# **Acknowledgement**

I want to express my greatest gratitude to all those possible who helped me to finish this project. It was never going to happen without the support of my teachers and beloved friends. I am deeply indebted to my teacher **Mr. Kiran Rana** whose suggestions, help and encouragement boosted me while I was passive in my research. All the researches I have done, made me more fulfilled with the knowledge and tech-experience. Helping during the class time and giving proper guidance while I was in dilemma **Mr. Niman Maharjan**, made me feel calm and composed while dealing with the code failures and bugs. Any satisfactory progress cannot be made without the proper guidance and continuous support; therefore, I want to thank all my classmates, family and teachers for the incessant support and encouragement.

# **Abstract**

Contents

[**Acknowledgement** 1](#_Toc6349633)

[**Abstract** 1](#_Toc6349634)

[**Chapter 1: Introduction** 4](#_Toc6349635)

[1.1: Background of the project 4](#_Toc6349636)

[1.2: Overview of the project design 4](#_Toc6349637)

[1.3: Problem statement 4](#_Toc6349638)

[1.4: Aims and Objectives 5](#_Toc6349639)

[1.4.1: Aims 5](#_Toc6349640)

[1.4.2: Objectives 5](#_Toc6349641)

[1.5: Scope and Limitations 5](#_Toc6349642)

[1.5.1: Scope 5](#_Toc6349643)

[1.5.2: Limitations 5](#_Toc6349644)

[1.6: Features 5](#_Toc6349645)

[**Chapter 2: Analysis** 6](#_Toc6349646)

[2.1: Introduction 6](#_Toc6349647)

[2.2: Waterfall methodology 6](#_Toc6349648)

[2.3: Object oriented analysis 7](#_Toc6349649)

[2.4: System Requirement Specification (SRS) 7](#_Toc6349650)

[2.5.1: Functional Requirement 7](#_Toc6349651)

[2.5.2: Non-functional Requirement 9](#_Toc6349652)

[2.6: Requirement Prioritization 10](#_Toc6349653)

[2.7: Use case Diagram 11](#_Toc6349654)

[2.8: Natural language analysis (NLA) 15](#_Toc6349655)

[2.9: Initial class diagram 15](#_Toc6349656)

[2.10: Architecture 16](#_Toc6349657)

[2.10.1: System Architecture 16](#_Toc6349658)

[**Chapter 3: Design** 17](#_Toc6349659)

[3.1: Structural Modelling 17](#_Toc6349660)

[3.1.1 Class diagram 17](#_Toc6349661)

[3.1.2: Data flow diagram 18](#_Toc6349662)

[3.2: Dynamic Modelling 18](#_Toc6349663)

[3.2.1: Activity diagram 18](#_Toc6349664)

[3.2.2: Sequence diagram 23](#_Toc6349665)

[3.3: Database Modelling 27](#_Toc6349666)

[3.3.1: Entity-relationship diagram 27](#_Toc6349667)

[3.3.2: Data dictionary 27](#_Toc6349668)

[3.4 UI modelling 31](#_Toc6349669)

[**Chapter 4: Implementation** 38](#_Toc6349670)

[4.1: Introduction 38](#_Toc6349671)

[4.2: Choice of Language 38](#_Toc6349672)

[4.3: Development Environment 38](#_Toc6349673)

[ Frameworks: 38](#_Toc6349674)

[ Standard libraries: 38](#_Toc6349675)

[4.4: Development Platform 38](#_Toc6349676)

[4.5: IDE 38](#_Toc6349677)

[4.6: Development strategy 38](#_Toc6349678)

[4.7: System Migration 39](#_Toc6349679)

[**Chapter 5: Testing** 39](#_Toc6349680)

[5.1: Introduction 39](#_Toc6349681)

[5.2: Unit Testing 39](#_Toc6349682)

[5.3: Black box testing 40](#_Toc6349683)

[**Chapter 6: Other project issues** 63](#_Toc6349684)

[6.1: Risk Management 63](#_Toc6349685)

[6.2: Configuration management 65](#_Toc6349686)

[6.3: Scheduling and Gantt chart 65](#_Toc6349687)

[6.4: Future work 66](#_Toc6349688)

[6.5: User manual 66](#_Toc6349689)

[6.6: Limitation 66](#_Toc6349690)

# **Chapter 1: Introduction**

Medino Pharmacy is a vigorous and technology integrated web-based application of an existing store which is located at Maharajgunj, Nepal. The main objective of this project is to provide online service through development of online website.

This website provides superior online experience including easy navigation and secure service. It sells genuine products under various category and it include not only medicine but also other medical products including herbal products, beauty care products etc. It is trusted medical website as it holds legacy of 20 years pharmaceutical success so they are reliable and safe for their customer. They are one of the best medical service providers in the town. (Anon., 2019)

## 1.1: Background of the project

In general, the current existing store is expanding their pharmacy with online services. Medino ease their customer to place order online and have their required medicine at doorstep without parting the comfort zone of their home. With online service, customer can have their product from anywhere at any time with possible quick deliverance. Buying from online is really easy and fast.

The main motive is to provide reliable and affordable medicines with fast online access and convenient home delivery. This application facilitates the customer to view genuine information of medicine and help easy medicine access. It includes various medicine under different product category. And also, they are intending to replace their manual system with computerized system reducing their day to day manual effort. Here admin as store’s pharmacist manages all medicines and other products and he or she will also be deciding to approve the order made by the customer. Customer can simply log on to the website and have their order and payment is done at delivery time. Medino will be trying to provide all possible flexibility to their customer as customer are valued highly to continue any project. And customer can also buy prescribed medicine by uploading their prescribed report suggested by their doctors. Customer can easily buy medicine under other category except prescribed medicine as it need an authorized document for its selling (due to legal issue).

## 1.2: Overview of the project design

The project is designed with all possible key features and major criteria. It is enhanced with user-friendly user-interface for convenient working environment which helps admin for easy operation to manage the system. And also improve user satisfaction to use the system. The website is mobile-compatible. User authentication and other security measures are implemented to secure the system. Authenticated users are enabled to order medicines and other product and admin are permitted to operate and manage the system.

## 1.3: Problem statement

Nowadays, online shopping has been trend for day to day shopping as online buying is fast and easy and also customer gets product at their doorstep. Due to this reason the pharmacy store is being low day by day. So, in order to compete with online sellers and to increase customer, an online site is being developed. And day to day manual work was really time consuming and needs more manual effort so computerized system is being implemented. Manual paper work while calculating the budgets brought many financial errors resulting business loss and poor customer service. To overcome this, auto calculation and bill generating system is being developed.

## 1.4: Aims and Objectives

### 1.4.1: Aims

* To develop user-friendly pharmacy website for online buying.
* To implement computerized system.
* To manage medicine and other products.
* To reduce daily paper work and overcome manual efforts.
* To generate auto bill generation system.

### 1.4.2: Objectives

* Identifying pros and cons of the current store to analyze how the system need to develop further.
* Developing and designing website with easy navigation and secure database.
* Providing user manual for convenient interaction with the system.
* Enhancing secure and reliable ordering of medicines.
* Developing auto calculation system to reduce manual calculation.
* Testing system under different environment before deployment.

## 1.5: Scope

### 1.5.1: Scope

## 1.6: Features

Some of the features included in the system are as follow;

* User registration and login
* View and edit user profile
* Add, Update, Delete medicine category
* Add, Update, Delete medicine details
* View and search medicine
* Order medicine

# **Chapter 2: Analysis**

## 2.1: Introduction

## 2.2: Waterfall methodology

I have chosen Waterfall methodology for the development of my project. Waterfall model is the simplest and earliest software development approach to ensure success of project. It is linear-sequential software development model in which development is seen as flowing steadily downwards like a waterfall through phases of requirement analysis, design, implementation, testing, deployment and maintenance. In this model, each phase must be completed before beginning another phase and the phases are not overlapping.

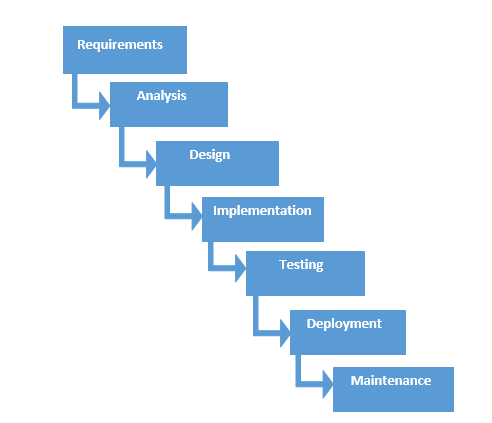


Figure 1: Waterfall model

And the five phases of water model are as follow;

1. **Requirement analysis**

All potential requirements are analyzed and documented in a proper requirement specification document. This phase describes what the system will do.

1. **Design**

With the input of requirement specification from first phase, overall system architecture is specified such as hardware and software, programming language etc.

1. **Implementation**

The system is then developed using all models, business logic and service integration.

1. **Testing**

Here the implemented system is tested -using unit testing, white box testing, integration testing and other testers- for its functionality. Any failure and faults are identified.

1. **Deployment**

After testing is done, the product is deployed to the customer environment.

I choose Waterfall model because of the following reasons;

1. As waterfall model is suitable for small projects like mine and it is easy to understand and simple to use.
2. This model works well where the requirements are clear and understandable.
3. Due to its rigidity, it is easy to manage and each phase has specific goals and review process.
4. It helps to measure project milestones.

## 2.3: Object oriented analysis

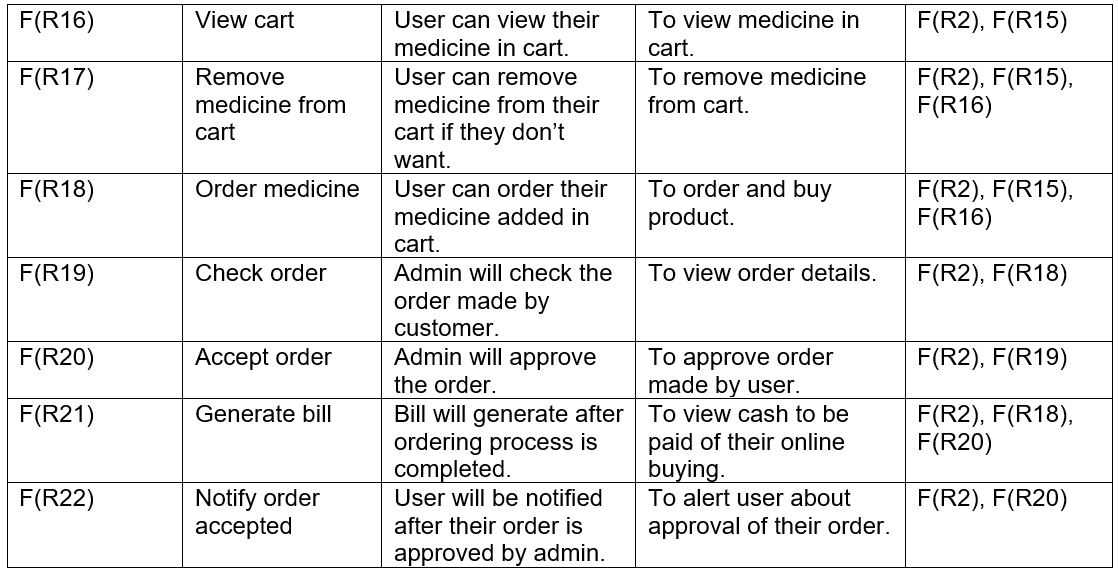
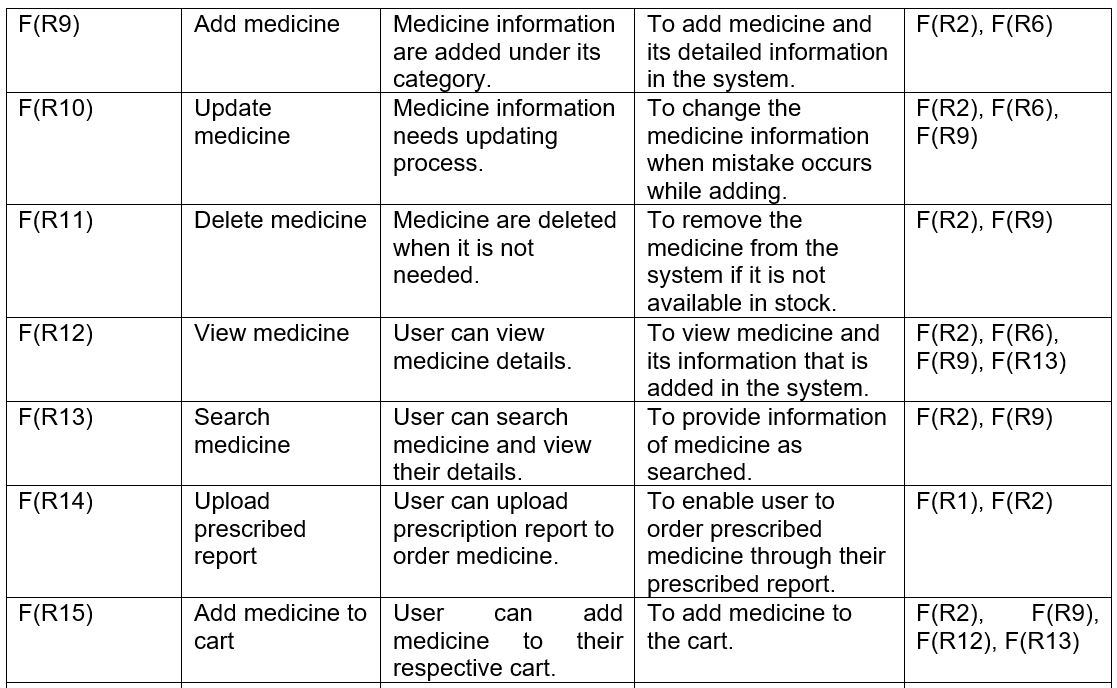
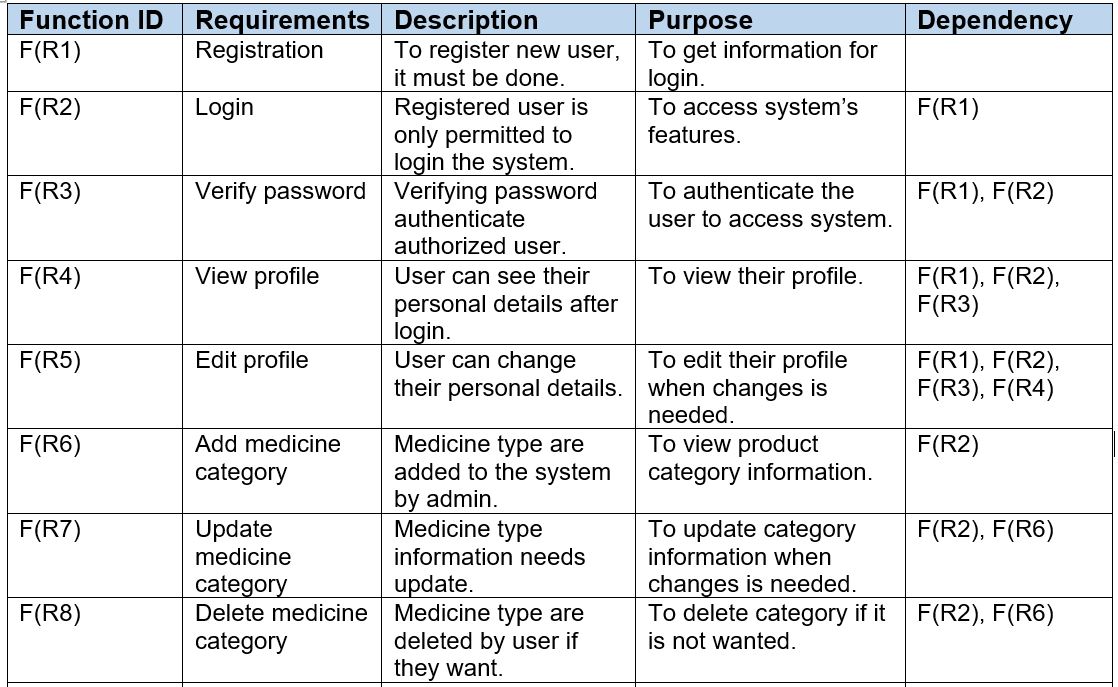
## 2.4: System Requirement Specification (SRS)

System Requirement Specification is a structured set of documents which contains complete description of features and behavior of a system to be developed (Anon., 2019). It includes all functional and non-functional requirement of the project.

### 2.5.1: Functional Requirement

Functional requirement are those requirements which defines the functionality of the system i.e. what the system should do (Anon., 2019).It includes significant features and functions that a user expects from the system.

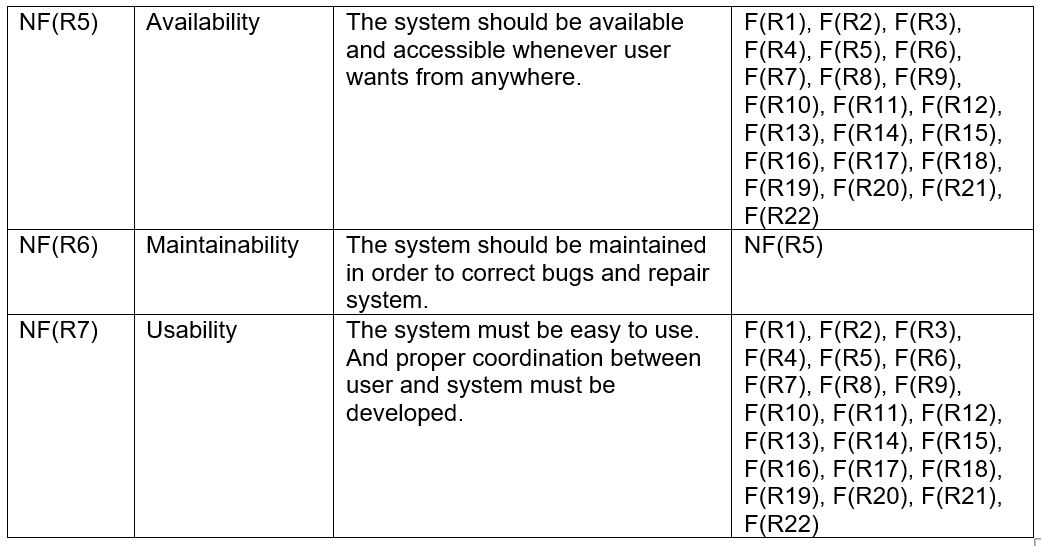
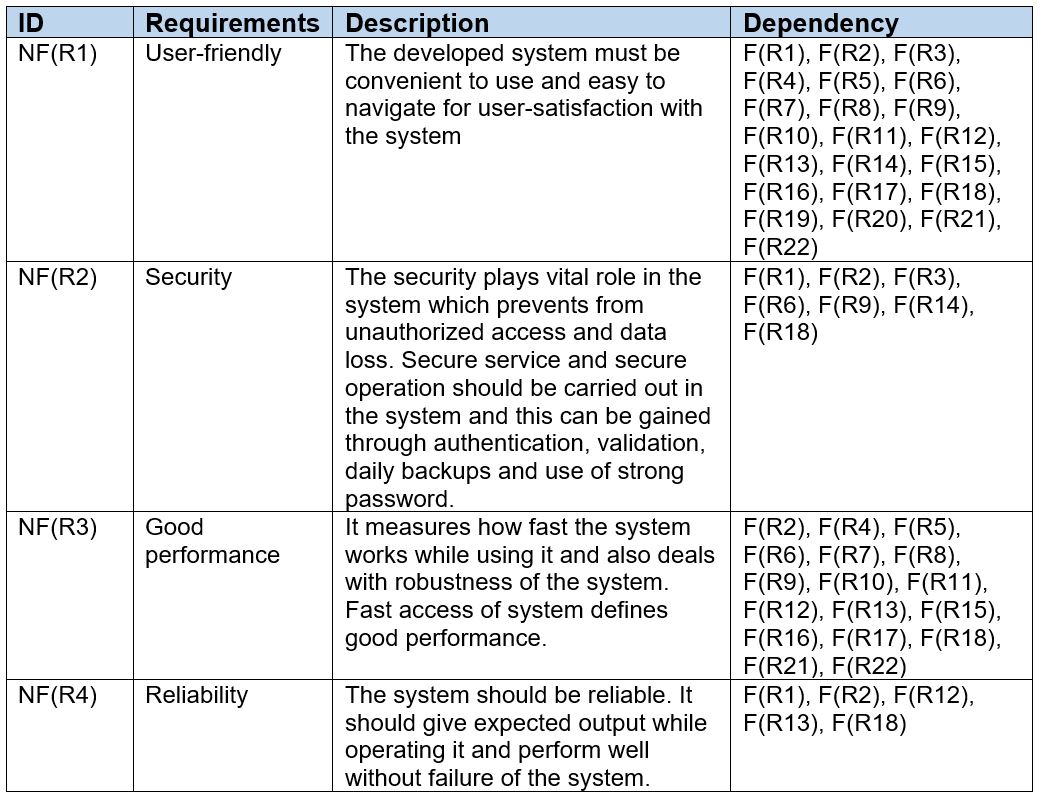
The several functional requirements included in the project are listed and described below;



## 2.5.2: Non-functional Requirement

Non-functional requirement are those requirements which defines the overall characteristics of the system i.e. it specifies how the system will perform a certain function (Anon., 2019).These requirements are also known quality attributes as it defines the quality of the system developed.

