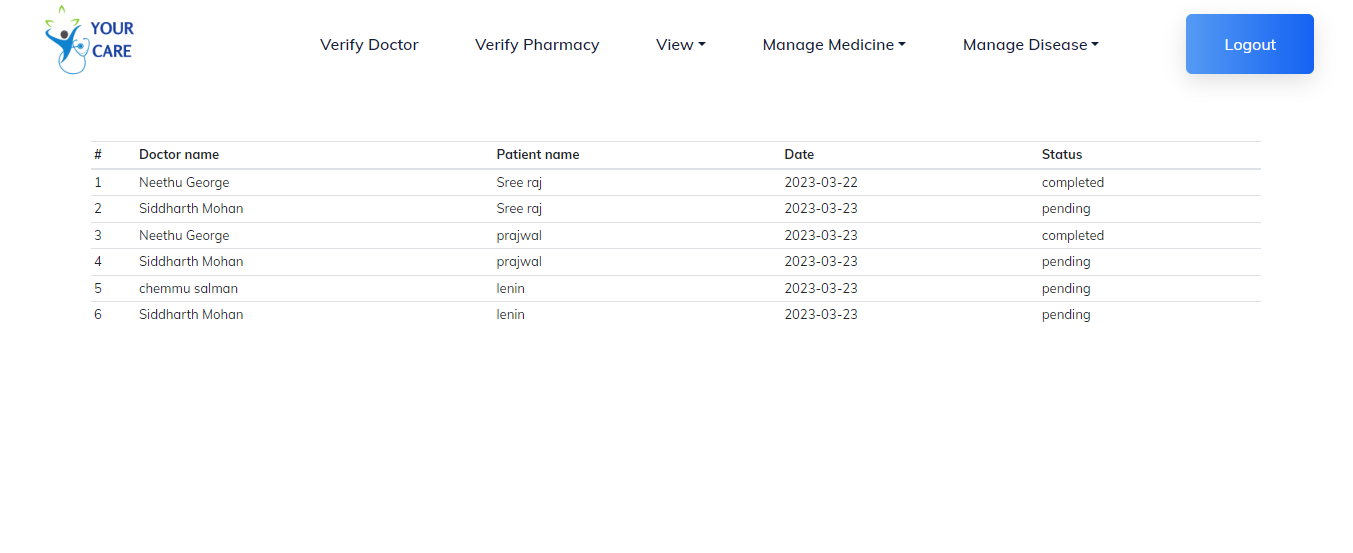
**Admin Verify Pharmacy**

****

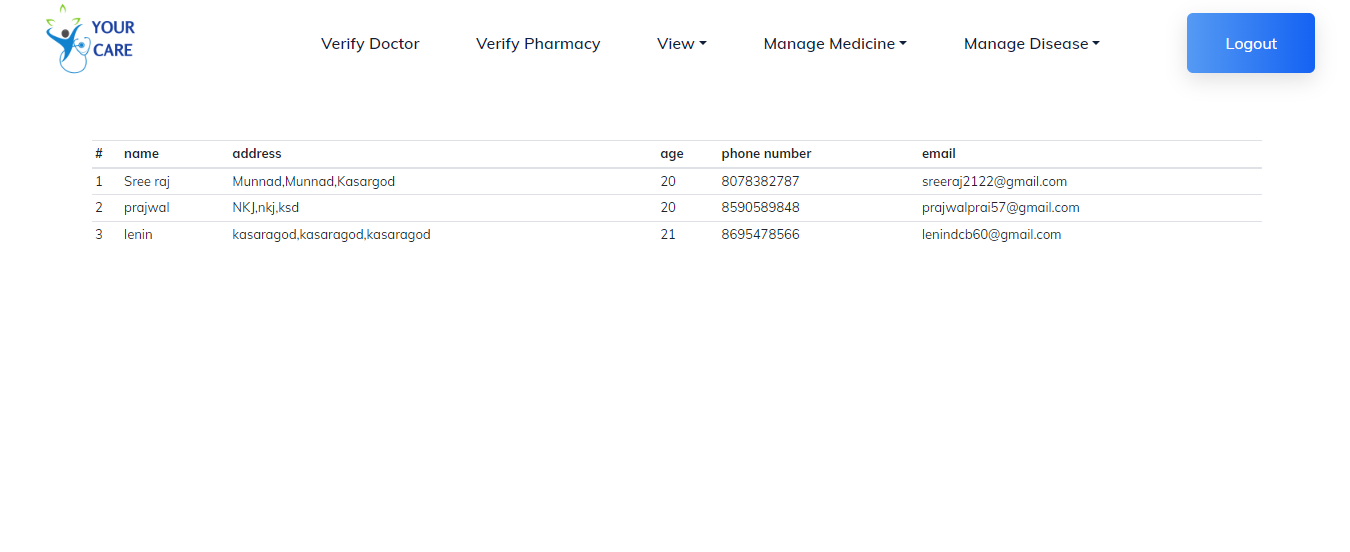
**Admin View Doctor**

****

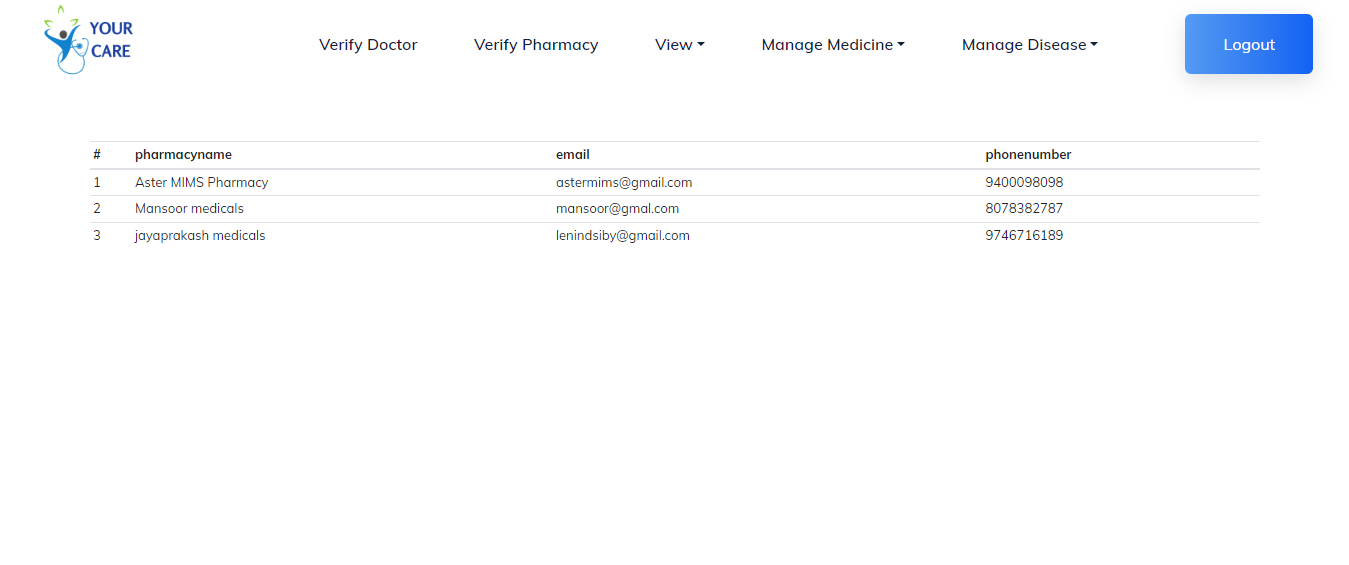
**Admin View Appointment**



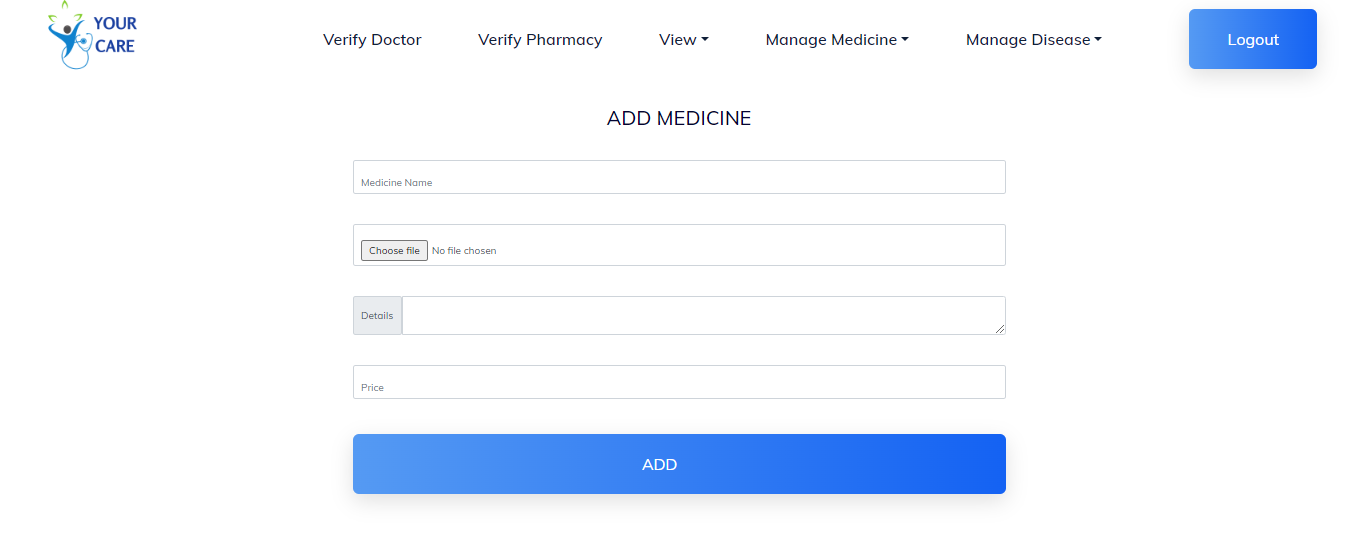