The non-functional requirements of the system are as follow;



## 2.6: Requirement Prioritization

Prioritization is the technique to understand and manage the requirements according to its relative significance in the system. It is necessary to prioritize the requirement in order to keep deadlines and progress. And highly prioritized requirement will be delivered in early stage during the development.

MoSCoW prioritization is used to identify the importance of the requirement. And MoSCoW acronym is made with first four letter i.e.

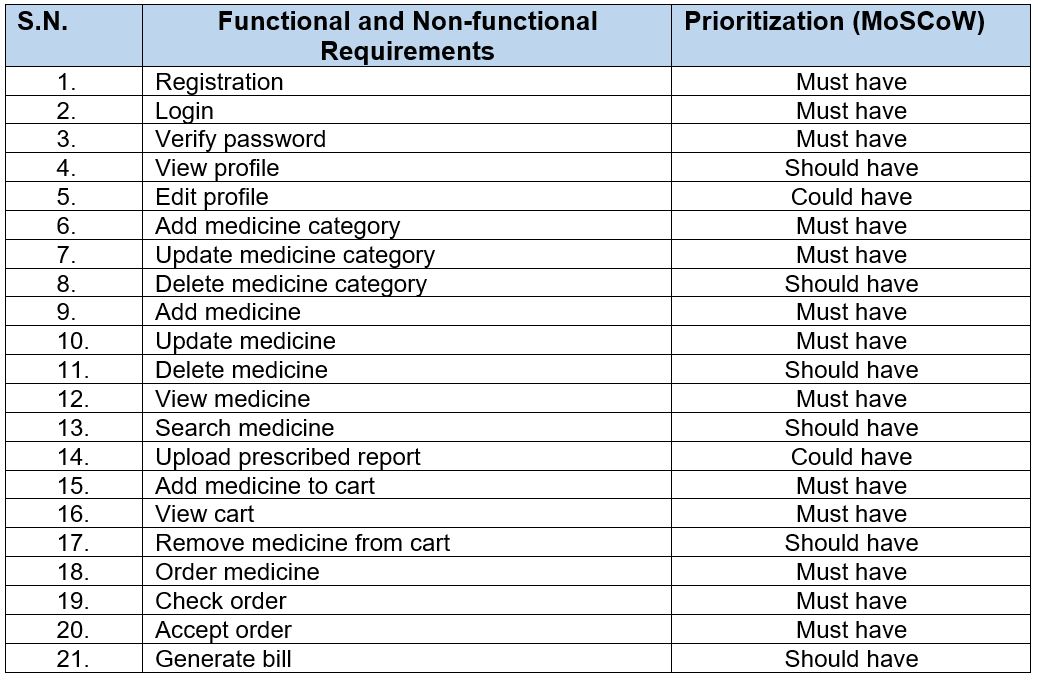
**M**ust have: Requirement is important and guaranteed to delivered on exact deadline.

**S**hould have: Requirement is necessary but not vital.

**C**ould have: Requirement is less important but desirable.

**W**on’t have: Requirement that won’t have now but will be delivered in future. (Anon., 2019)

The MoSCoW prioritization of functional and non-functional requirements are as follow;



## 2.7: Use case Diagram

Use case diagram is a tool which portray broad interaction between actors and the system through its functionality defined as use case. It helps in requirement analysis as well as modelling the behavioral structure of the system. Also, it visualizes the functional requirement of the system translating them into a development priority. (Anon., 2019)

The symbol and notation used in use case diagram are;

* **System:** A rectangular box showing system’s boundaries and contain use cases.
* **Use case:** A oval shape showing the role of the user within the system.
* **Actors:** A stick figure showing user who performs action in the system. Here, Pharmacist and customer are the actors.
* **Relationships:** A simple line showing the relation between actors and their related use cases.

The use case diagram representing actors and their functionality is below;



Above diagram is the use case diagram which contains two types of actors i.e. Admin (one who manages all the stuffs in the system) and customer (one who order and consume the medicine). Both actors should register in a system providing their information and then login with their unique email address and password. After login both actors can view and edit their profile. They can change their password whenever they like. Customer can add their required product in cart and order. Admin can only operate the system and record and accept order made by the serious customer. General customer can view and search medicine. Authenticated customer can upload their prescribed report and place their order.

The use cases are explained below;

* **Registration**

To register in the system, user will have to provide their information along with unique email address.

* **Login**

With correct email address and password, user can log into the system and can access other functionality.

* **View profile**

After authentication, user can view their personal information.

* **Edit profile**

Edit profile is a function which will help user to update their profile when they like to change their details.

* **Add category**

Add category is the process of adding medicine category in the system. Admin can only use this function after registration and login.

* **Update category**

Update category is the function where admin can update the medicine category details if there is any mistake while adding the category.

* **Delete category**

It is the function to delete the medicine category when the category is not needed.

* **Add medicine**

Add medicine is the process of adding a medicine in the system. Admin will be using this function to add different types of medicine.

* **Update medicine**

Update medicine function will help the admin to update the details of medicine if there is any mistake while filling up the add medicine form.

* **Delete medicine**

This function helps admin to delete the medicine when the medicine is out of stock.

* **View medicine**

View medicine is the function which helps the user to view the details of medicine.

* **Search medicine**

Search medicine is the process of searching the medicine the user wants. Any user can search medicine and see their details.

* **Upload prescribed report**

Registered customer can upload their prescription report and place their order for the medicine.

* **Add to cart**

Customer add their required medicine and other products in the cart.

* **Proceed to order**

Logged-in user can order medicine added in their cart.

* **Check out**

Customer check their products and payment amount after proceeding their order.

* **Check order**

It’s a function where admin will view the list of order of the medicine.

* **Accept order**

Admin will accept the order after checking the requested order. Order will only be accepted if the medicine is available in the stock.

* **Generate bill**

After accepting order, bill will be generated. Bill will be saved as well as printed.

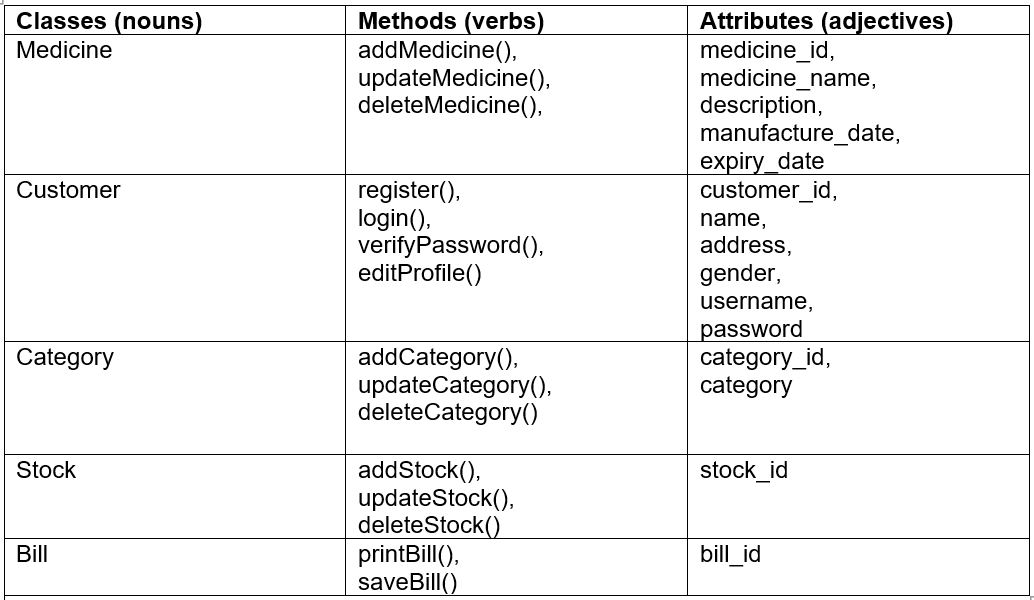
* **Notify order accepted**

The user will be notified after their order is accepted by the admin.

## 2.8: Natural language analysis (NLA)

Natural language analysis is the procedure to obtain class diagram by identifying candidate nouns, verbs, and adjectives where noun represents class, verb represents functionality of the class and adjective represents attribute of the class.

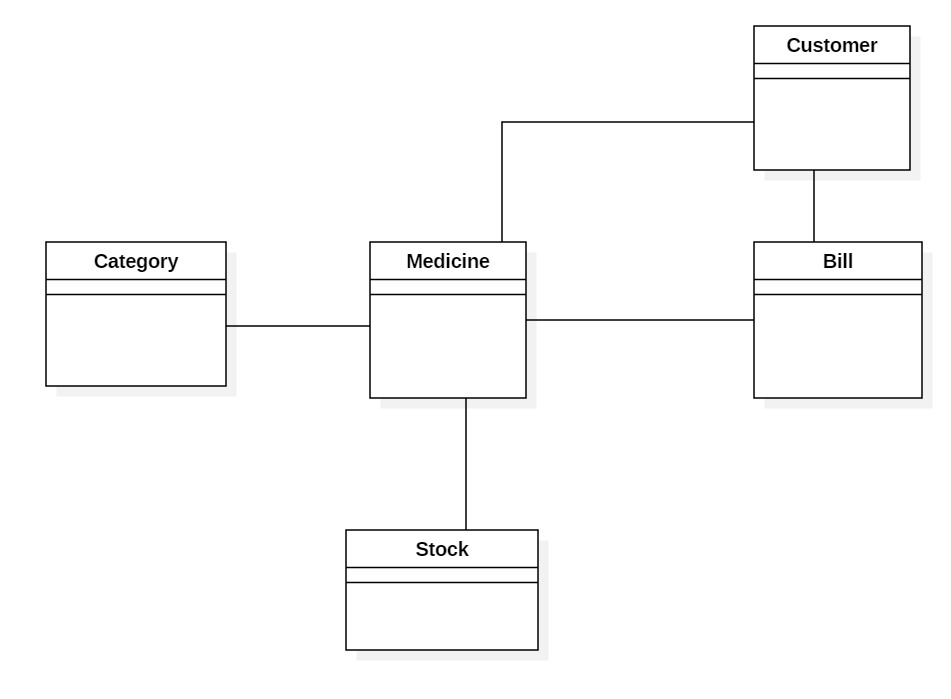
And candidate noun, verb and adjective for the system is identified by analyzing the background of the project. The candidate classes, methods and attributes are as follow:



## 2.9: Initial class diagram

The static structure providing the overview of the system through classes and defining the relationships among the classes.

And the initial class diagram is shown below;



The above class diagram is the initial phase which is designed through the help of candidate classes identified using Natural language analysis. It includes 5 classes which are interlinked with each other with association relationship. These classes are most likely to be included in the final class diagram. Initial class diagram helps to visualize the classes and their relationship for the construction of final class diagram.

## 2.10: Architecture

### 2.10.1: System Architecture

MVC architecture is used to implement the system and MVC stands for “Model-View-Controller” (Anon., 2019). It is design model of three interconnected parts and is widely used for developing user-interface.

The three interconnected part includes model, view, and controller. And each component has their own role in MVC architecture.

1. **Model**

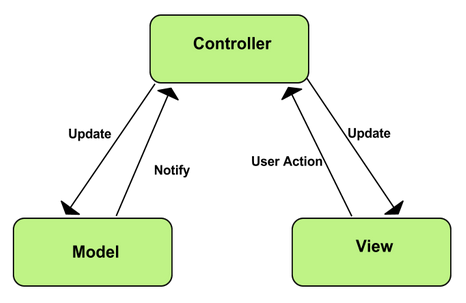
It carries data and can contain business logic used by the program.

1. **View**

It represents the GUI part with which user interact.

1. **Controller**

It controls the data flow into model and update view. It connects user with the system.



The reason behind choosing MVC design pattern for my project are as follow;

1. Supports fast and parallel development system.
2. Provides multiple views.
3. Less code duplication as data and business logic is separated.
4. Modification can be done easily that changes won’t affect entire model.
5. Supports for asynchronous technique.

# **Chapter 3: Design**

Design is logical and physical planning of the project and is the most crucial phase of project development. It implements the system based on user requirements and detailed analysis phase. And the purpose of design is to convert the requirements to a complete and detailed system design specifications.

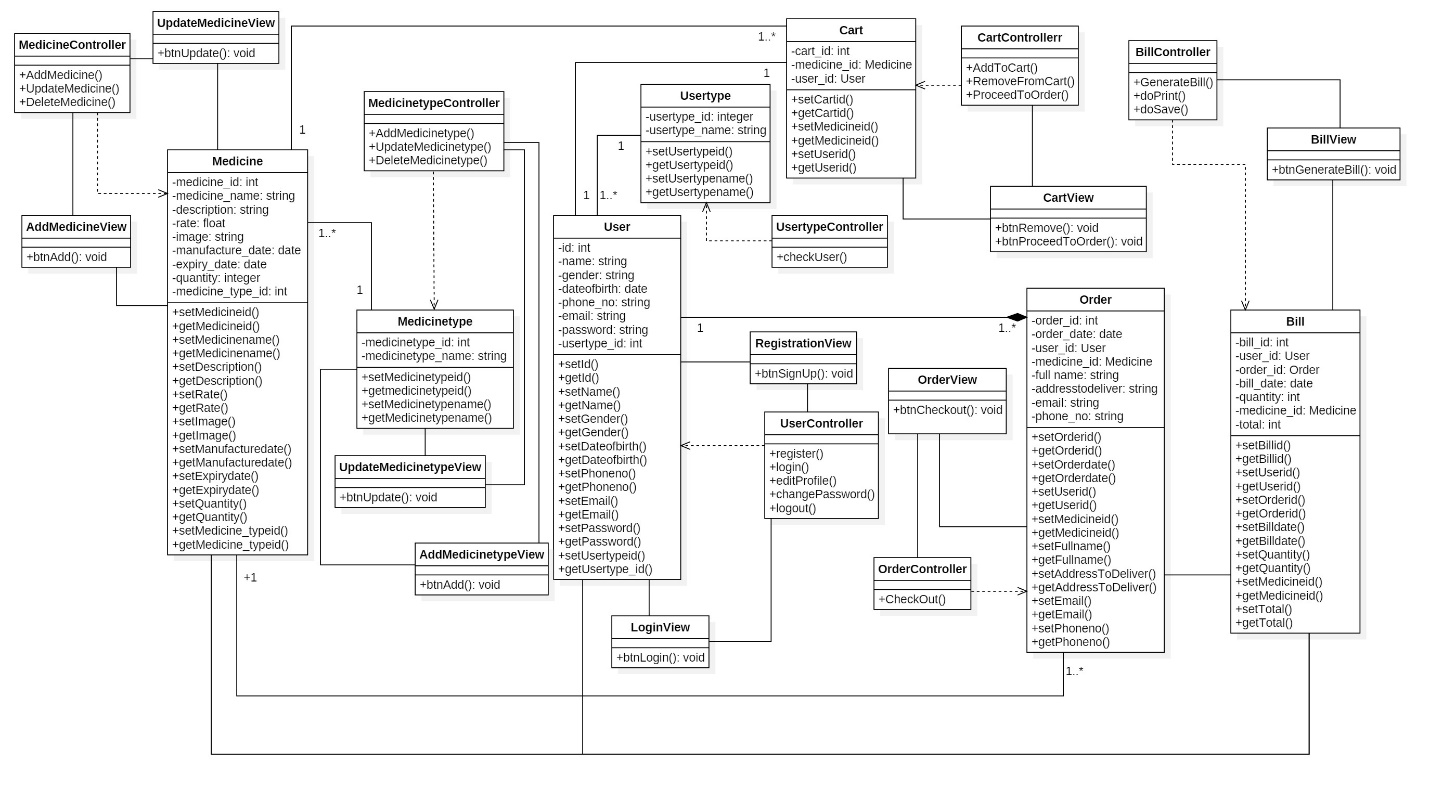
It includes structural and behavioral design for the development of the system.

## 3.1: Structural Modelling

A design that concerns with the structure of the project and is illustrated as diagram. Class diagram and Data flow diagram are structured below.

### 3.1.1 Class diagram

Class diagram is static structure of the system modelling the system’s class, attributes, operations and the relationship between the objects (Anon., 2019). It helps to describe the general overview of the system and provides base for data model.



The represented class in the above class diagram are: user, usertype, medicine, medicinetype, cart, order and bill and they are associated with different relationships. The relationship i.e. used are association, dependency and composition. Each model has their controller which are dependent to them and are within dependency relationship. And view is connected with its controller and model within association relationship. User model have different association relationship as user is the one who buys medicine and add them in cart. User and order are within composition relationship as order cannot be placed without user. Medicine is associated with medicinetype and user. Cart and bill are associated with medicine class.

### 3.1.2: Data flow diagram

Data flow diagram (DFD) is the graphical representation of flow of data through process. It describes the processes involved in system and explains transfer of input data from process to database. It helps to understand logical information flow of purposed system. (Paradigm, 2019)

## 3.2: Dynamic Modelling

A model which concerns with the behavior of the system and define its functionality through diagram. Activity diagram and Sequence diagram of the system is structured below.