**Admin View User**



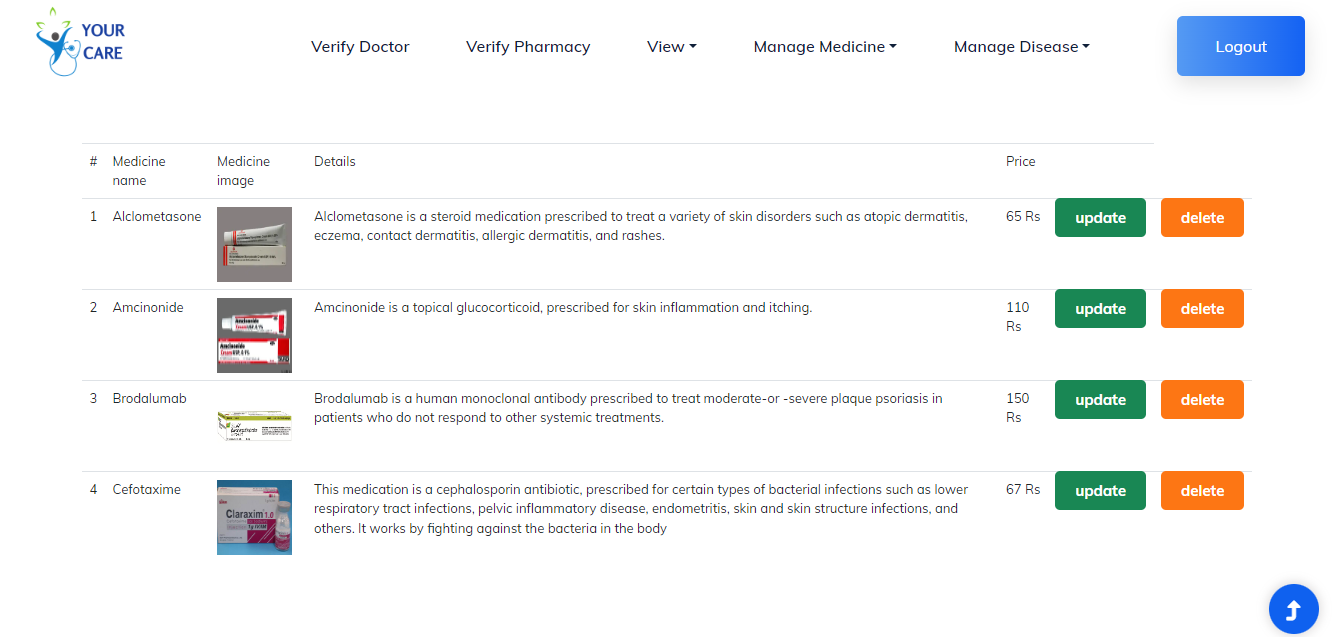
**Admin View Pharmacy**

****

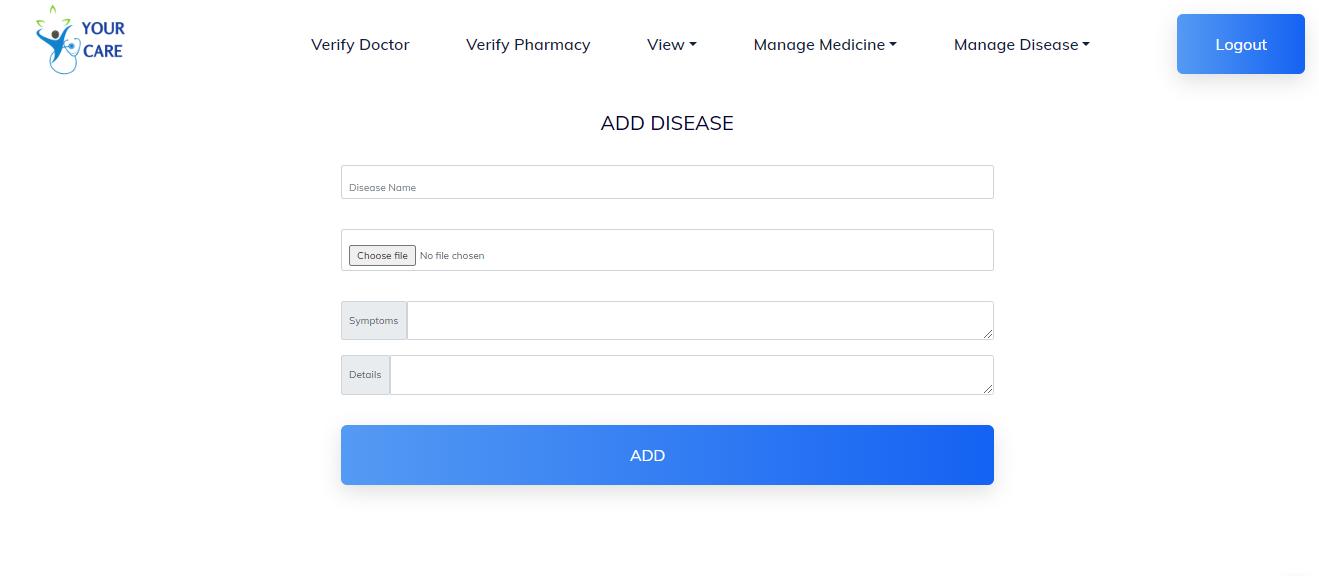
**Admin Add Medicine**



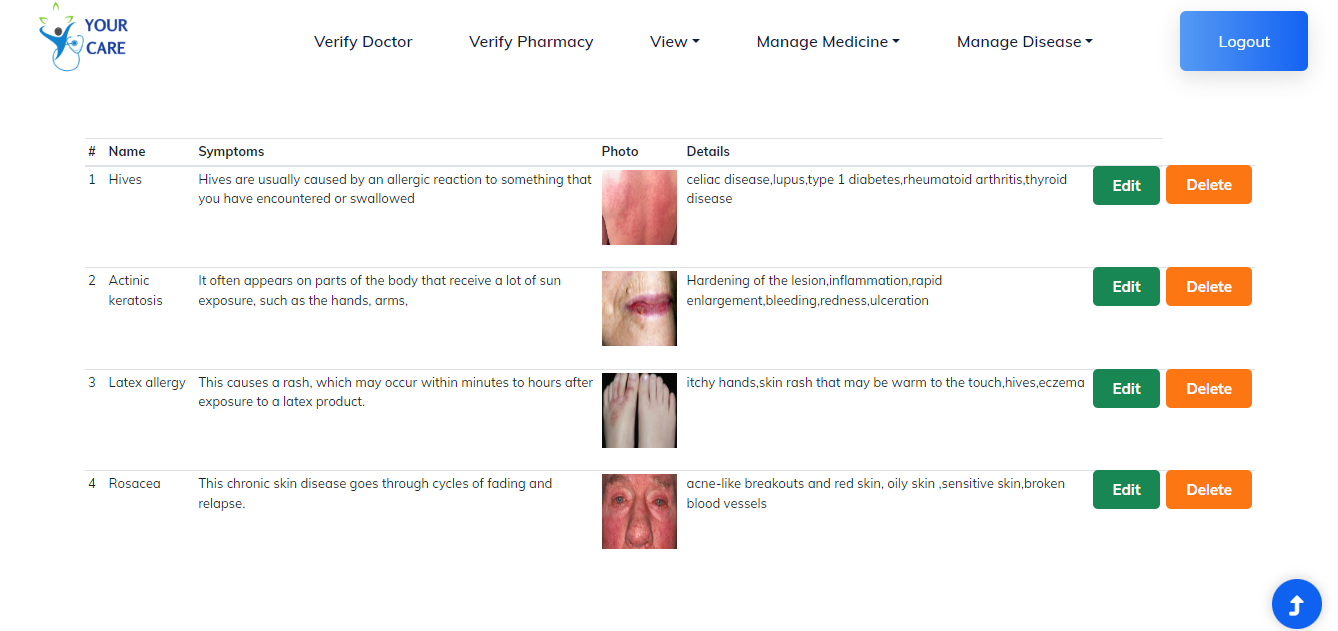
**Admin View Medicine**



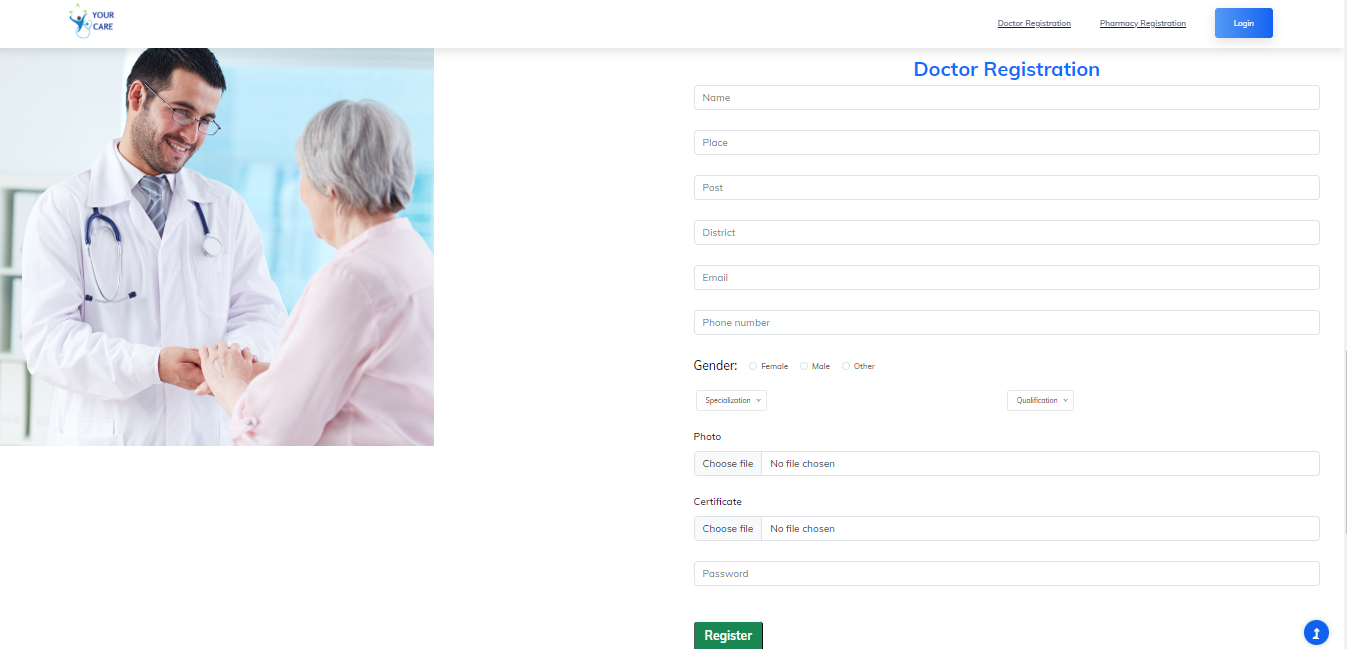
**Admin Add Disease**



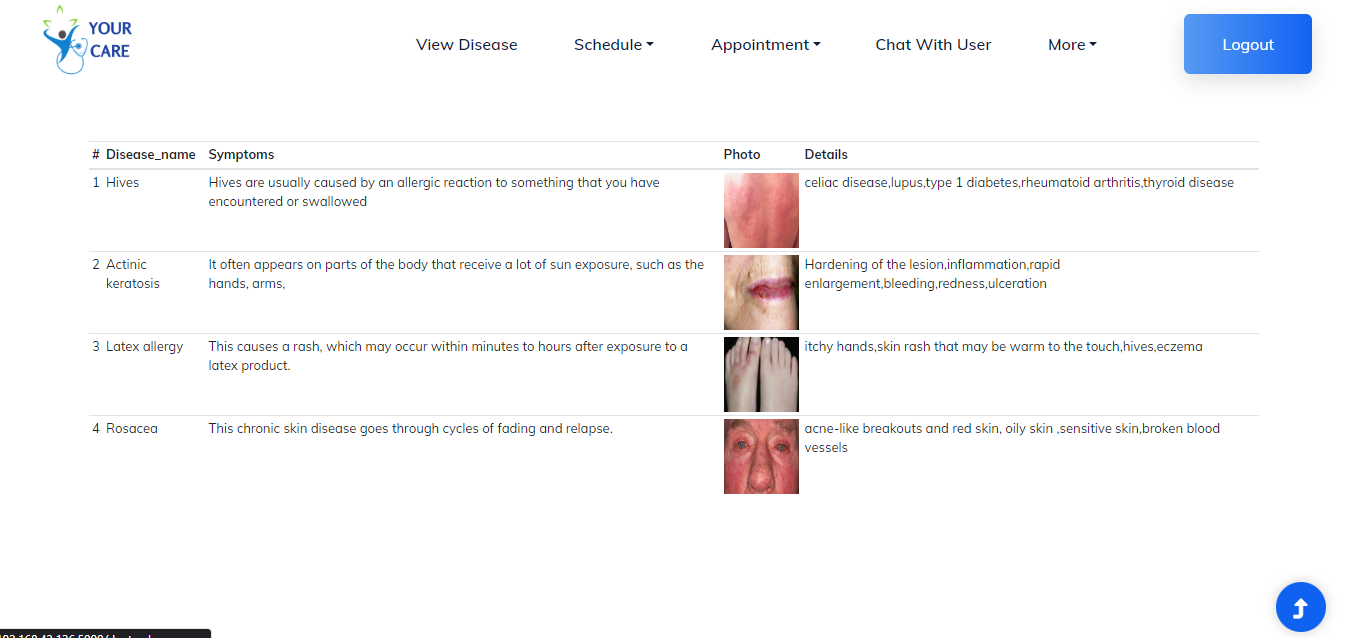
**Admin View Disease**



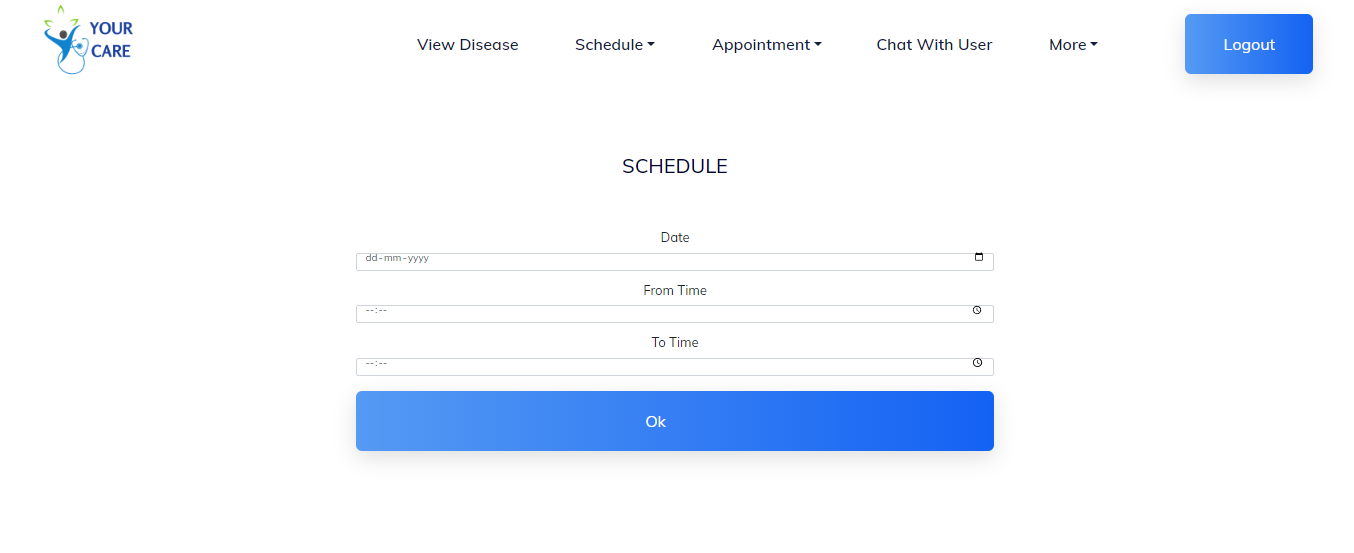
**Doctor Registration**

****

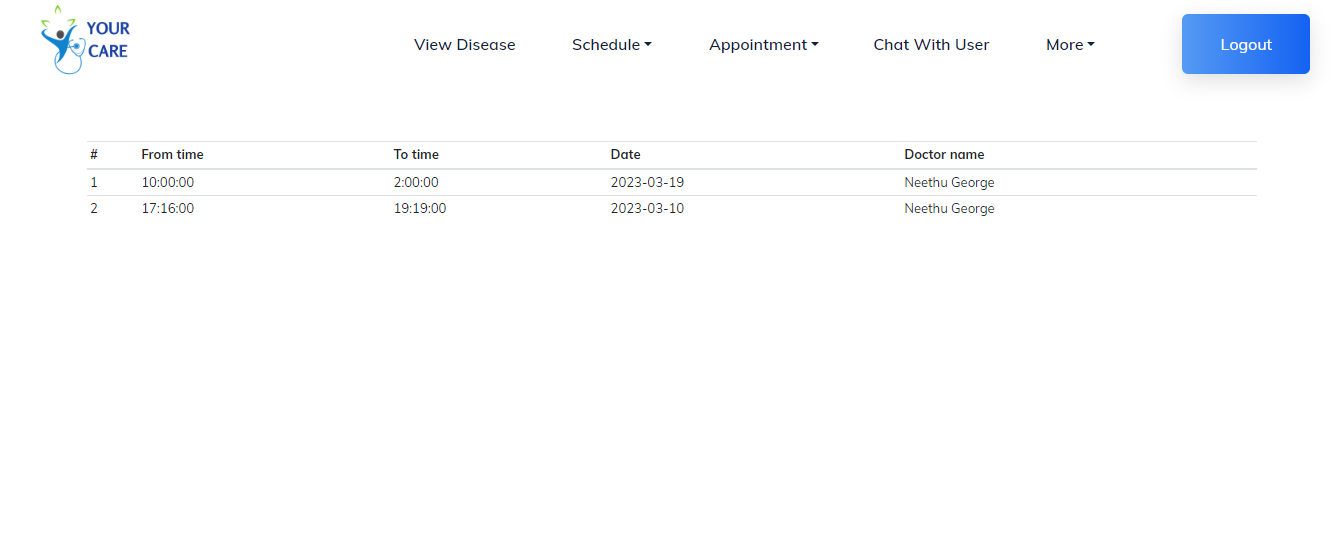
**Doctor View Disease**

****

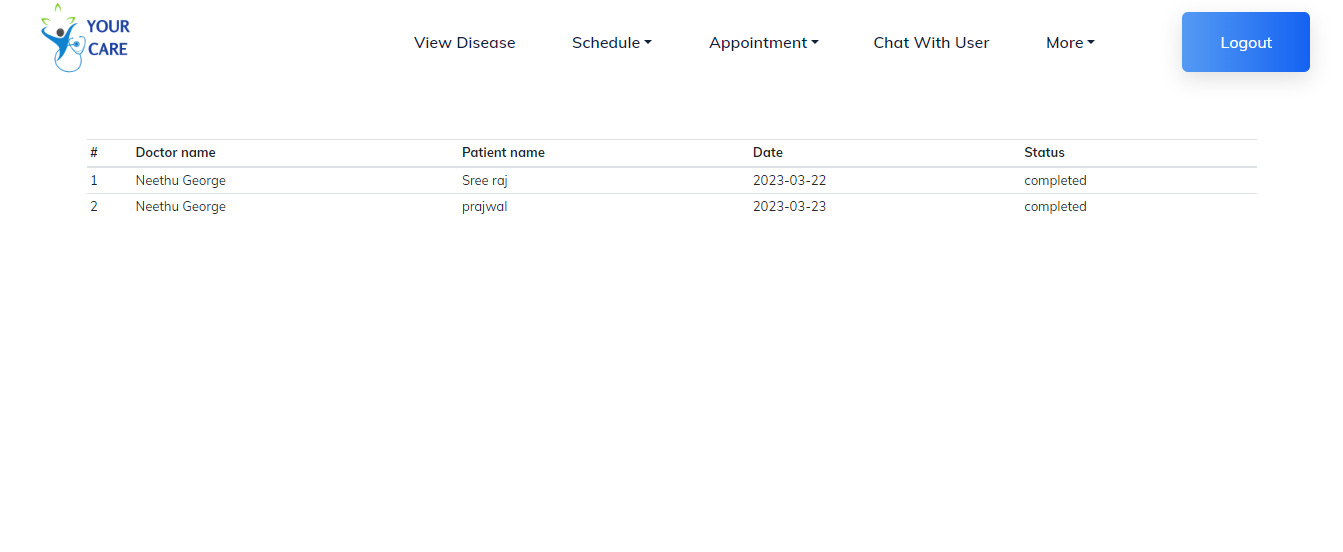