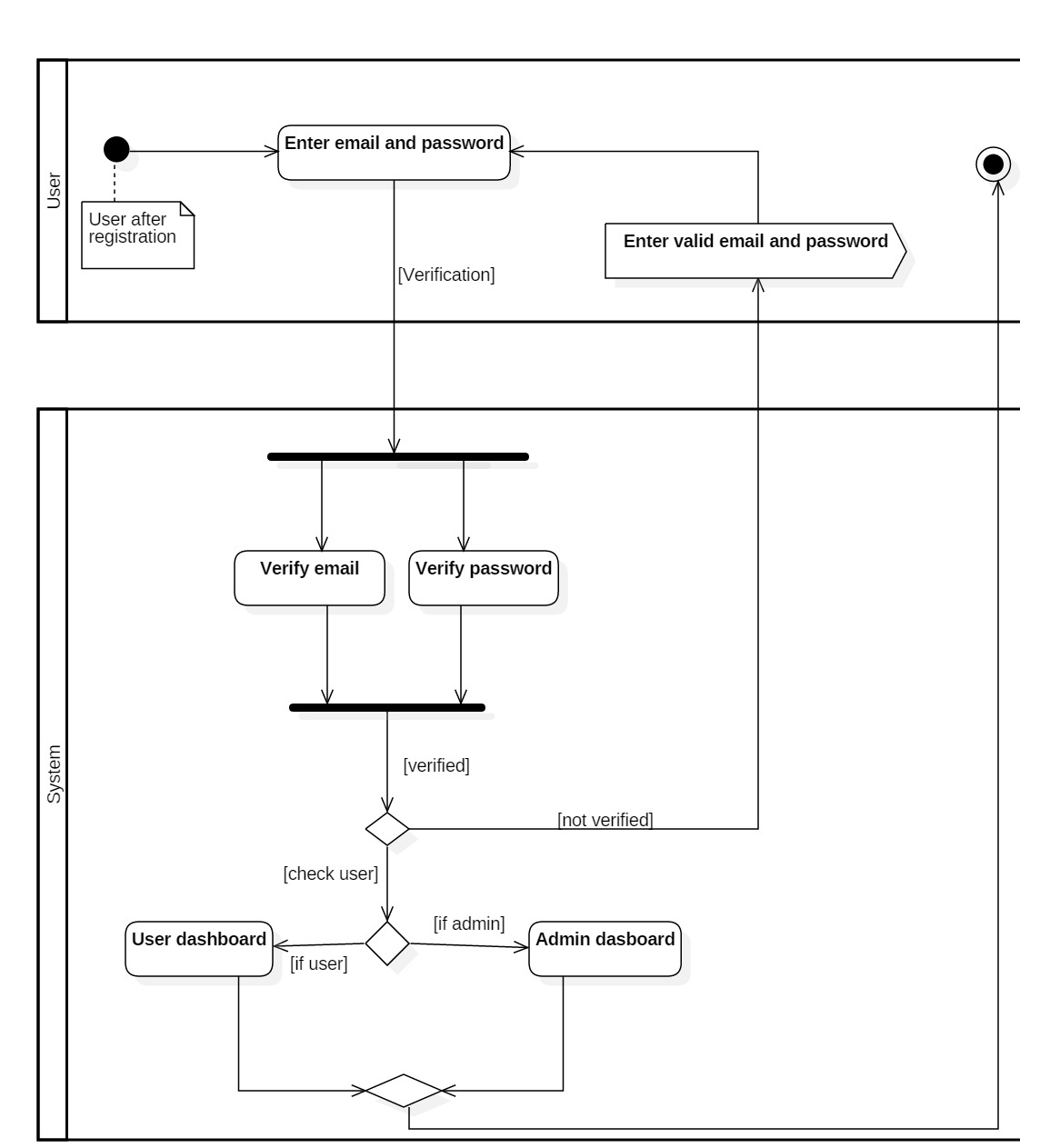
### 3.2.1: Activity diagram

Activity diagram is the dynamic structure modelling the dynamic behavior of the system. It describes the flow of behavior from one activity to another within a system. (Tutorialpoint, 2019)

The activity notation used are as follow;

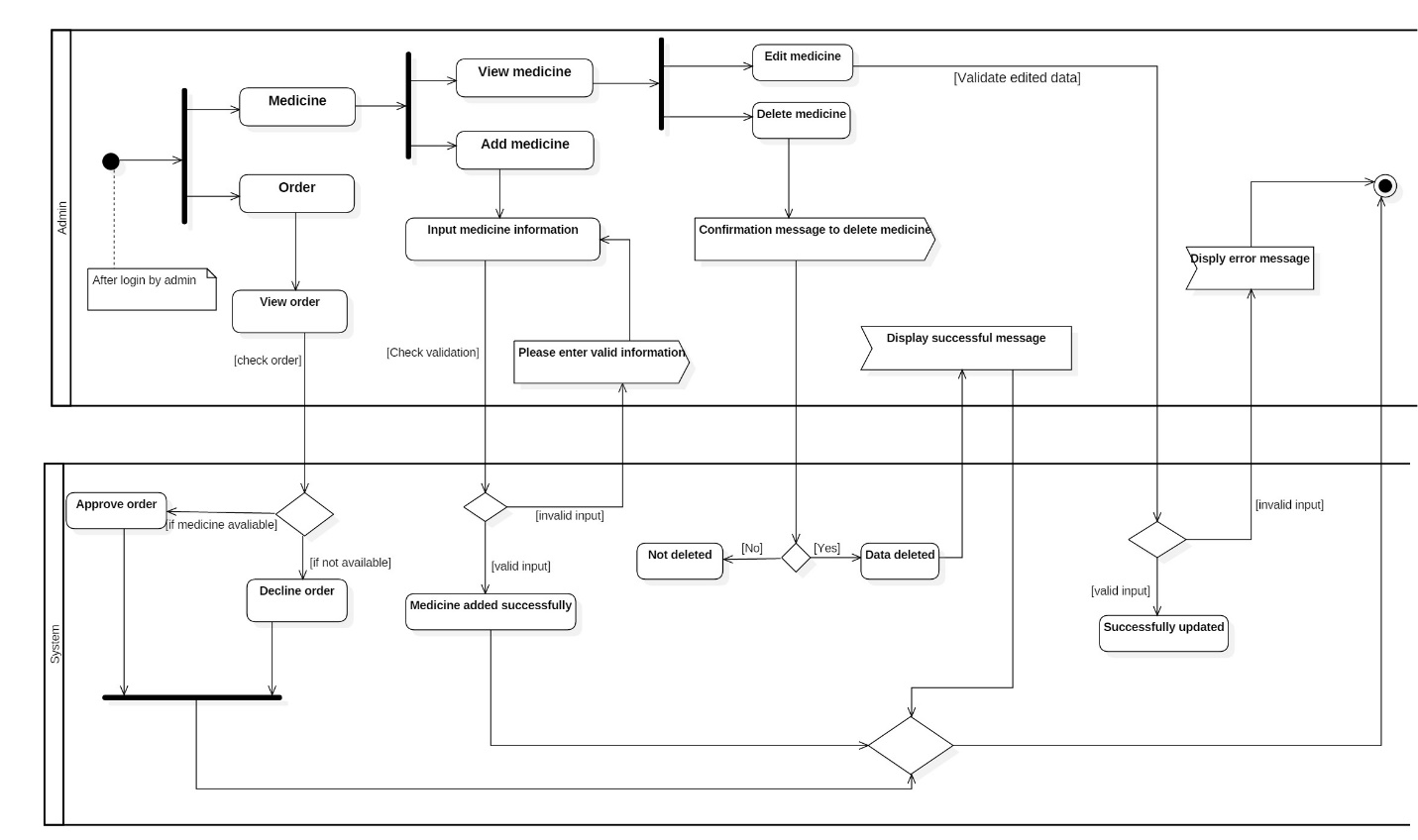
* **Start point:** A small filled circle that indicates the beginning of the process.
* **Activity:** A rectangle with rounded corners that indicates the action.
* **Action flow:** A arrowed line that indicates paths from one activity to another.
* **Decision:** A diamond shape that indicates conditional decision.
* **Merge event:** A diamond shape that merge multiple flow as one.
* **Fork node:** A multiple arrowed line from join that splits single activity into two parallel activities.
* **Join node:** A multiple arrowed line from thick horizontal line that combines two concurrent activities as one.
* **Time event:** An hourglass that indicates that the action flow stops for sometimes.
* **Sent and received signal:** Sent signal consists of a comment to be sent and Received signal indicates that signal is being received.
* **Swimlane:** A horizontal or vertical column that sets related activities within it.
* **End point:** A arrow with filled circle inside another circle which indicates end of action. (Smartdraw, 2019)

#### Login activity



The above diagram shows the activity of user during login process. User should first register in the system providing their details. After registration, they can log on to the system with unique email and password. User should provide their email and password which will be verified by the system. If they are valid user then system will again check their usertype i.e. either they are admin or normal user. And if the user is invalid, an error message is thrown by the system as “Enter valid email and password”. After verifying and checking user, if the valid user is admin then admin dashboard is displayed and if the valid user is normal user then user dashboard is displayed.

#### Admin activity



The above diagram shows the activity of admin’s performance in system. As an admin user, they can manage medicine and order which requires login process before performing these actions.

Admin can manage medicine as adding medicine, modifying them, deleting them and viewing their details. To add product a valid input must be provided for successful storing otherwise an error message is displayed. They can view medicine to edit and delete medicine details. To modify medicine details, a valid data must be input for successful edit. Invalid data input throws an error message. And to delete medicine details, a confirmation message is displayed. If they click “yes” then data will be deleted and if “no” is clicked then data will not be deleted. After deletion a successful message is displayed.

Admin can manage order and view order details. They can approve order and decline order according to products available in stock. If medicine is available, admin will approve the order and if medicine is not available the order will be canceled.

#### Search and order activity



The above diagram shows the activity of user during medicine search and ordering them. General user can search product in our website but to add them in cart and to order them, they have to log in to the system. When user search the medicine by medicine’s name, it will take nearly 5 sec to check the searched product in the database. If it exists in the database then the searched product is displayed to their screen. And if it does not exist then a message is sent as “No data found”. After viewing the searched product, they can add to their cart and can check their cart. They can also update their cart by adding their required quantity of product or they can delete added product in cart. After that they can proceed to order providing their delivery details. They can check out their amount to be paid during delivery time.

### 3.2.2: Sequence diagram

Sequence diagram is an interaction diagram which represents interaction of object arranged in time sequence. It helps to simplify complex system with the sequence of fragments and to understand the sequential flow message exchanged between the objects. (Wikipedia, 2019)

The sequence notation used are as follow;

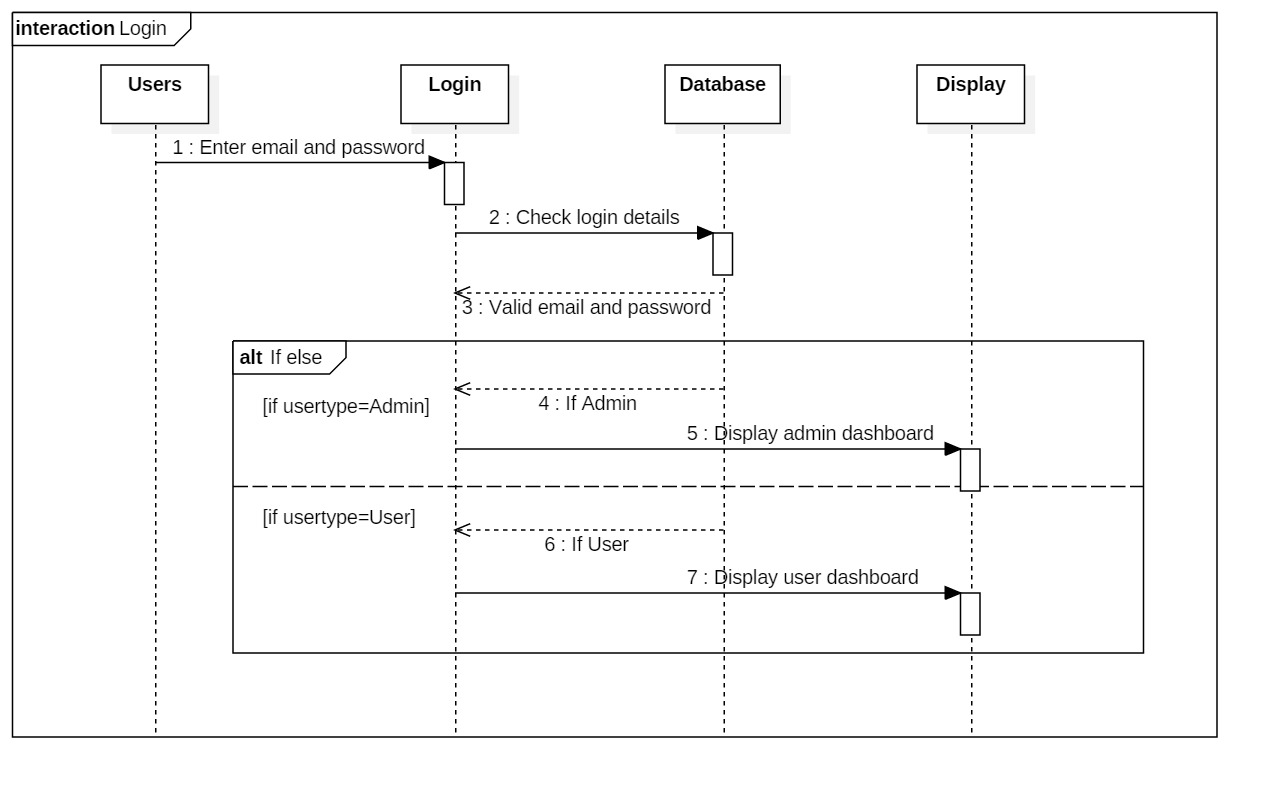
* **Object:** A rectangular box at top that describes the behavior during the system.
* **Activation bar:** A vertical box that represents the time an object is active during interaction between two objects.
* **Lifelines:** A vertical dashed line that represents presence of object overtime.
* **Messages**
* Synchronous message: A forward arrow line that sets communication between two objects which requires a response.
* Return message: A backward dotted arrow line that reply the message.
* Self-message: A U-shaped arrow line that an object sends message to itself.
* **Combined fragments**

A rectangular box with ‘alt’, ‘opt’, ‘loop’ etc.

* Alternatives (alt): It is used when making choice between two or more message.
* Options (opt): It is used to indicate that a sequence occurs under certain condition.
* Loops (loop): It is used to indicate iterative sequence.

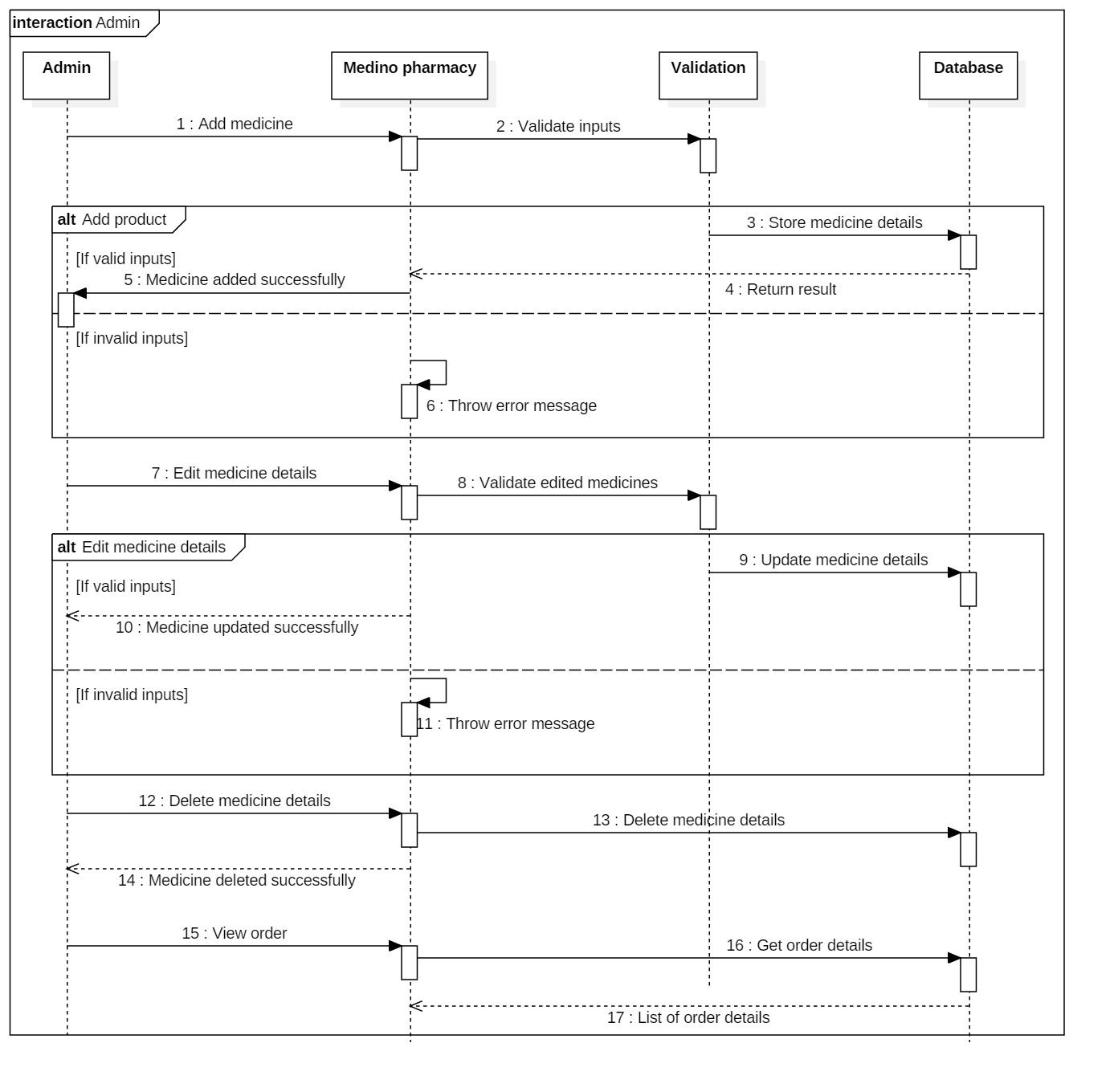
(Smartdraw, 2019)

#### Login sequence



The above sequence diagram shows the sequential flow of login process. User after being registered, login to the system. User interact with the login object providing their email and password which flow a message to check login details in database. After validation, valid user’s usertype will be checked. If a return message as “Admin” comes then admin dashboard is displayed and if “User” is replied then user dashboard is displayed.