**Doctor Schedule**

****

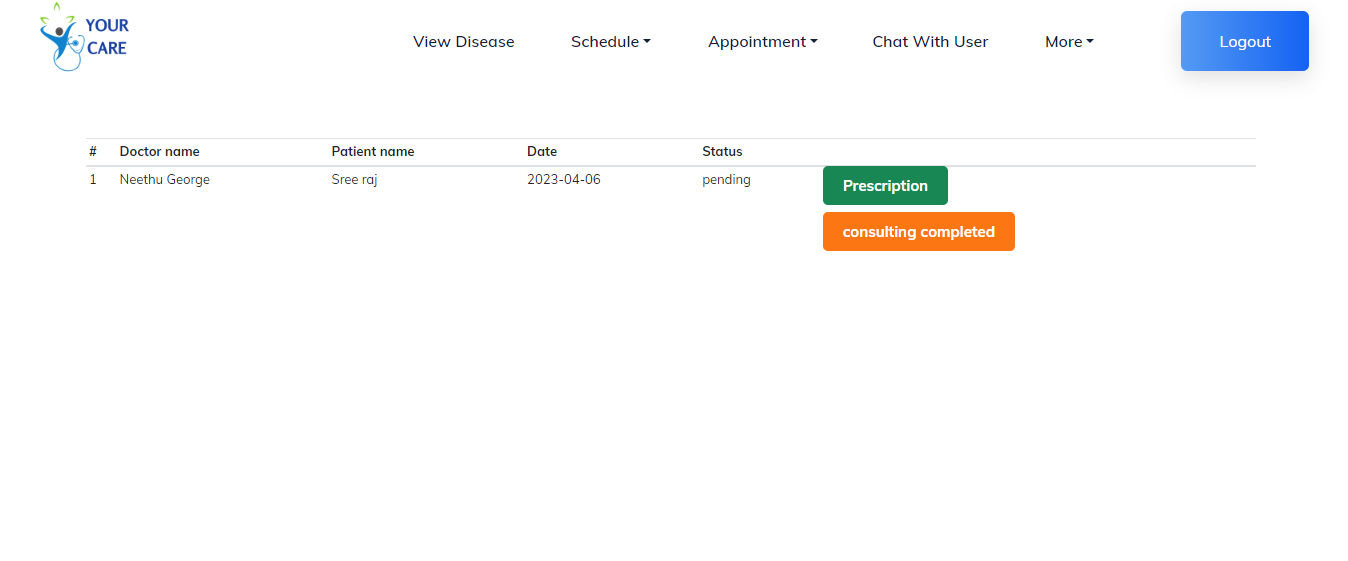
## Doctor View Schedule Date

****

## Doctor View Appointment



## Doctor View Today Appointment

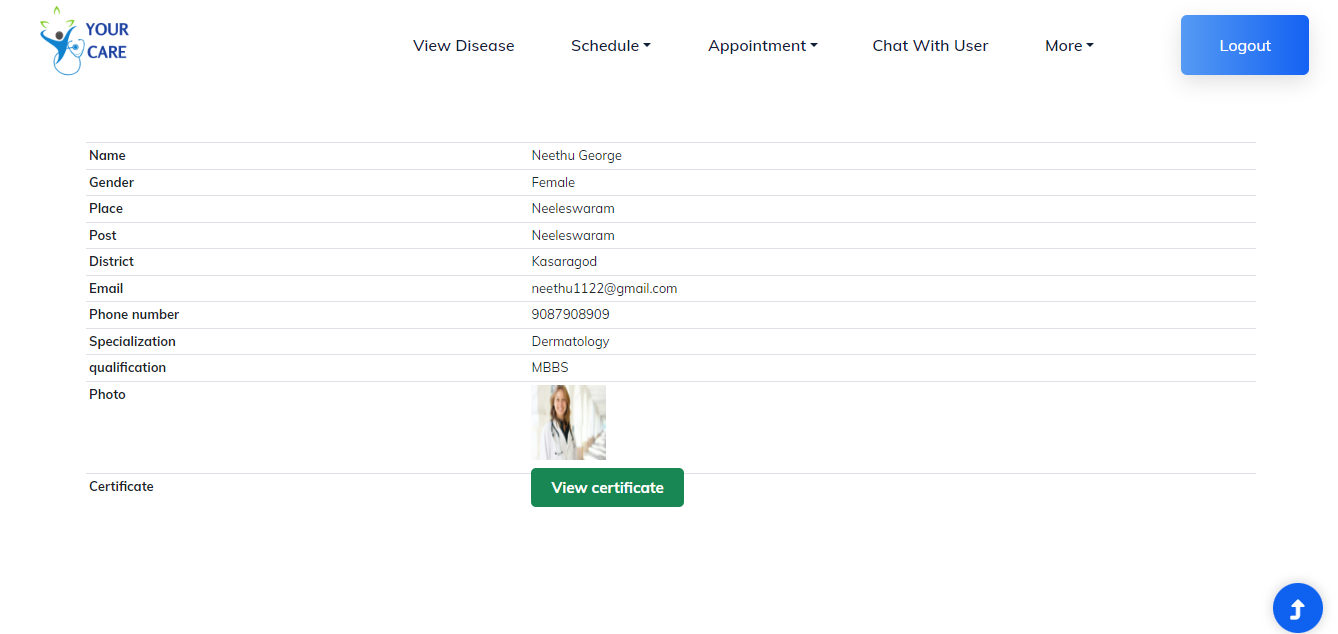
****

## Doctor Chat With User

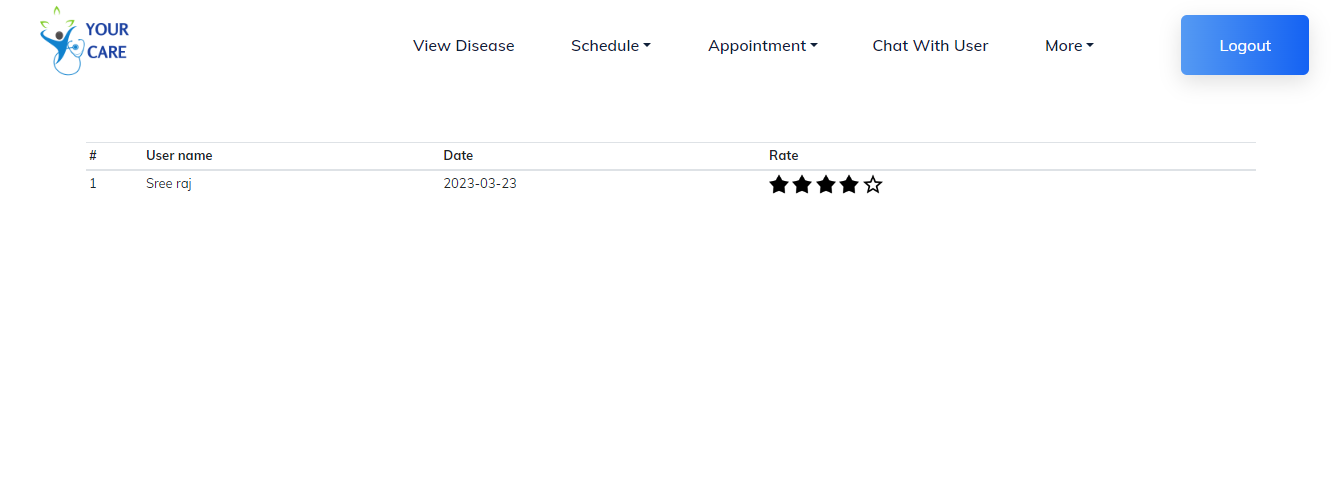
## 

## 

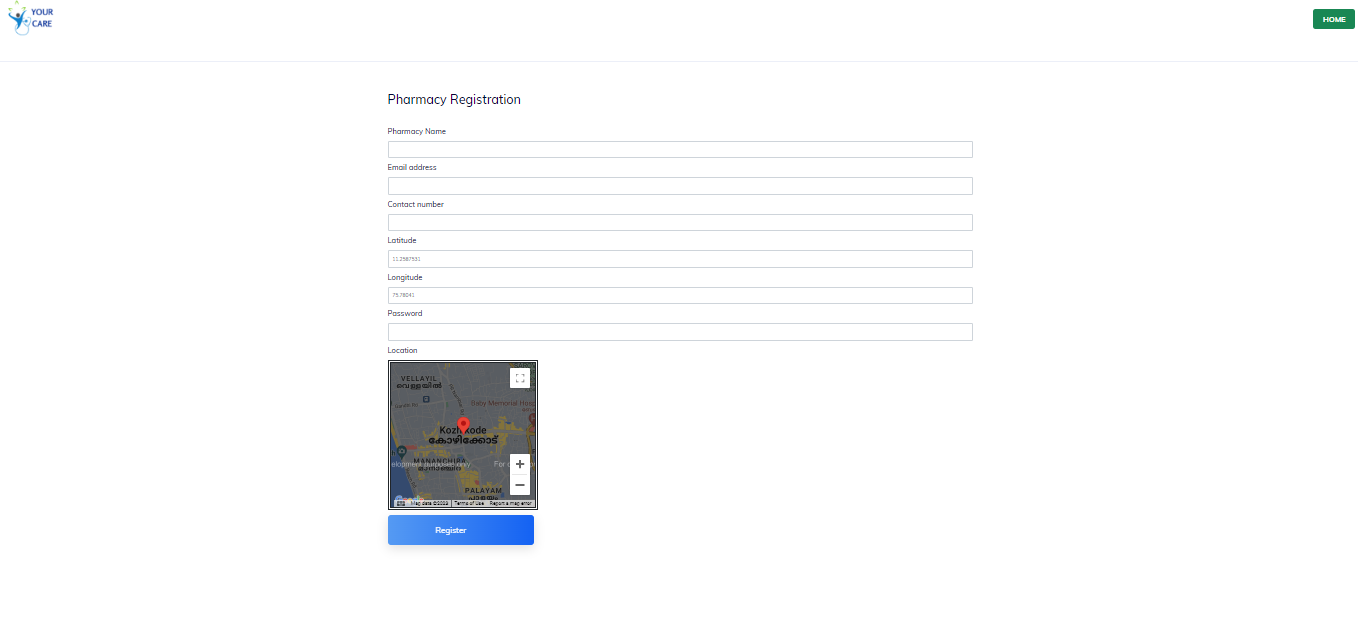
## Doctor View Profile

****

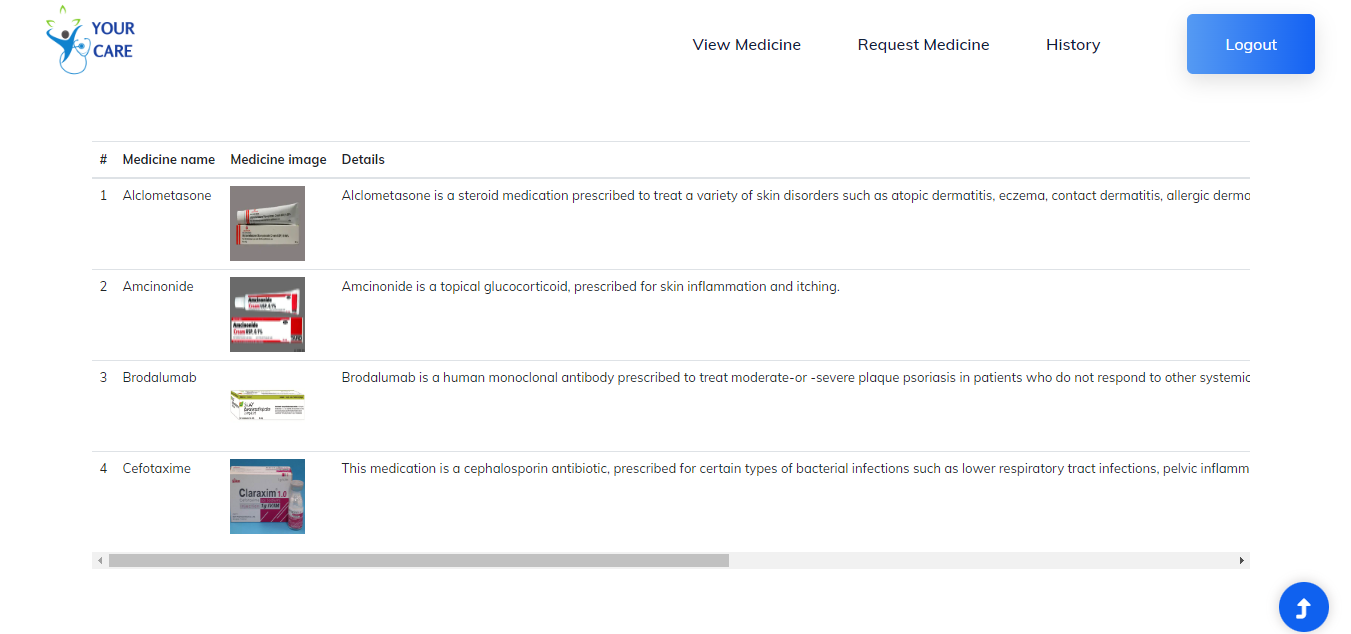
## Doctor View Rating

****

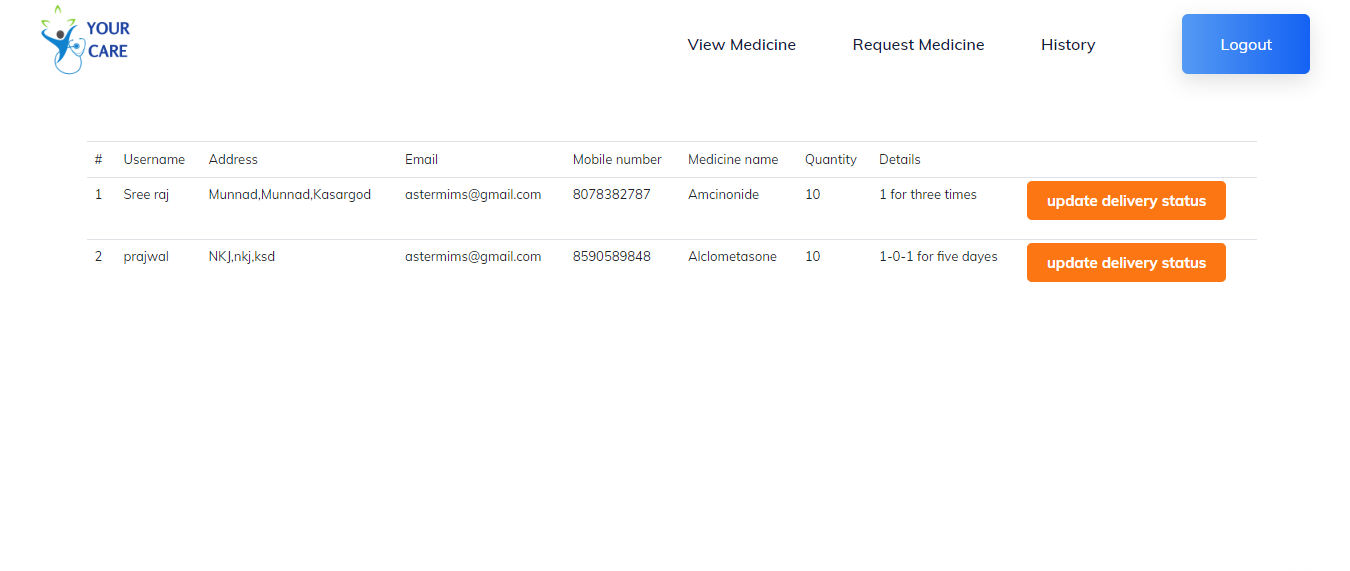