#### Admin operation sequence

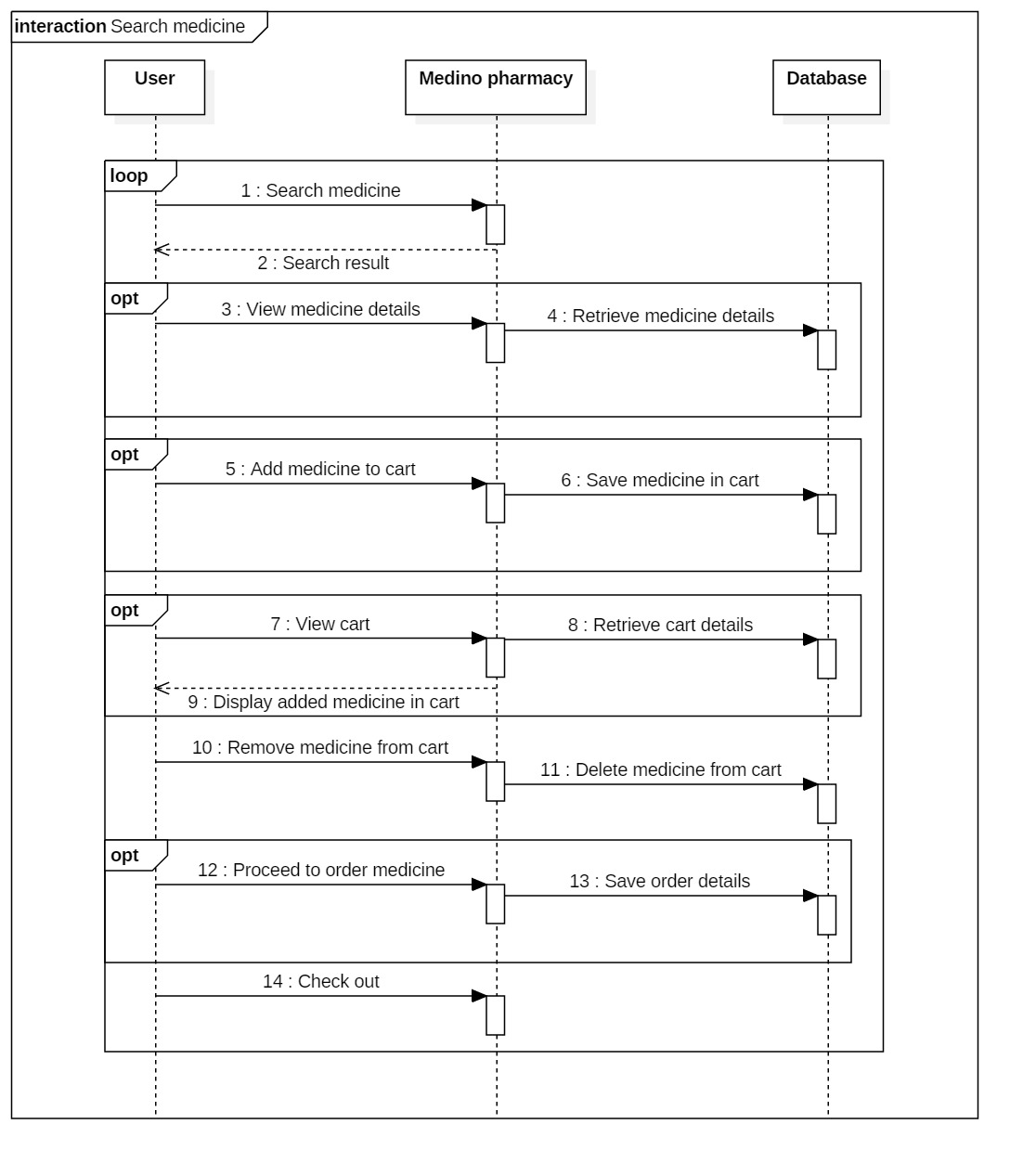


The above diagram is the sequential flow of admin operation in the system. Admin as user can only perform add, update and delete operation in the system. They can also view order made by the customer. After login process, admin can add medicine details which will be stored in database after validation. Admin inputs the medicine details which interact with the validation objects providing message as “Validate the details”. After validation, if data is valid then data gets stored and returns a successful message to admin. And if the details are invalid, a self-message of error is thrown by the system.

For updating medicine details, same sequential flow is performed in the system. And for deleting medicine, admin gives a synchronous message to delete the medicine to the system. System directly interact with the database forwarding a message and a return message is displayed to the user as “medicine deleted successfully”.

To view order, system interact with the database object providing a message as “get order details” which then return message displaying the list of orders.

#### Search and order sequence



The above diagram shows the sequential flow of searching and ordering medicine. User search medicine in medino pharmacy which will interact with the system returning the search result. They can view the medicine details which is retrieved from database. They have various options to add the medicine in cart, view the cart and order them and checkout.

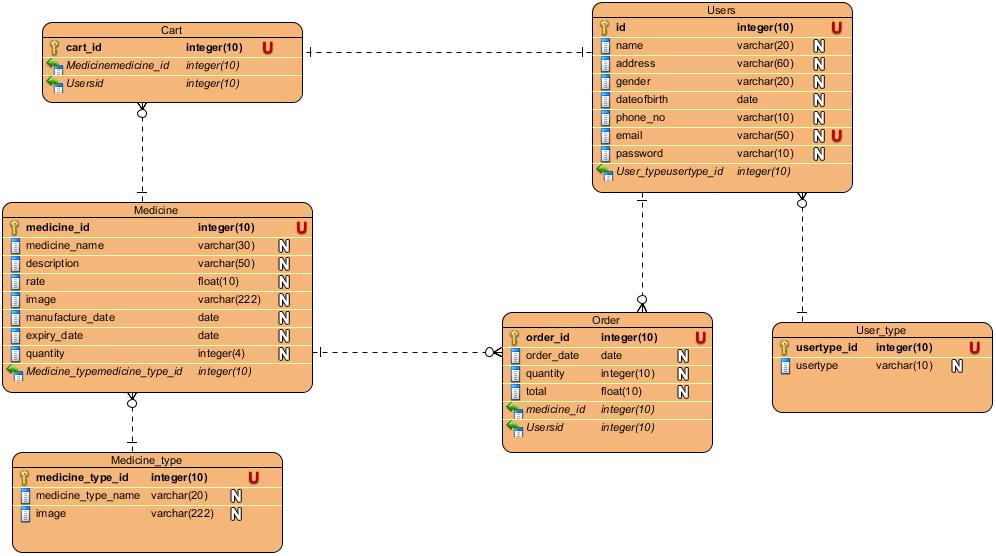
## 3.3: Database Modelling

A model which concerns with the logical structure of a database and defines the structure of data stored, organized and manipulated within database management system.

Entity-relationship model and Data dictionary is represented below.

### 3.3.1: Entity-relationship diagram

Entity-relationship diagram is the analytical structure of entities and their relationships that are stored in database. It is also represented as the physical storage of data. And it provides a visual representation of initial phase of database design. (TechTarget, 2019)



In above ER-diagram, there are total six entities including their respective attributes and constraint. Medicine type and medicine have many to one relationship as one medicine type will have many medicines. User type have one to many relationships with user. Medicine and user have many to many relationships which is linked through order. One medicine can be ordered many times and one user make many orders. A cart will have different medicines. And one user will have only one cart so they have one to one relationship.

### 3.3.2: Data dictionary

Data dictionary is a structured place that contain database’s metadata. Metadata is data about data. And it includes name of tables, name of columns and indexes, constraints and relationship and other related data. It helps to manage and documents database details in efficient way.

There are seven tables in database and they are shown below with their metadata.

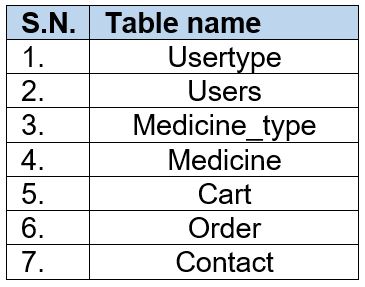
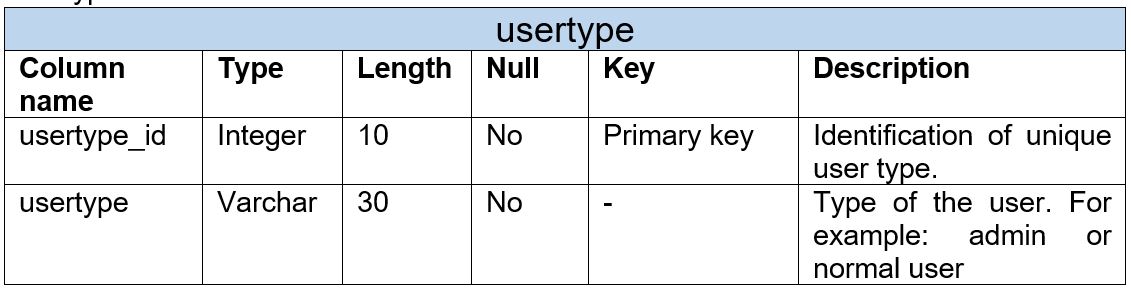
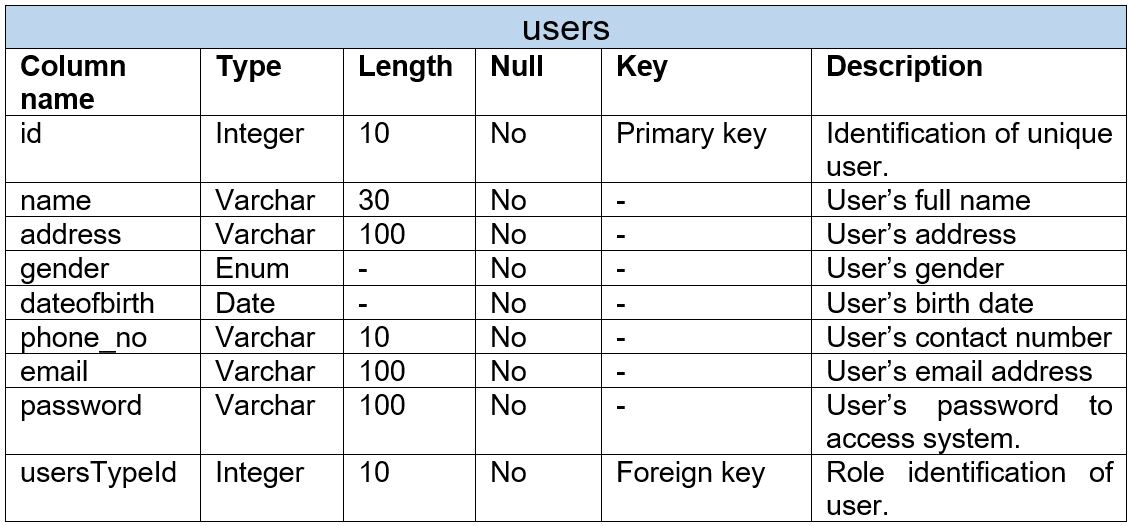


Figure 1: All tables

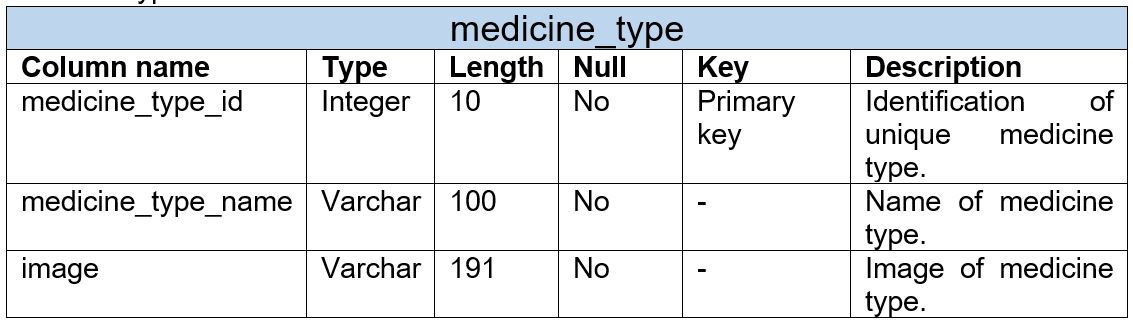
1. Usertype table



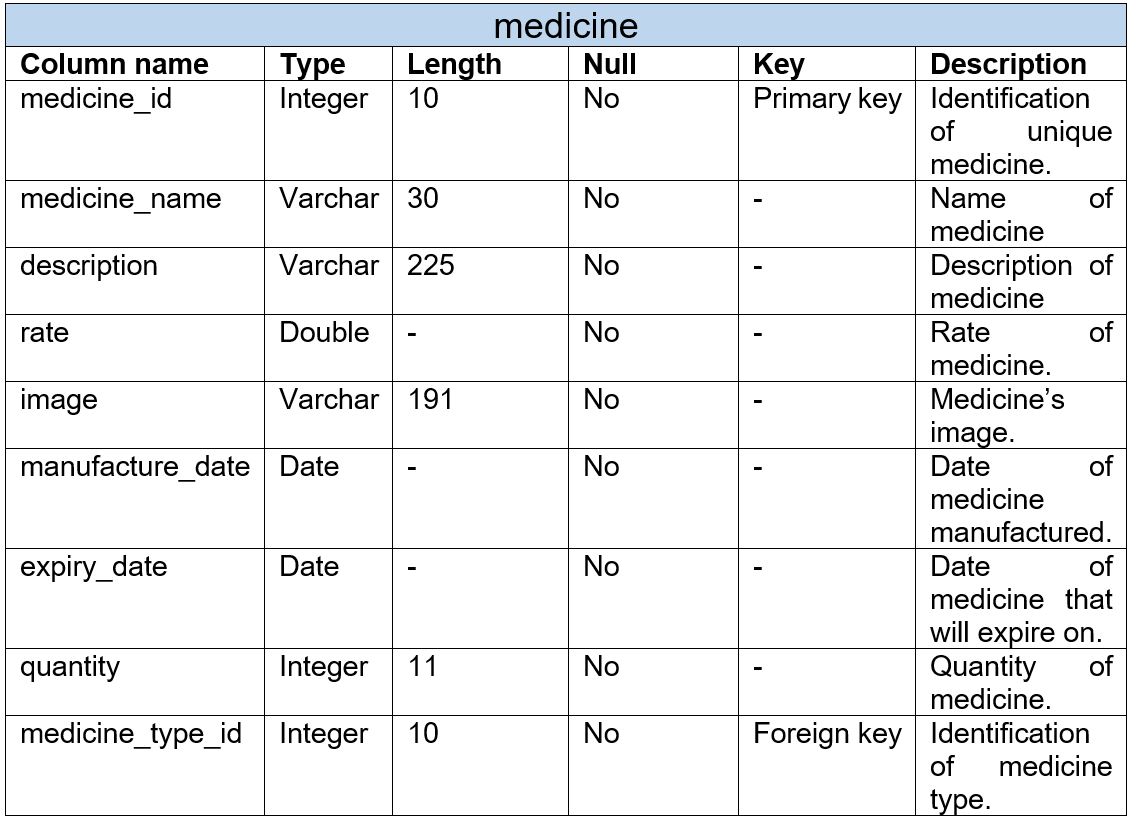
1. Users table



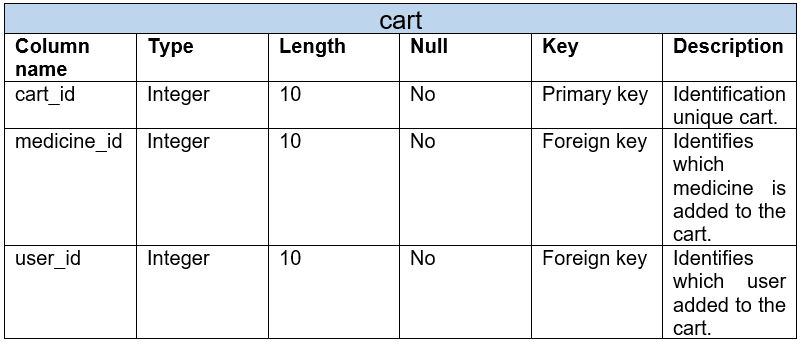
1. Medicinetype



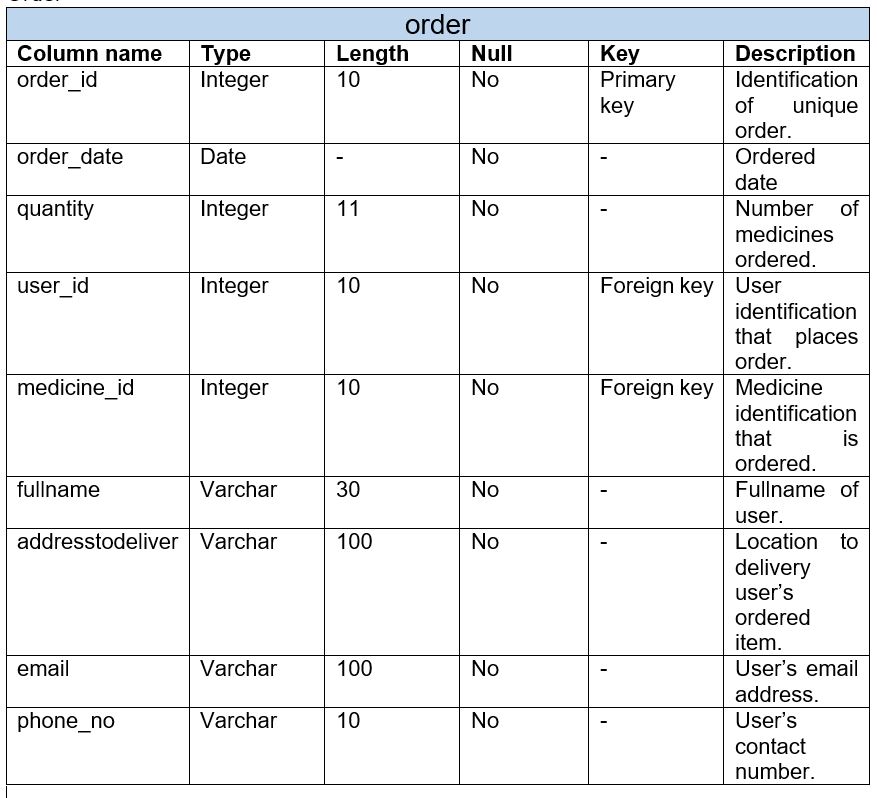
1. Medicine



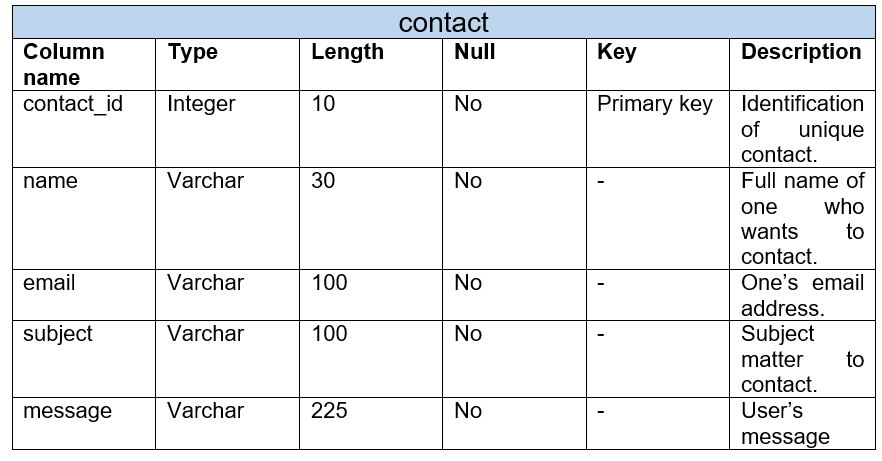
1. Cart



1. Order



1. Contact



## 3.4 UI modelling

UI modelling is a technique to develop a front-end design using computer application. I have used Balsamiq application which is an interface designing tool.

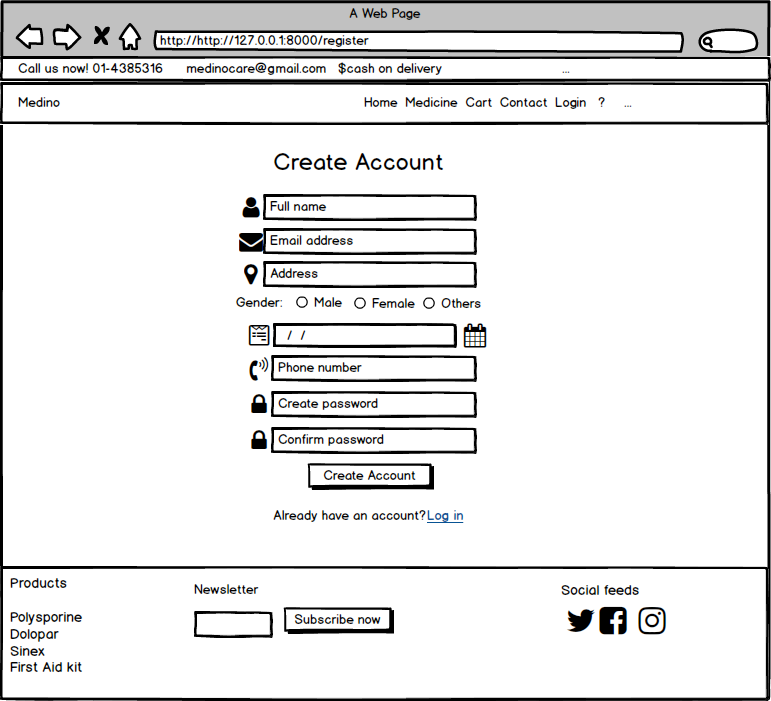
#### 3.4.1: Prototype

Prototype is the early sample of a product to examine the concepts or processes before implementing it. The prototype design for Medino website are as follow;

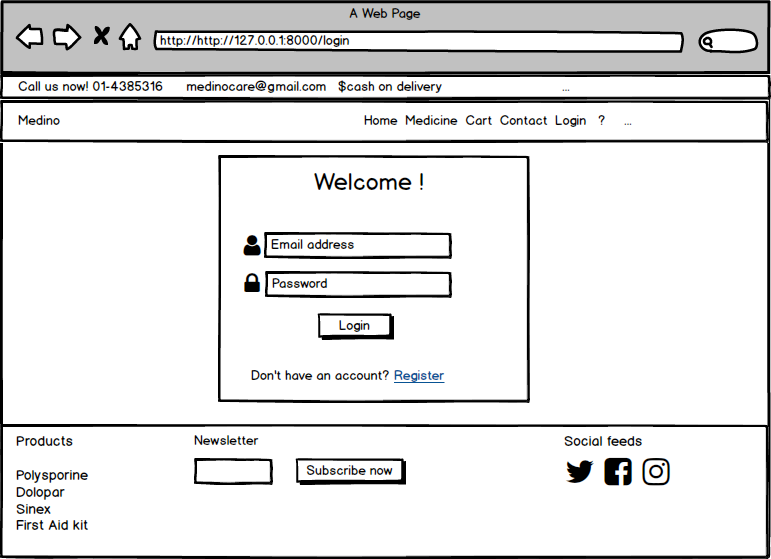
* Welcome page



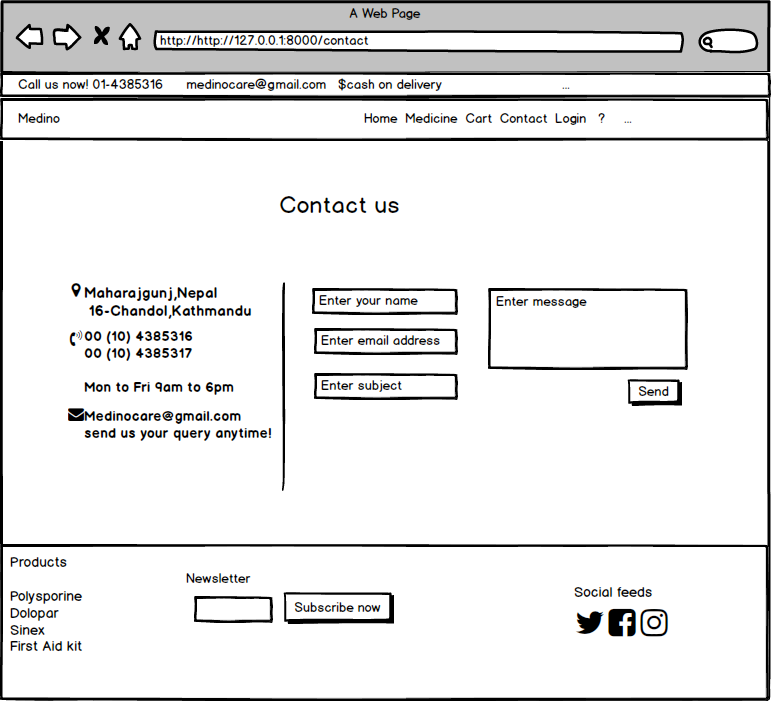
* Registration page



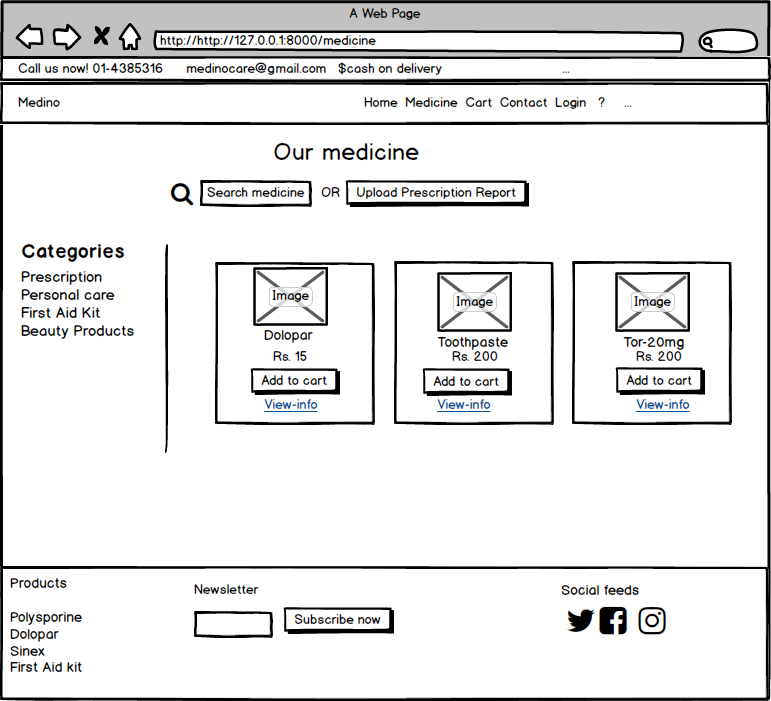
* Login page



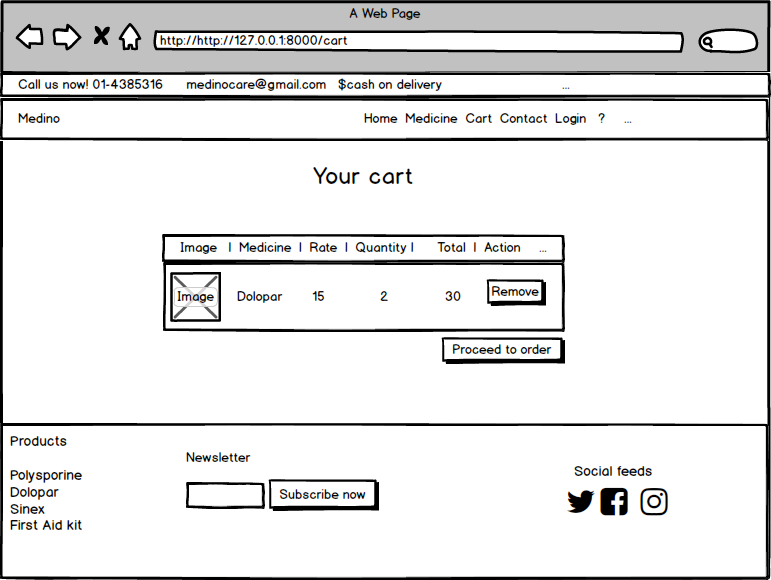
* Contact page



* Medicine page



* Cart page



# **Chapter 4: Implementation**

## 4.1: Introduction

Implementation is the process of converting a design into an actual product where a developer begins to build and code under the use of programming language. It is the phase where client’s idea for the project is implemented in real working environment.

## 4.2: Choice of Language

The purposed website is developed using PHP language. PHP stands for “Hypertext pre-processor” which is a server-side scripting language to develop web application. As PHP is open source and free and is cross platform which makes flexible and easy to achieve the required goal for developing website. (Guru99, 2019)

## 4.3: Development Environment

### Frameworks:

The project is developed in Laravel web application framework. I have used 5.5 Laravel version. Laravel is open source and robust application which includes all majority task required for web like routing, authentication, testing, expressive migration system and other common tasks. It supports MVC architecture which we are going to follow. (Laravel, 2019)

### Standard libraries:

To design the front-end, CSS and JS library file are used. jQuery is used as client-side scripting language which helps in client-side validation.

## 4.4: Development Platform

The website is designed and developed in Windows platform.

## 4.5: IDE

IDE acronyms Integrated Development Environment which is software application that facilitates to develop software.

Visual studio code is best IDE for website development. VS code is inbuilt with various features which makes flexible while working. It includes command line interface, git integration, debugging, integrated terminal, testing and many more. (Scotch, 2019)

## 4.6: Development strategy

It is a way of planning to deploy the developed project into a client environment. It includes different steps and they are:

* Release: Developed system is released in real environment.
* Installation: System is then installed in computer with execution of software component.
* Adapt: System modification that is being installed.
* Update: Replacing older version with new one with fixing of bugs.

## 4.7: System Migration

System migration is the process of transferring an application from one working environment to another in order to gain better commercial value. The physical and logical dependency should be fulfilled during migration and they are:

* Server: A computer program which helps to access service in a network.
* Apache Server: A web-server
* MySQL: A database server
* Composer: A package manager of PHP

Note: Required front-end and back-end designing code is included in Appendix.

# **Chapter 5: Testing**

## 5.1: Introduction

Testing is the process of evaluating the functionality of the software in order to check whether developed software met the actual requirement or not (Tutorial point, 2019). It identifies defects and errors made during development phase and ensures that the system is defect free. Also, it measures the quality of the product developed.

And testing is done in order to make sure customer’s reliability and their satisfaction with our developed software.

There are different types of testing to make sure it works as expected. Black box testing, Unit testing, White box testing, Acceptance testing etc. are types of testing. And among them I have performed **Unit testing** and **Black box testing** with test planning.

Test case or Test plan: It is the documentation of specific test with objectives, resource and process of testing.

### 5.2: Unit Testing

Unit testing is the process of isolating a section of source code and executing each individual unit of source code to determine its correctness (Guru99, 2019). It is performed during the development of application.

Advantages of unit testing;

* Fix bugs in early development cycle.
* Improves the quality of code.
* Reduces defects and bugs.
* Less costly.

### 5.3: Black box testing

Black box testing is the process of examining the functionality of the system through user interface without knowing the working of internal structure (Guru99, 2019). It only focuses on inputs and outputs of the software system.

Advantages of black box testing;

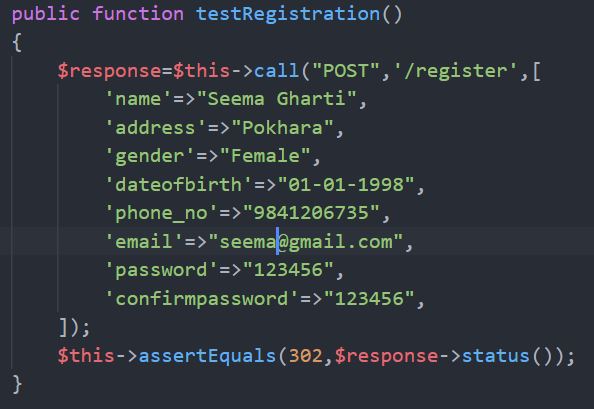
* Validates the functional requirement.
* Test can be done through end user’s point of view.
* No need of technical person for testing.

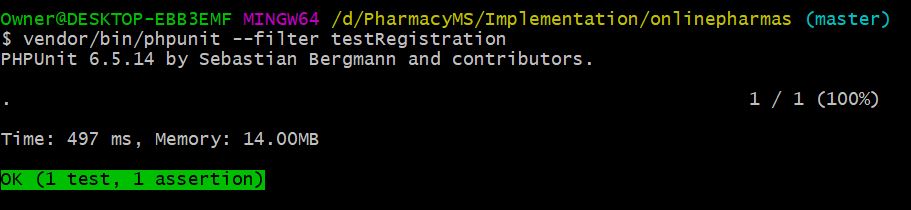
Here, 20 unit and black box testing with its test case for Medino pharmacy is show below;

1. User registration

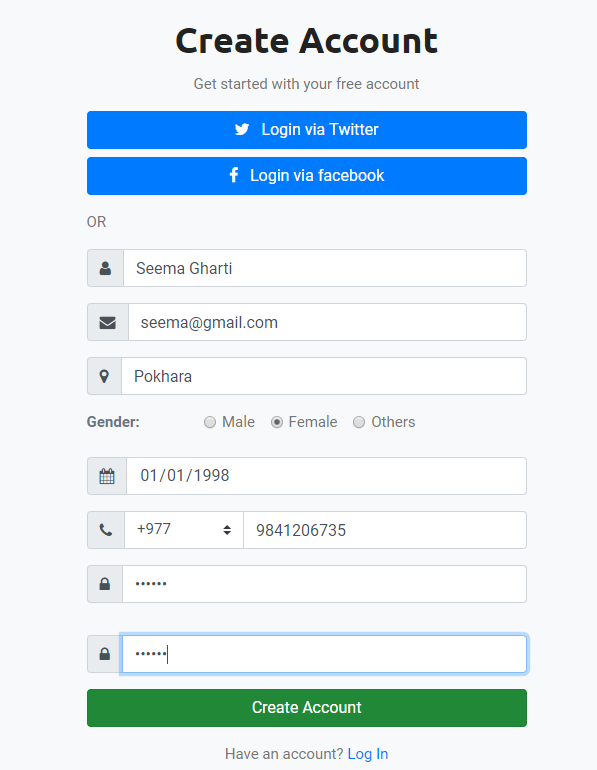
|  |  |
| --- | --- |
| **Test case no.** | **T1** |
| Purpose of the case | Checking whether user can register or not. |
| Test data | name: Seema Gharti  address: Pokhara  gender: Female  dateofbirth:01-01-1998  phone\_no: 9841206735  email: [seema@gmail.com](mailto:seema@gmail.com)  password: 123456  confirmpassword: 123456 |
| Class name | pharmacyTest |
| Function name | testRegistration |
| Expected result | Successful registration |
| Actual result | User added successfully |
| Conclusion | **Yes,** expected result matched with actual result. |

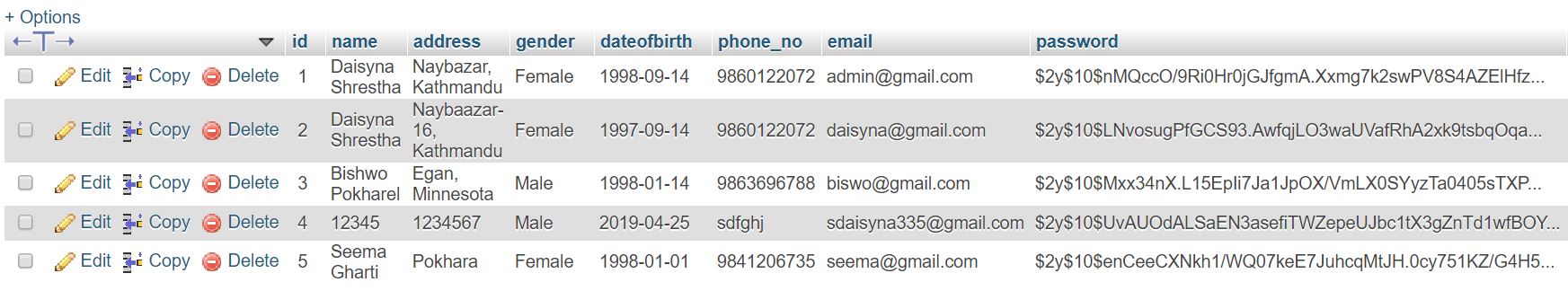
Unit testing:





Black box Testing:



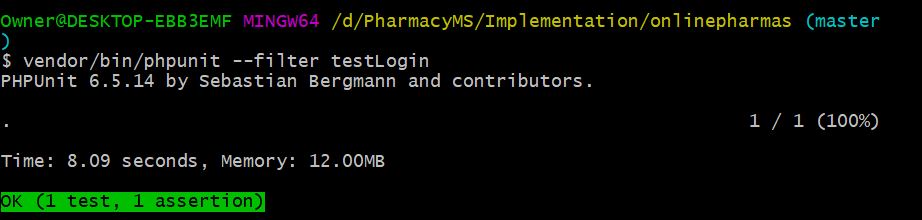


New user successfully added.

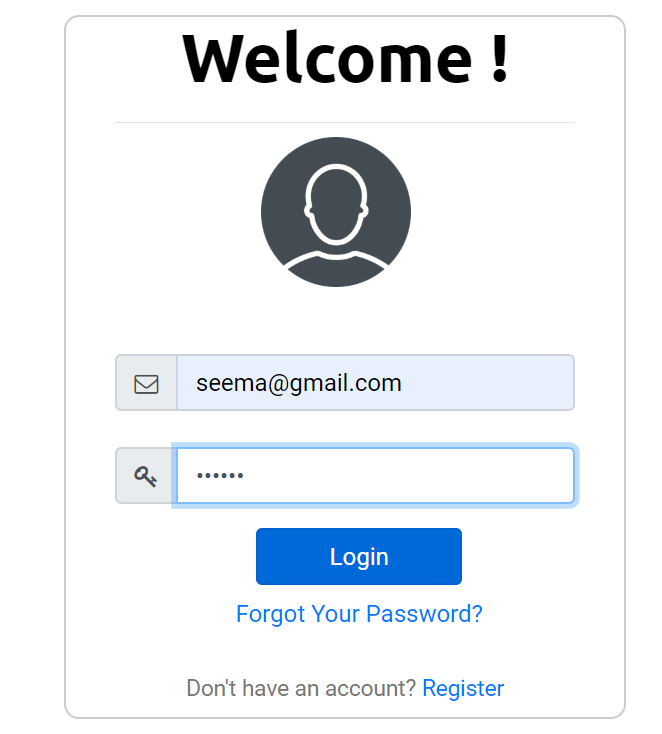
1. Login

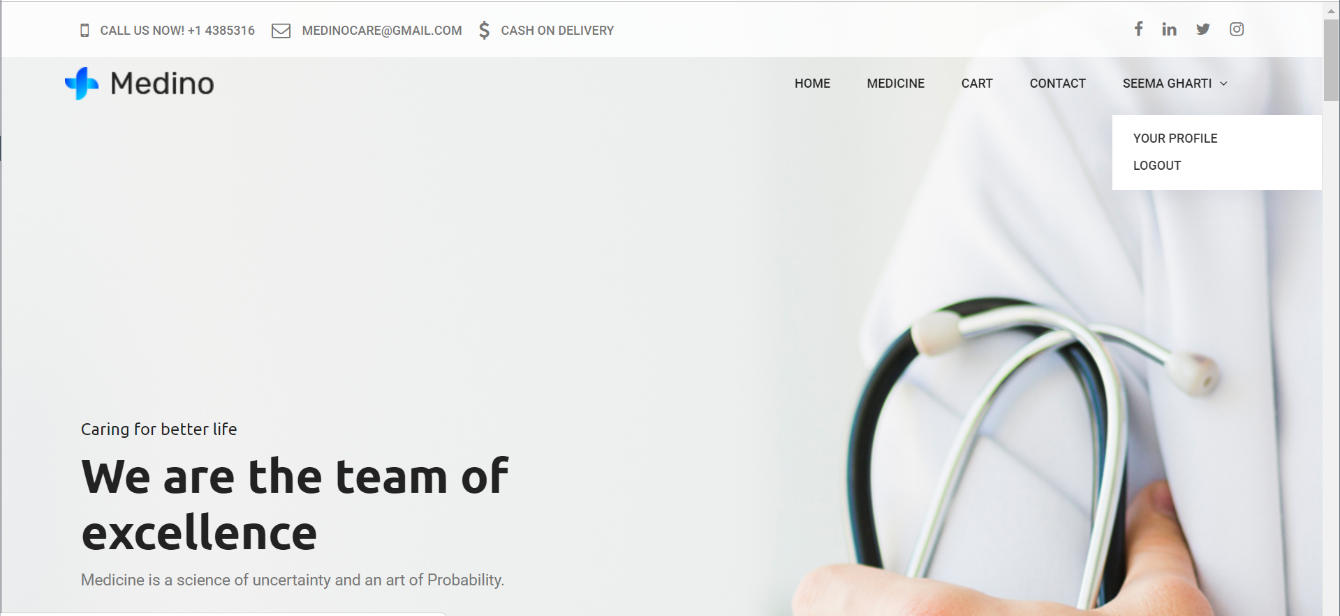
|  |  |
| --- | --- |
| **Test case** | **T2** |
| Purpose of the case | Login to the system by retrieving email and password. |
| Test data | email: seema@gmail.com  password: 123456 |
| Class name | pharmacyTest |
| Function name | testLogin |
| Expected result | User login to the system |
| Actual result | Successful login |
| Outcome as Expected | **Yes,** expected result matched with actual result. |

 Unit Testing:



Black box Testing:



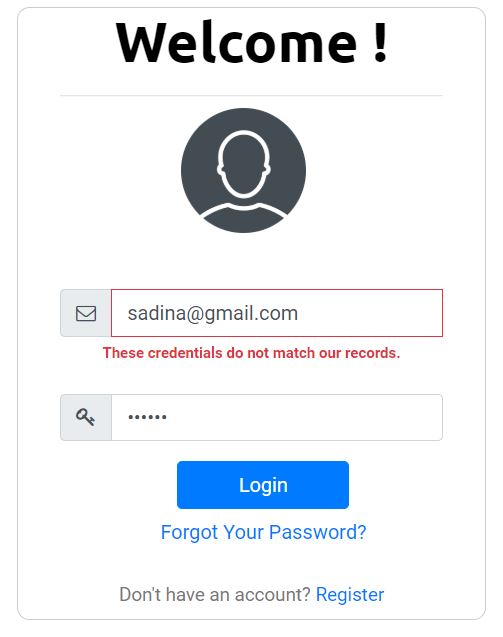


User logged in successfully.