## Pharmacy Registration



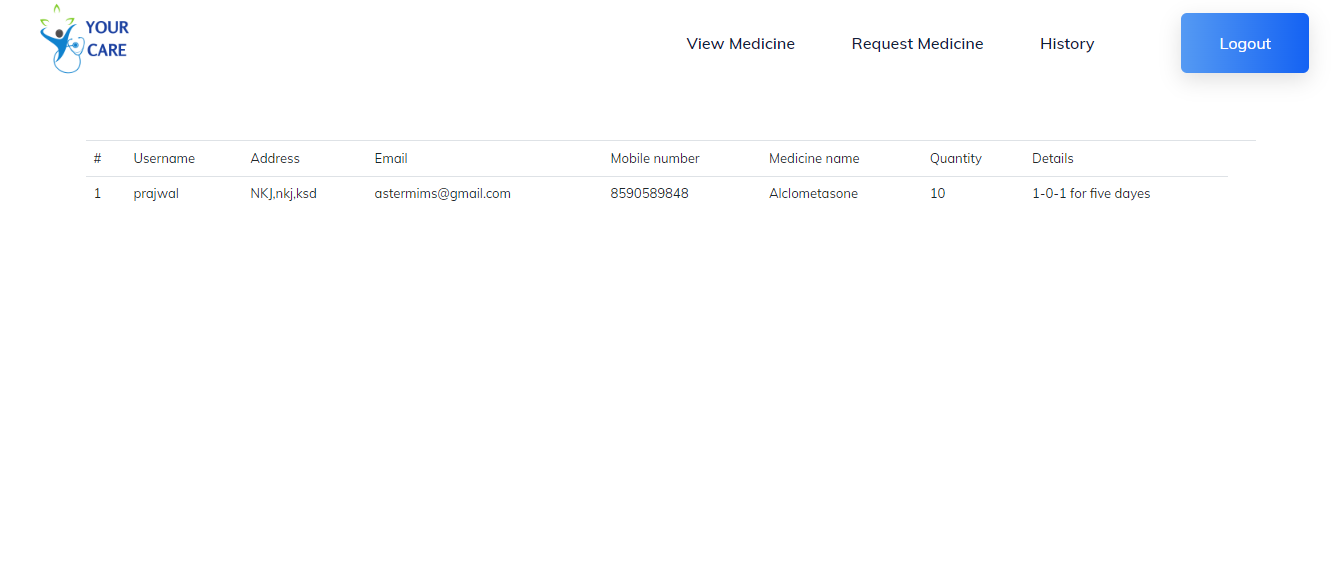
## Pharmacy View Disease

****

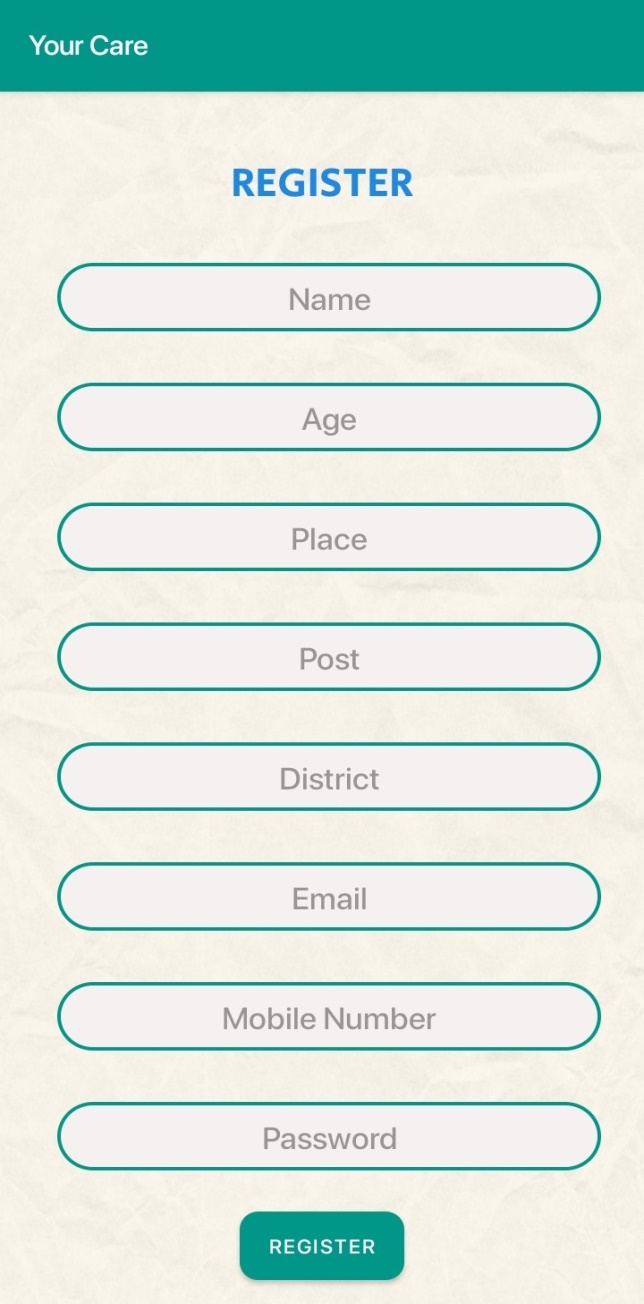
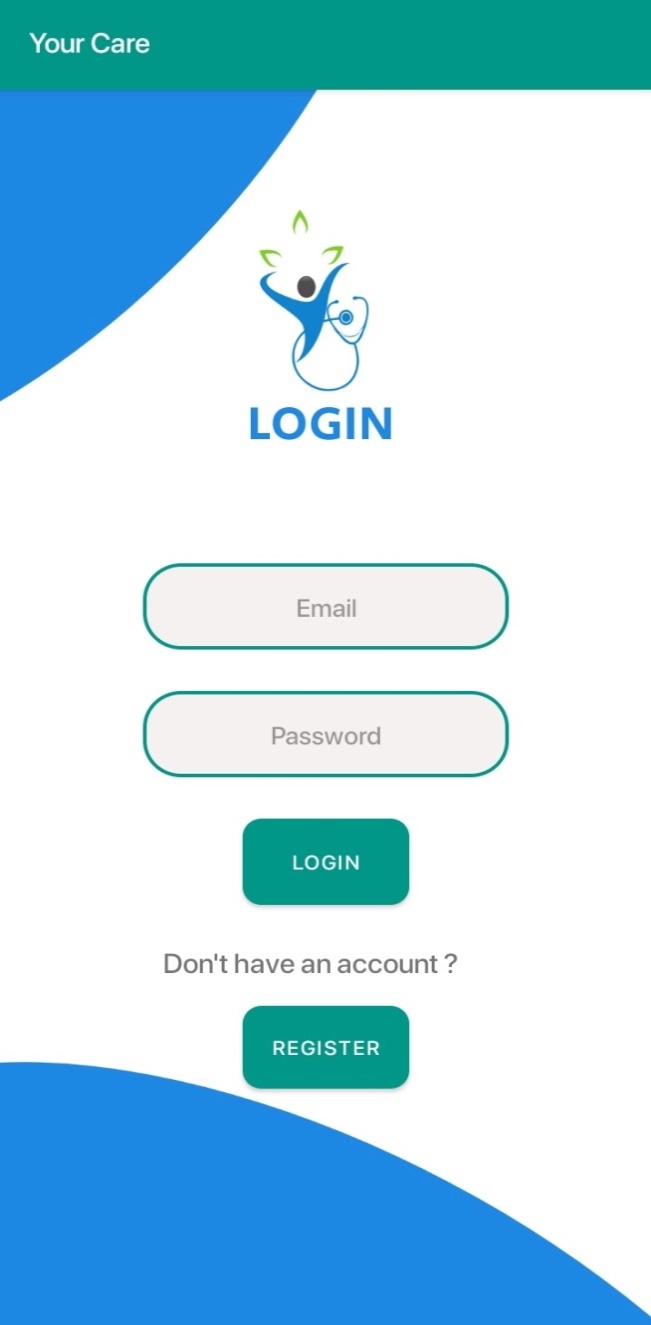
## Pharmacy View Request Medicine

****

## Pharmacy View History

****

## User Register User Login

## User View Profile User Home

## 

## User Upload Disease Affected Area User View Doctor Details

## 

## User Rate Doctor User Chat With Doctor

## 

## User View Appointment Details User View Disease Details

## 

## User View Prescription User Send Request Medicine

## 

## User View Medicine request

## 

**BIBILIOGRAPHY**

**BIBILIOGRAPHY**

[www.bootstrap.com](http://www.bootstrap.com/)

[www.w3schools.com](http://www.w3schools.com/)

[www.stackoverflow.in](http://www.stackoverflow.in/)