1. Invalid user

|  |  |
| --- | --- |
| **Test case no.** | **T3** |
| Purpose of the case | To check whether invalid user can login or not. |
| Test data | email: sadina@gmail.com  password: sadina |
| Expected result | Throw error message. |
| Actual result | Error message shown successfully. |
| Conclusion | **Yes,** expected result matched with actual result. |

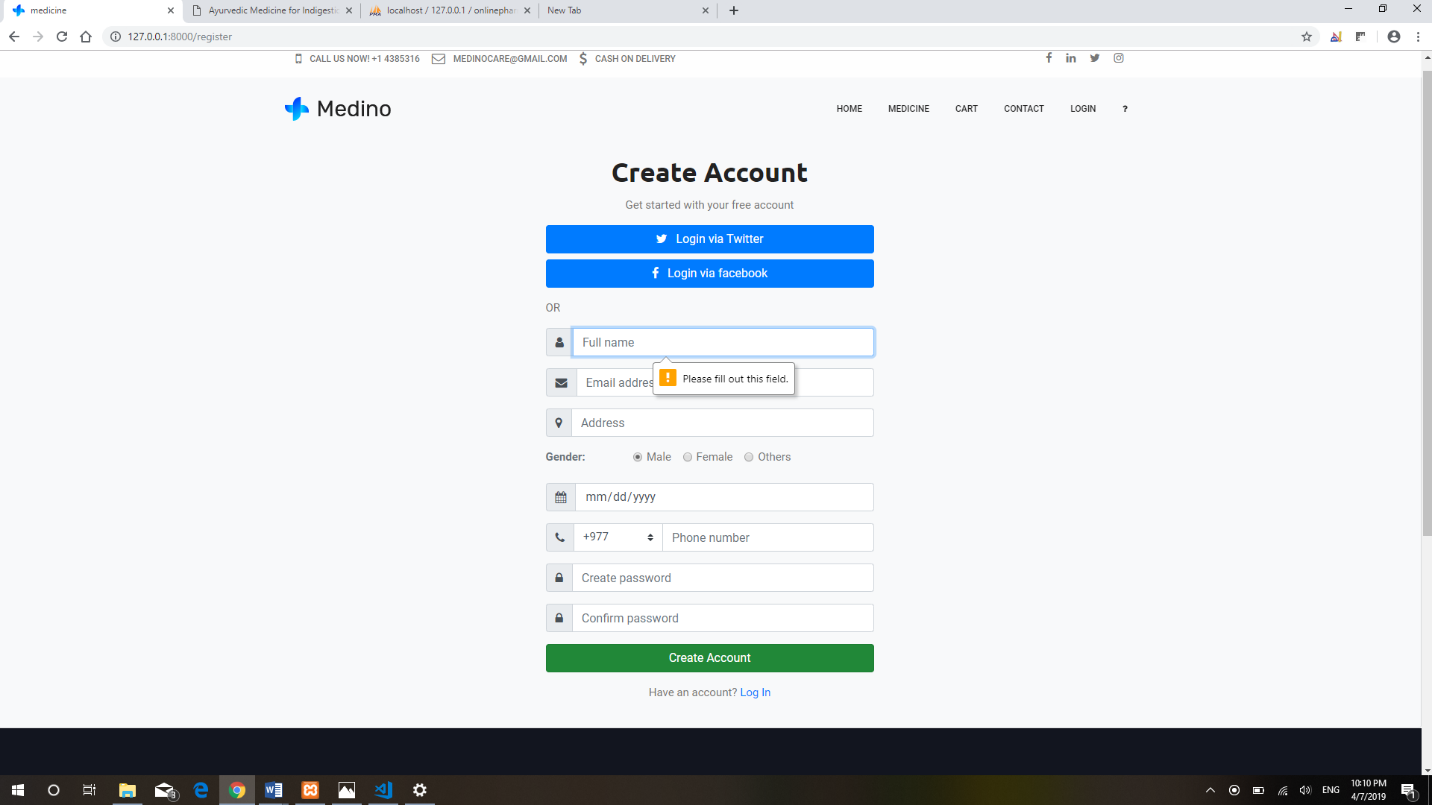
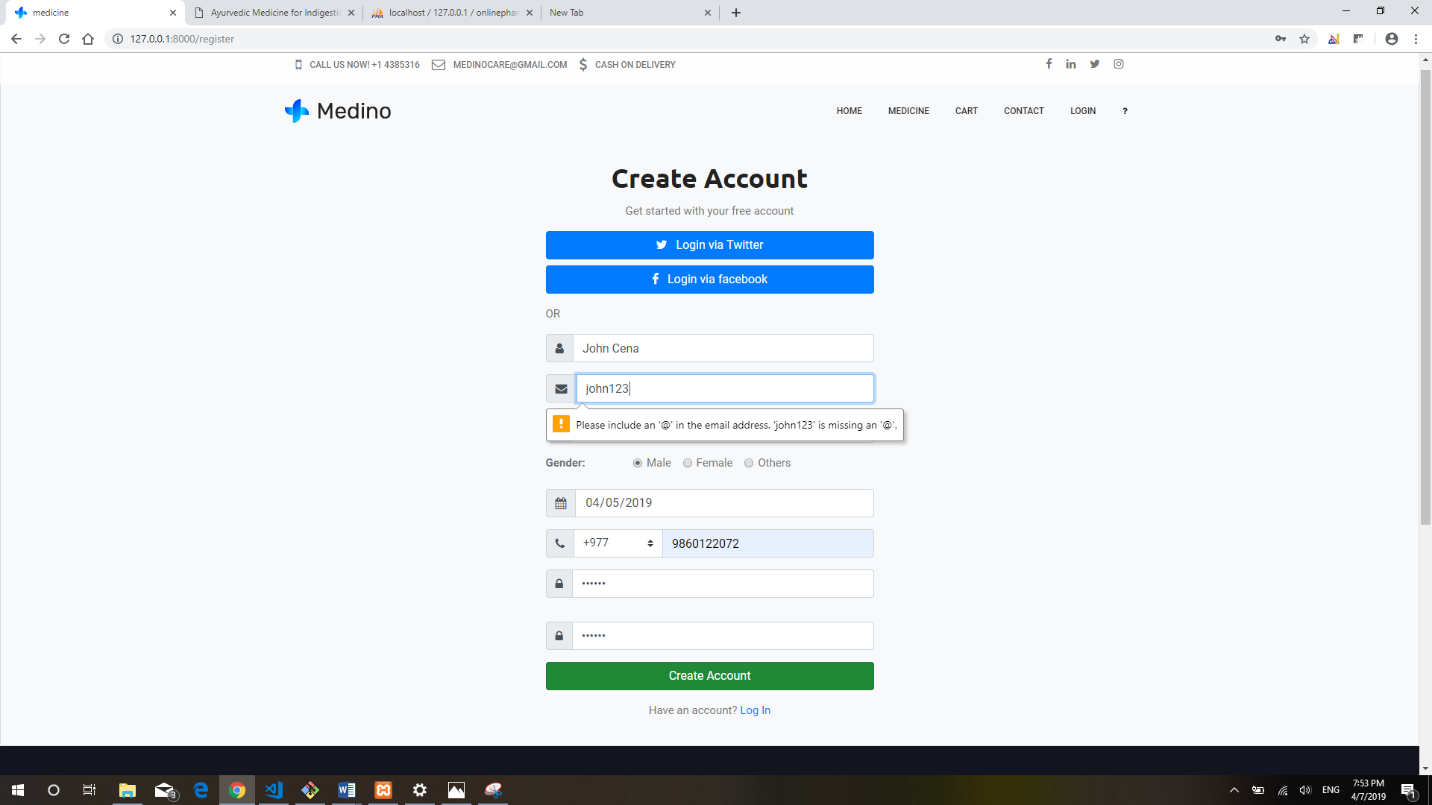
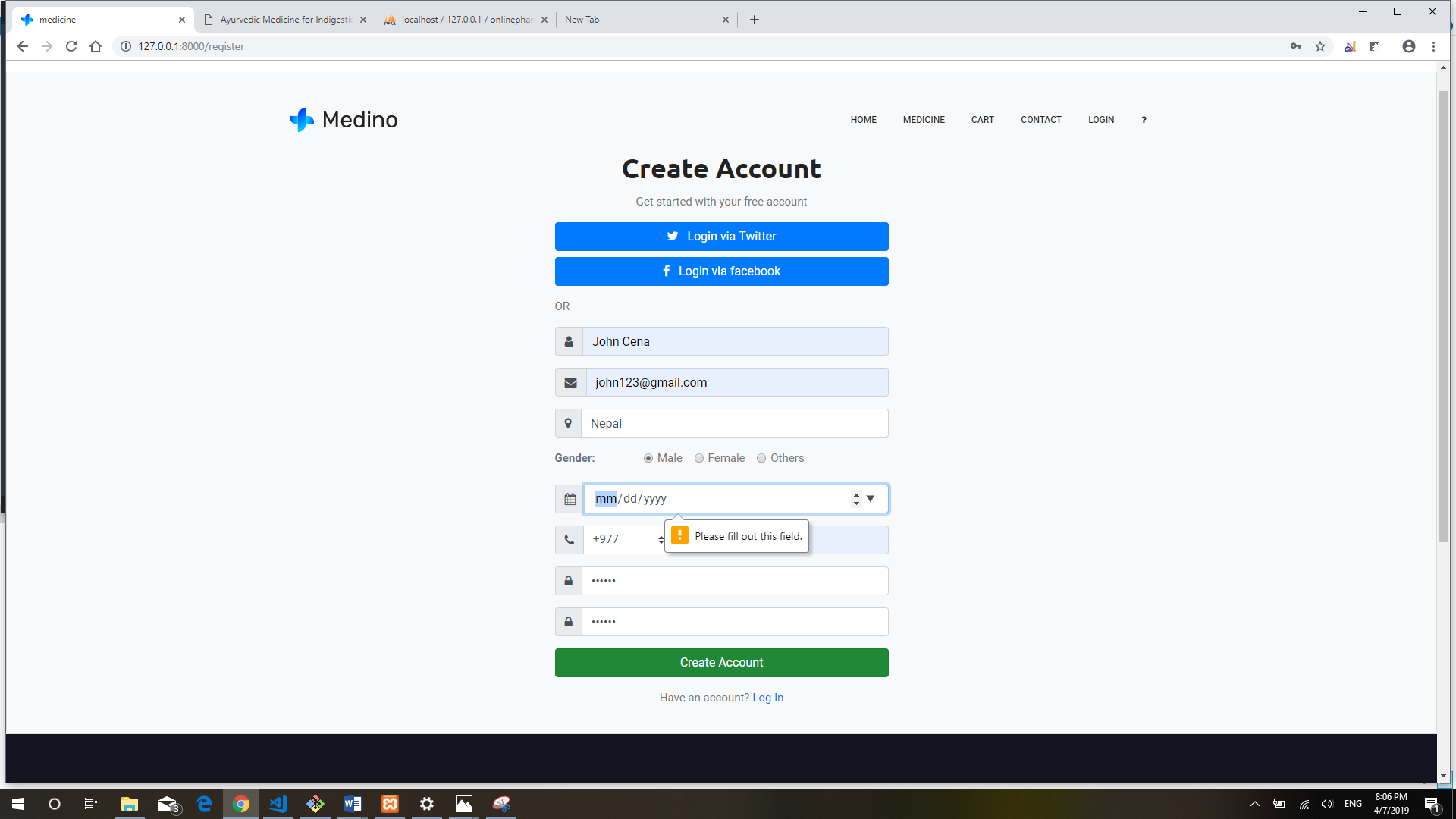
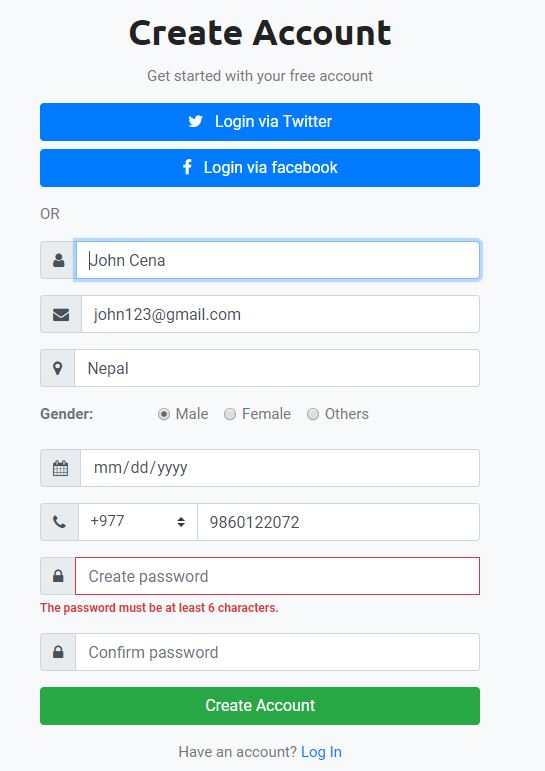
Black box Testing:

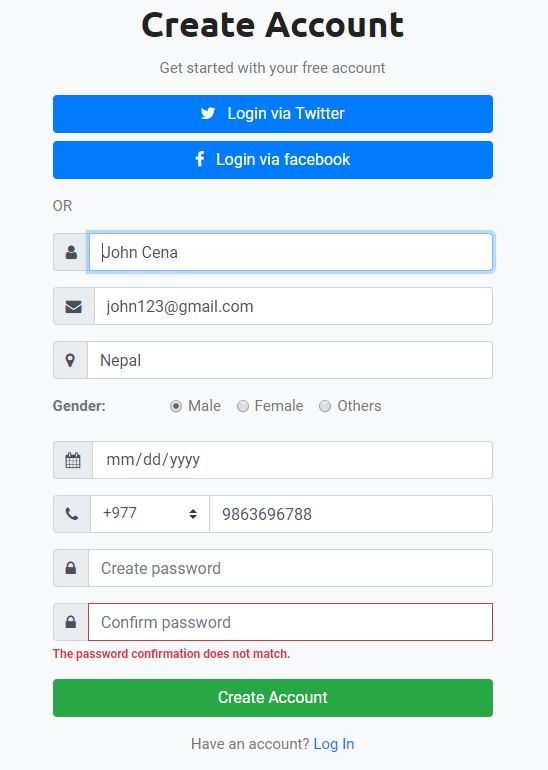


1. Validation of registration form

Black box Testing:

|  |  |
| --- | --- |
| **Test case no.** | **T4** |
| Purpose of the case | To test validation of registration page. |
| Test task | 1. User tries to register with empty field. 2. User doesn’t provide correct email format. 3. User tries to register with one field empty. 4. User enter less than 6-character password. 5. User enter different password and confirm password. |
| Expected result | 1. Throw error message “Please fill out this field”. 2. Throw error message “Please include an @ in email”. 3. Throw error message “Please fill out this field”. 4. Throw error message. 5. Throw error message. |
| Actual result | Error message shown successfully. |
| Conclusion | **Yes,** expected result matched with actual result. |

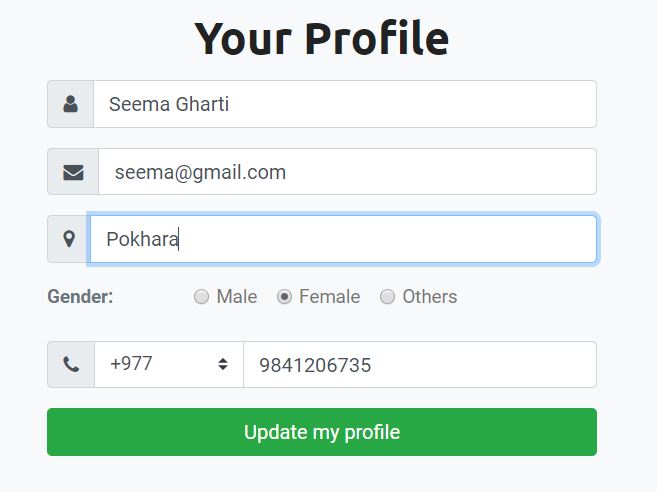


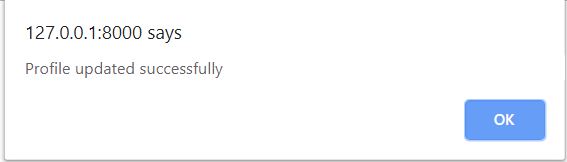


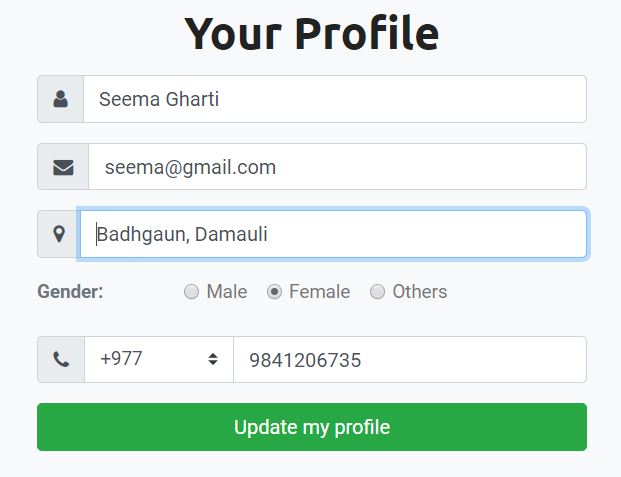
1. Edit profile

|  |  |
| --- | --- |
| **Test case no.** | **T5** |
| Purpose of test case | To check user can update their profile or not. |
| Test task | User change their profile details. |
| Expected result | Throw success message and update data in form. |
| Actual result | Successful message shown and profile updated. |
| Conclusion | **Yes,** expected result matched with actual result. |

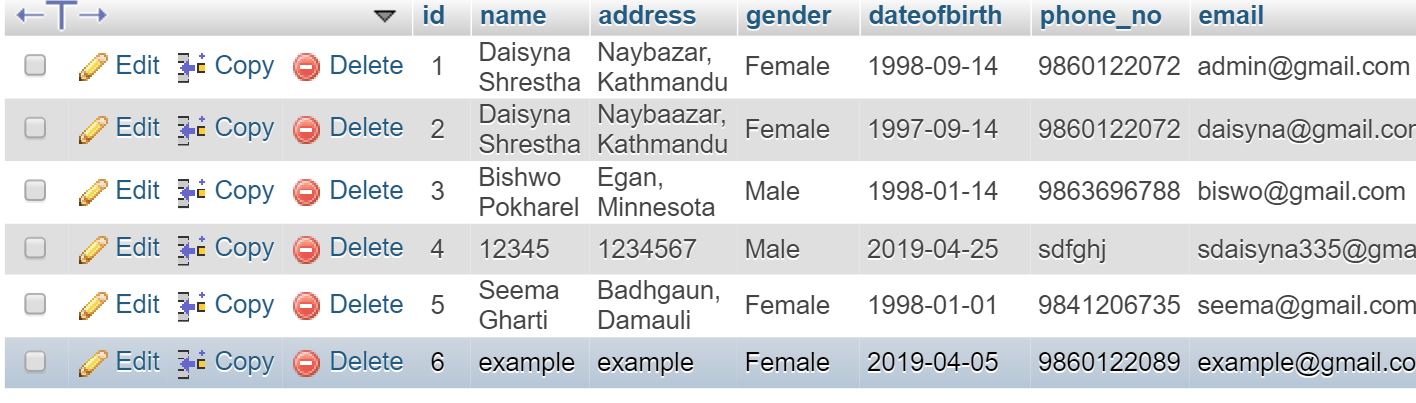
Black box testing:







Address updated.

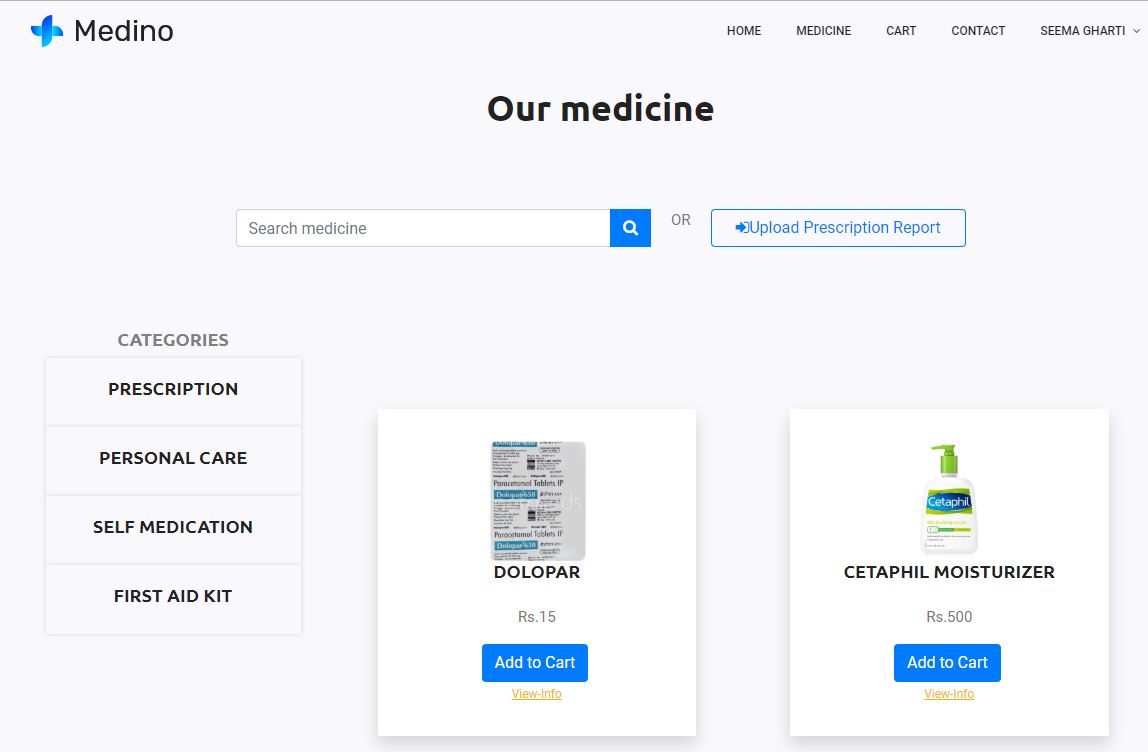


Data updated in database.

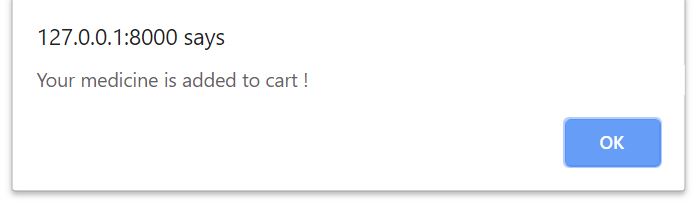
1. Add to cart

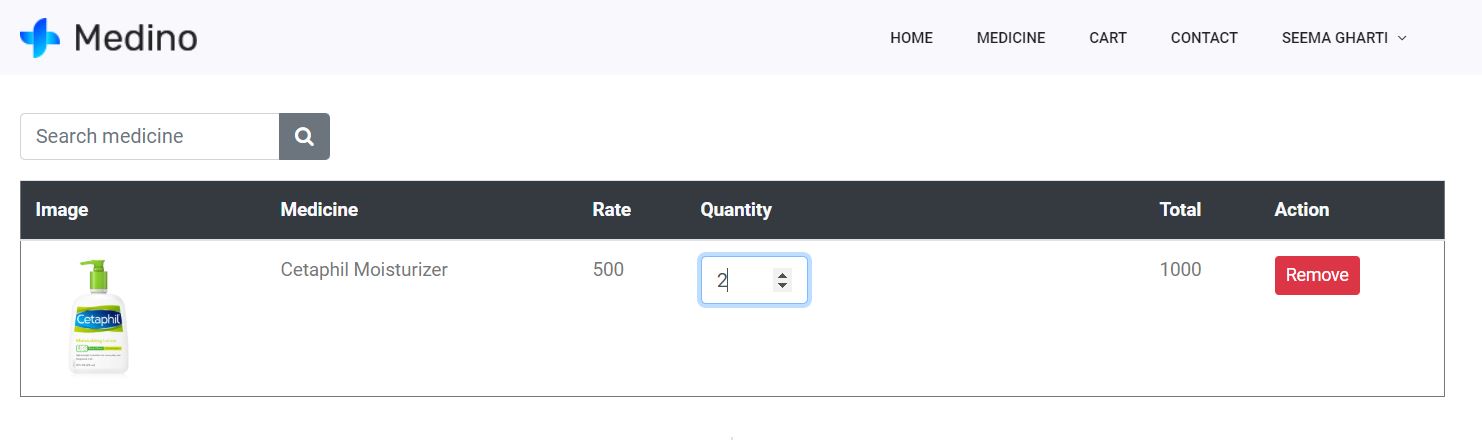
|  |  |
| --- | --- |
| **Test case no.** | **T6** |
| Purpose of test case | To check add to cart by user. |
| Test task | Addition of medicine in user’s respective cart. |
| Expected result | Throw success message and medicine must add to user cart. |
| Actual result | Successful message shown and medicine is successfully added in cart. |
| Conclusion | **Yes,** expected result matched with actual result. |

Black box Testing:



This product is added to Seema’s cart.

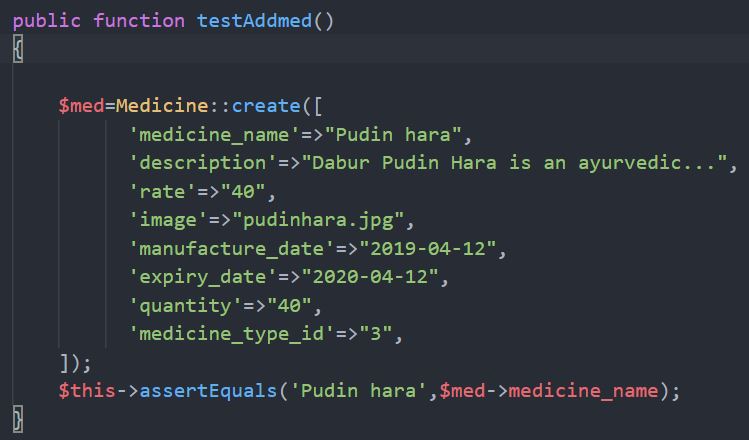


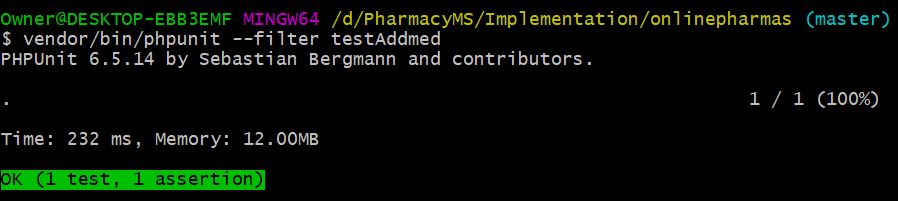


1. Add medicine

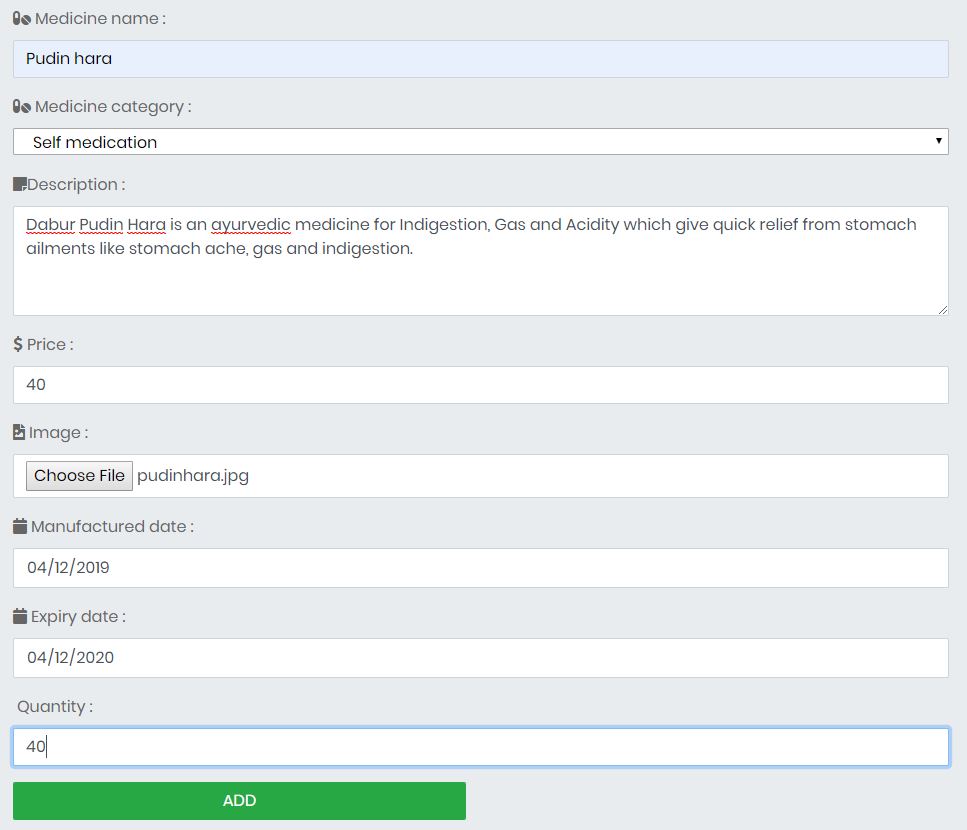
|  |  |
| --- | --- |
| **Test case no.** | **T7** |
| Purpose of test case | To check add medicine. |
| Test data | Medicine name: Pudin hara  Medicine category: Self medication  Description: Dabur Pudin hara is auyurvedic……  Price: 40  Image: pudinhara.jpg  Manufactured date: 04-12-2019  Expiry date: 04-12-2020  Quantity: 40 |
| Class name | pharmacyTest |
| Function name | testAddmed |
| Expected result | Throw successful message and medicine added in table. |
| Actual result | Successful message shown and medicine added. |
| Conclusion | **Yes,** expected result matched with actual result. |

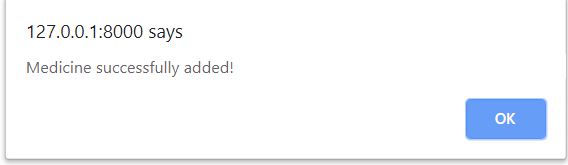
Unit Testing:

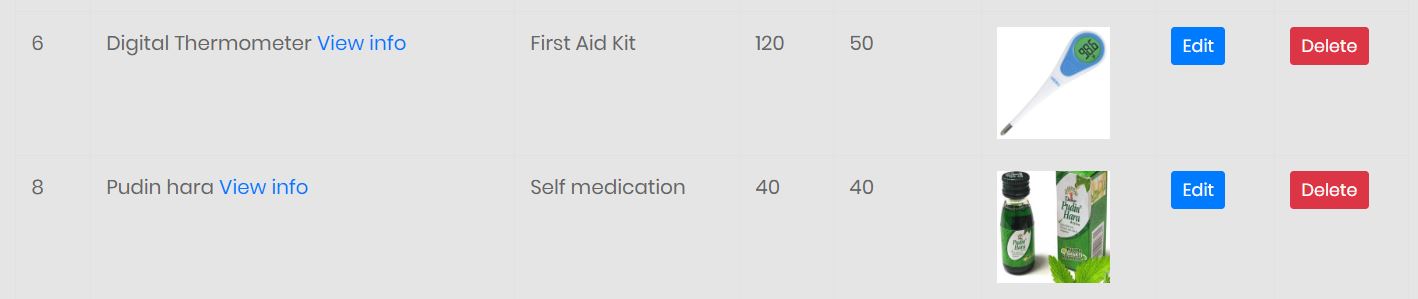




Black box Testing:



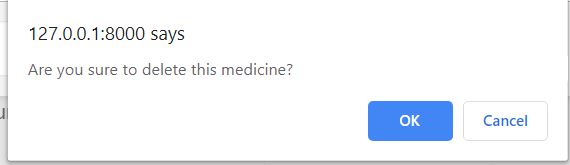
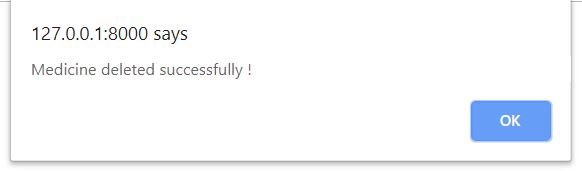


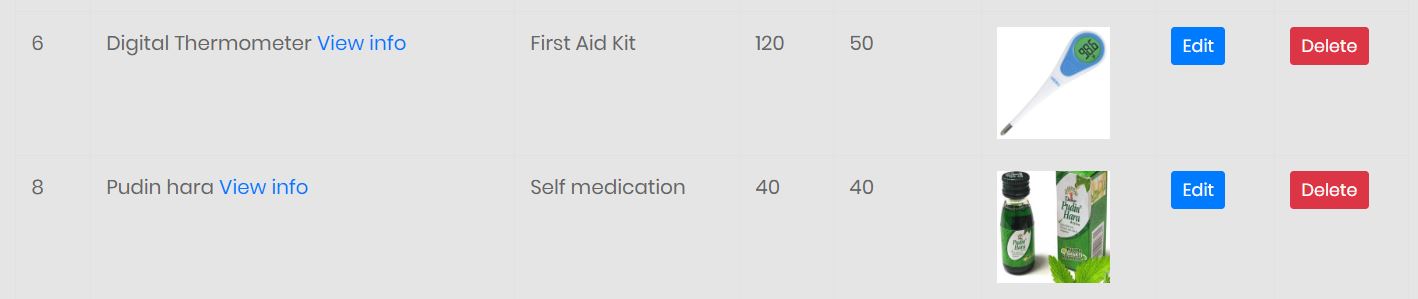


Pudin hara added successfully.

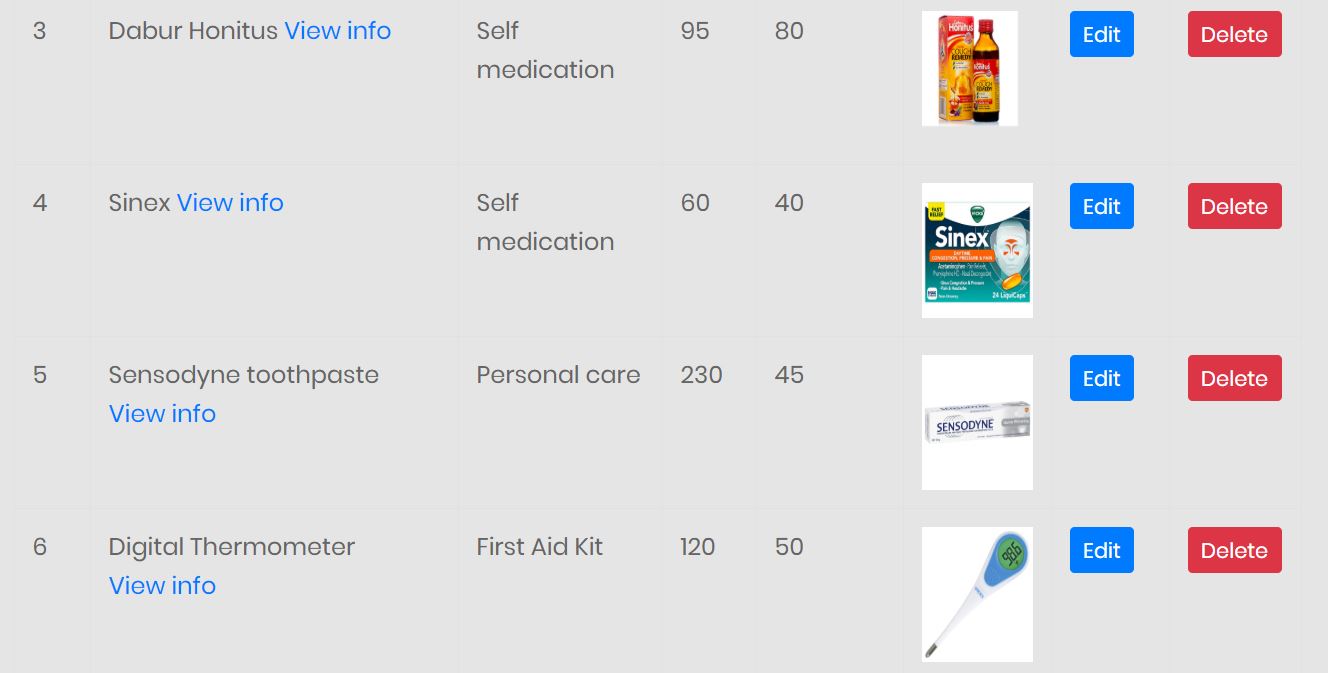
1. Delete medicine

|  |  |
| --- | --- |
| **Test case no.** | **T8** |
| Purpose of test case | To check medicine deletion |
| Test data | medicine\_id: 8 |
| Expected result | Throw confirmation message and success message. |
| Actual result | Successful message shown with deletion of medicine. |
| Conclusion | **Yes,** expected result matched with actual result. |





Click delete button to delete product

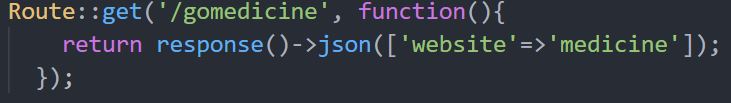


No data of Pudin hara

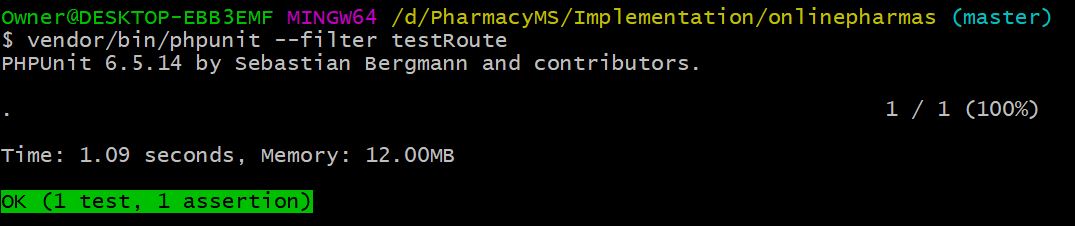
1. Route

|  |  |
| --- | --- |
| **Test case no.** | **T9** |
| Purpose of the case | To check response from the original route. |
| Test data | Route: ‘/gomedicine’ |
| Class name | pharmacyTest |
| Function name | testRoute |
| Expected result | Return same array response from origin route. |
| Actual result | Returning same array response. |
| Conclusion | **Yes,** expected result matched with actual result. |

Unit Testing:



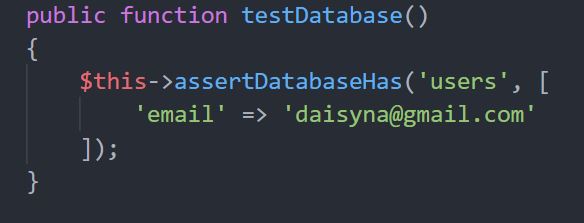


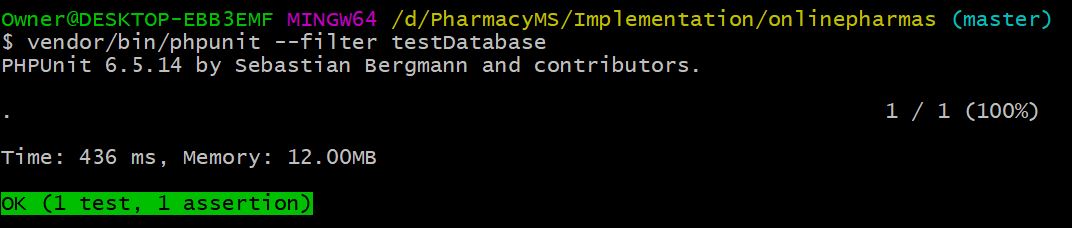


1. Database

|  |  |
| --- | --- |
| **Test case no.** | **T10** |
| Purpose of the case | To check whether data exists in database or not. |
| Test data | table: users  email: daisyna@gmail.com |
| Class name | pharmacyTest |
| Function name | testDatabase |
| Expected result | Data passed through function should match the data stored in database. |
| Actual result | Successful data matching. |
| Conclusion | **Yes,** expectedresult matched with actual result. |

Unit Testing:

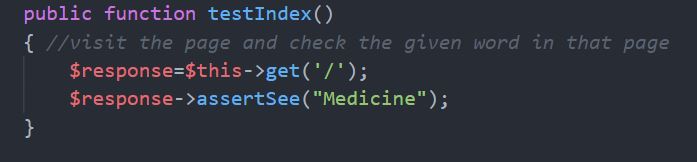


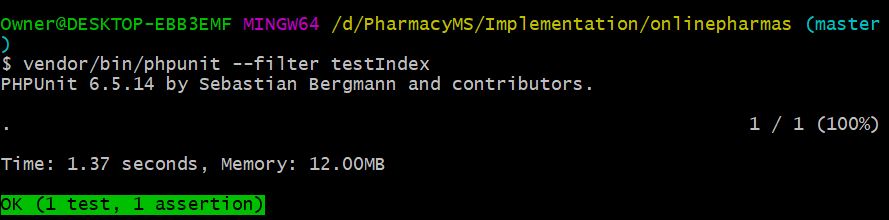


1. Visit page

|  |  |
| --- | --- |
| **Test case** | **T11** |
| Purpose of the case | To visit the page and check the given word in that page. |
| Test data | Route: /  Keyword: Medicine |
| Class name | pharmacyTest |
| Function name | testIndex |
| Expected result | Keyword passed through function should match the keyword in that page. |
| Actual result | Successful visiting and checking passed keyword. |
| Conclusion | **Yes**, expected result matched with actual result. |

Unit Testing:



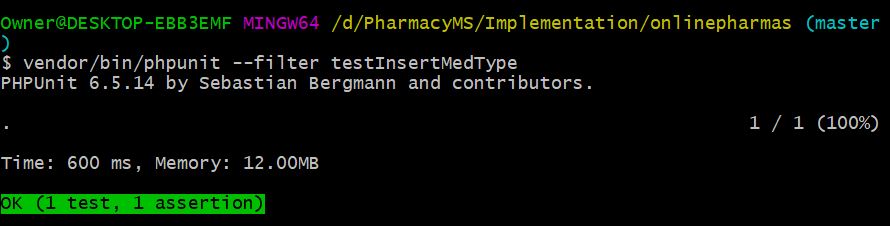


1. Add medicine type

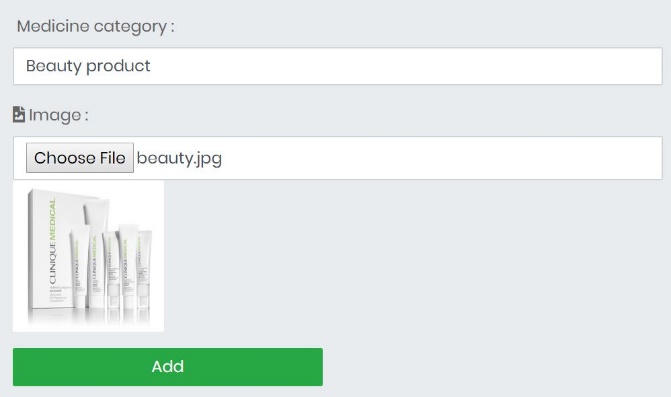
|  |  |
| --- | --- |
| **Test case no.** | **T12** |
| Purpose of the case | To check add medicine type. |
| Test data | Medicine\_type\_name: Beauty Product |
| Class name | pharmacyTest |
| Function name | testInsertMedType |
| Expected result | Throw success message and category should add in table. |
| Actual result | Successful message shown and added in table. |
| Conclusion | **Yes,** expected result matched with actual result. |

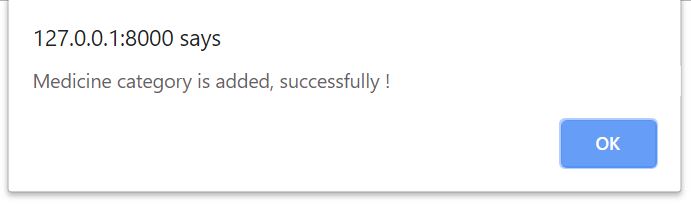
Unit Testing:

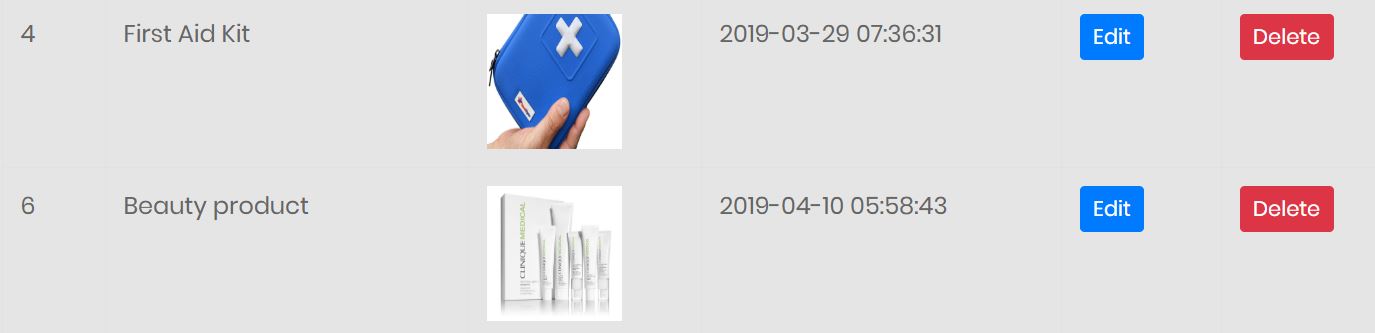




Black box Testing:





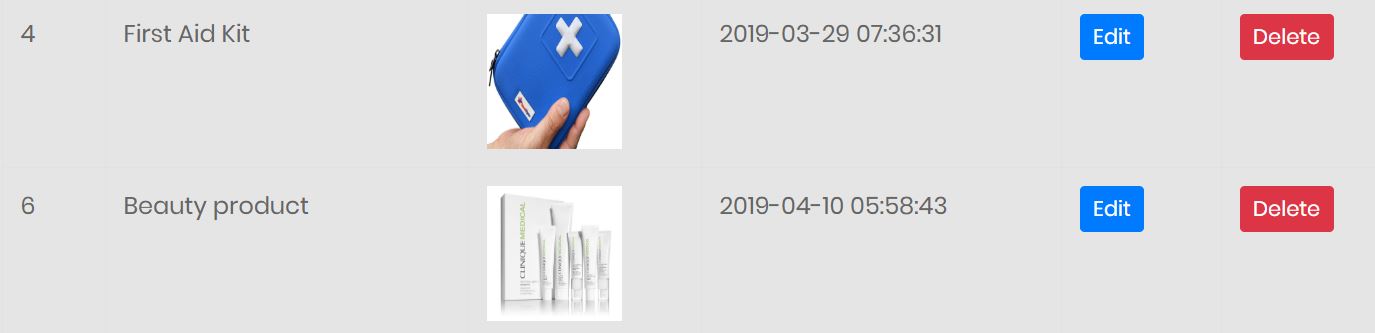


Medicine category is added successfully!

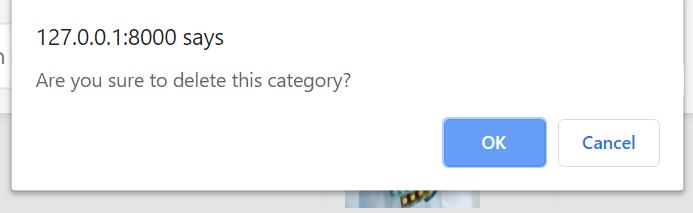
1. Delete medicine category

|  |  |
| --- | --- |
| **Test case no.** | **T13** |
| Purpose of the case | To check medicine type deletion. |
| Test data | medicine\_type\_id:6 |
| Class name | pharmacyTest |
| Function name |  |
| Expected result | Throw confirmation message and category should be deleted from database. |
| Actual result | Successful deletion. |
| Conclusion | **Yes,** expected result matched with actual result. |

Black box Testing:



Click delete button to delete category.



Click OK to confirm deletion



No data of Beauty product in database.

# **Chapter 6: Other project issues**

## 6.1: Risk Management

Risk management is the identification and prioritization of risks and analyzing with various tools and economical application resources to minimize their impact on the system. It also identifies the possible problem before it arises so as to plan the risk-handling activities. Vulnerability assessment is also another term for risk management.

Risk can be managed in following five steps:

1. Identify the risk
2. Analyze the risk
3. Evaluate or rank the risk
4. Treat the risk
5. Monitor and review the risk

And risk can be controlled by identifying their likelihood, consequences and impact on the system. To analyze impact of risk, I have used following formula;

Impact = Likelihood \* Consequence

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N. | Risks | Likelihood | Consequences | Impact | Solution |
|  | Denial of service attack | 2 | 4 | 8 | Bandwidth must be added in order to handle the spikes in the traffic and make the system DDoS resistant. |
|  | Hard disk failure | 1 | 4 | 4 | All the data must be backed up in external drive or in cloud and hard disk must be monitored properly. |
|  | Changes of requirement | 3 | 5 | 15 | Requirements must be analyzed properly from the stakeholders before beginning the project. And any changes must be done as per them. |
|  | Stakeholder conflicts | 2 | 5 | 10 | Identify powerful and influential stakeholder in order to have correct set of resources and invest. |
|  | Fault in Scheduling | 3 | 4 | 12 | Proper schedule planning must be done and make sure none of the task will exceed its deadline. |
|  | Power fluctuation | 3 | 3 | 9 | Power spikes results component failure so point of use protector should be implemented in every specific area which provide protection from variance of power supply. |
|  | Untrained user | 2 | 4 | 8 | User must be guided to use the system and training class should be implemented. |
|  | Unauthorized access | 1 | 4 | 4 | Strong password policy must be implemented and other security measures should be managed. |
|  | Legal Risk | 2 | 5 | 10 | Legacy documents must be prepared before developing a system and develop the system according to the law. |
|  | Lack of resources | 1 | 4 | 4 | Before beginning the project, all the resources required must be analyzed and collected. |

## 6.2: Configuration management

Configuration management is the way of creating and maintaining steadiness between physical and logical assets in an operational environment throughout project life (WhatIs.com, Configuration management, 2008). For example: consistency of product’s performance, design, operational information etc. It identifies bad configuration changes and ensures quick service restoration.

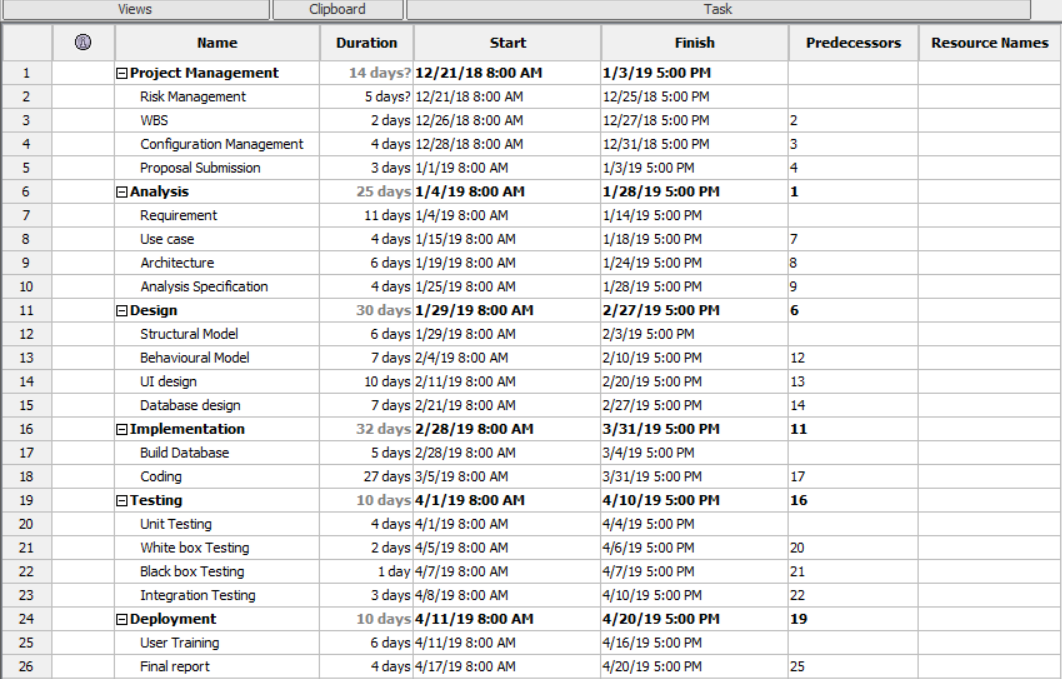
And the configuration management of my project is successfully created which consists of the directories included in the project. The backup of vital aspects is maintained.

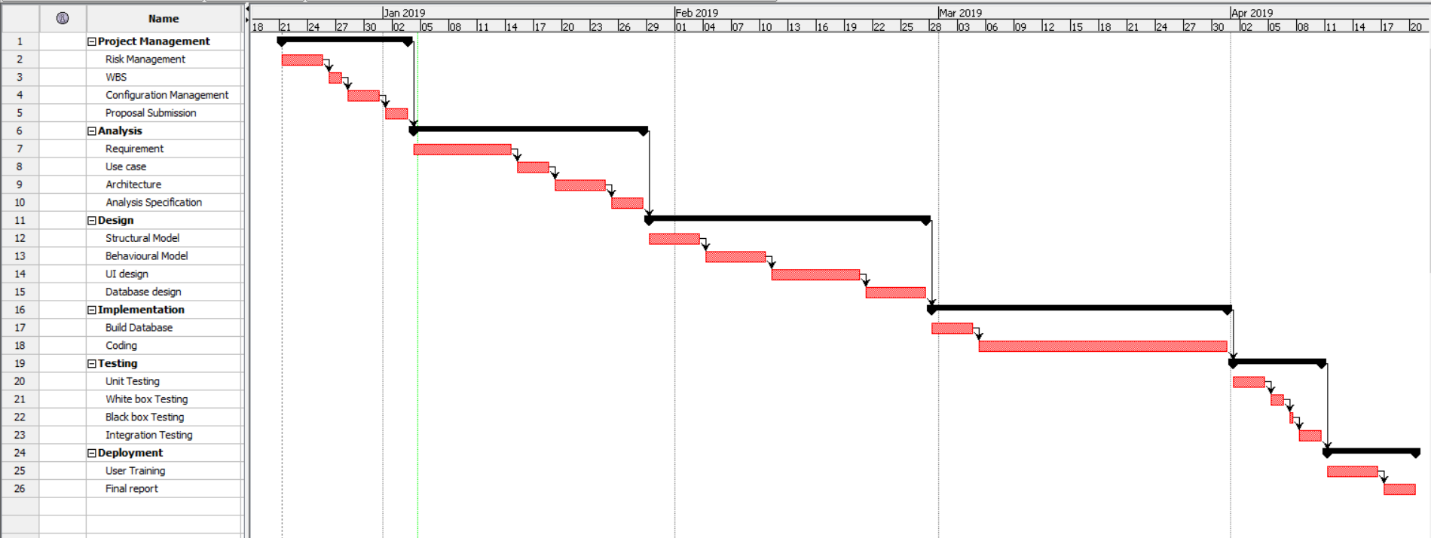
The tree structure of the project is shown below;

## 6.3: Scheduling and Gantt chart

Scheduling is the project planning which lists the project’s milestones, activities and deliverables that includes start and finish dates of each task in the project. It ensures project’s success, allocate resources and set realistic time-frame.

And scheduling of my project is shown through the following Gantt chart;





## 6.4: Future work

The features which were unable to implement at current situation due to time constraint and other issues, will be fulfilled in coming days i.e. the system will be updated with many newer functionalities in future to keep user satisfied with our website.

The features that will be added in future are as follow:

* User-friendly and attractive interface.
* Online payment system.
* Currency converter system.
* Online chat system for direct communication with seller.
* Validating and verifying prescribed report after uploading.
* Notification alert when order is accepted.
* Automatic deduction of medicine quantity according to expiry date and medicine sold.
* Accessibility function

## 6.5: User manual

User manual is a technical communication document which guide user and gives instruction to use the system. It will give all necessary information that a user needs to know about the website.

## 6.6: Limitation

The negative parts of the system are included in limitations. And the drawbacks of the system are as follow:

* It lacks online payment system.
* No user alert when their order is accepted as admin have to call or email them.
* No deduction of quantity according to expiry date and product sold.
* No validation during upload of prescribed report by user.

# Chapter 7: Conclusion

# Chapter 8: References

# Chapter 9: Appendix