Final Year Project Report

E-Meds (An online medical hub)

Project Advisor:

Sir Owais Hakeem

Submitted By:

Muhammad Fahad Aslam (S2016266066)

Hassan Mustafa Zaidi (S2016266006)

Muhammad Ahmed Raza Awan (S2016266101)

Umair Naseer (S2016266124)

Session

Spring 2020

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

We devote this work to our God Almighty, who guide us and never leave us in making this project. He has been the source of our strength throughout this project. A special feeling of gratitude to our loving parents and teachers who has encouraged us all the way, and their encouragement has made us to finish that which we have started.

Final Approval

- **Head of Department** _____
  Department of Computer Science
  University of Management and Technology
  Lahore

- **Program Director (Final Year Projects)** _____
  Department of Computer Science
  University of Management and Technology
  Lahore

- **Supervisor** _____
  Department of Computer Science
  University of Management and Technology
  Lahore

- **Co-Supervisor** _____
  (If any)

Acknowledgment

We would like to give the credits to every person who gave us the chance in the fulfillment of the project. A very warm thanks to our mentor and our FYP advisor Sir Owais Hakeem who helped us throughout all the phases of the project and encouraged us with their suggestions and motivation especially in the fulfillment of the documentation.

We also thank to all the faculty members who provided us all the essential tools to fulfill the project. We also appreciate all the group members, who finalize their tasks on time with all the dedication and gave many useful suggestions in the making of the project. Many thanks to the group leader who had given the time and effort in the conclusion of the project.

Project Title	E-Meds (An Online Medical Hub)
Objective	To servre customers so they can purchase medicine from any medical store on their mobile phone and get a delivery.
Undertaken by	University of Management and Technology
Supervised by	Mr. Owais Hakeem
Starting Date	
Completion Date	
Tools Used	Javascript (for WebApp) & Android Studio (for Mobile Application)
Operating System	
Documentation	

Plagairism Report

Declaration Form

I have carefully examined the documentation of the Final Year Project titled *"E-Meds"*; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with user manual and installation guide.

FYP Advisor Name: _____

Signature: _____ Date:_____

Abstract

E-meds is an online medicine ordering mobile and web application that connects users with all the medical stores. Users can browse through medical stores and directly place an orders of medicines for home delivery at the best price.

E-Meds, an online medicine delivery marketplace, connects the customers to place orders from a huge selection of medical stores. This way medical stores will generate business and customers will get comfort while their purchase. People now a day suffers a lot in finding the medicine from different medical stores and it is possible that the medicine is not available on the stores (means he get the medicine after visiting 2 to 3 stores), so it is a process of time wasting. We gather all the medical stores on one platform so the customers can check that at which medical store all his desired medicines are available so they can take medicines directly or get a delivery.

One of the advantages of buying from E-Meds is that you can purchase medicines without telling anyone. This procedure is good for the people who don't want that their personal issues will show publicaly. E-Meds provides the facility that you don't want to go anywhere, you only have to open the application and order your desired medicines with your delivery location and the medicines will deliver at your doorstep.

CONTENTS

Definitions and Acronyms

Table 1: table of acronyms and definitions

Acronym	**Definition**
DFD	Data Flow Diagram
API	Application Programming Interface
FR	Functional Requirements
NFR	Non Functional Requirements
UI	User Interface
ERD	Entity Relationship Diagram
OTP	One Time Password
JS	JavaScript
RID	Requirement ID
UCID	Use Case ID
TID	Test Case ID
UC	Use Case

List of Figures

List of Tables

CHAPTER # 1

1. INTRODUCTION

1.1 Motivations

Motivation behind choosing E-med project is filling the gaps in the market because there is no such application or website exists in the market right now. We got E-Meds idea from food panda application, food panda covers the restaurants and their customer can have their desired food deliver at their doorstep, In our Project (E-Meds) we gather all the medical stores on one platform so the customer can check that at which medical store all their desired medicines are available so the customer can place an order and get their medicines delivered at their door step. It is easy for the customers that they can find their required medicine easily with the help of our application. Online Medical Stores are unstoppably receiving the green light to facilitate customers.

1.2 Project Overview

- E-Meds is an online medicine ordering mobile application and web that connect users with different medical stores.
- Users can browse through every medical store, which are associated with our application and place order for home delivery at the best price.
- Users can check nearby medical stores by entering their current location.
- E-Meds focuses on making their users life much easier by saving both time and money.
- With just a few clicks, E-meds can deliver the users their required medicines at their doorstep.
- By sending medicine availability request to the medical stores, the user can easily verify that at which medical store their requested medicines are available.
- If the requested medicines are available at different medical stores, then the user can place an order to a specific medical store from which they want to purchase medicines.
- Users can also upload prescription.

There are 3 sides that we cover:

- User Side (Customer).
- Admin Side (Firebase Console)
- Medical Store Side (The online portal for medical store)

    - **User side** is the application side, where the user can send request to all the medical stores to check the medicine availability and place online order. The user have to create account (Registered themselves on application) to check the availability of the medicine, if the medicine is available then the user can place order.

    - **Admin Side** manages all the queries on Firebase.

    - **Medical store side**, also having an online portal. Medical stores login with username and password. In medical store portal, they receive the requests of

availability of medicines from different users and respond according to their requests, for example if the medicine is available medical stores respond yes or otherwise no. Same condition apply on placing order, after responding to the request of required medicine, medical stores confirm the order.

1.3 Problem Statement

People now days facing a lot of difficulties in finding right kind of medicines, because there are some medicines which are unique or for rare illness, and the user or have to spend a lot of time in finding the suitable medicines. It is possible that the medicine is not available on different stores (means the user will get their medicines after visiting many stores), so it is a process of time wasting. We gather all the medical stores on one platform so the customers can check that at which medical store their desired medicines are available so the user can buy medicines directly from our application and saves a lot of time.

1.4 Objectives

The following are the objectives that we expect after the completion of this project:

- We expect that we gather all the medical stores on one platform.
- Satisfy the customers with our system.
- Providing platform, in which a customer can check the availability of medicine from all medical stores simply by generating a request.
- E-Meds provide users privacy that that user can buy medicines without other people knowing.
- To make our application **Decision Convenience**, Ratings and reviews for medical stores will upload frequently from users, which help the user to find right medical store for their medicine.
- To make our application **Location Convenience,** Customers can search nearby medical stores.

CHAPTER # 2

2. DOMAIN ANALYSIS

2.1 Customer

E-Meds, an online medicine delivery marketplace provide ease to the customers and connects the customers to place orders from a huge selection of medical stores. We know that customers are the backbone of any company. E-Med provides their customers healthy range of option to purchase medicines from different stores. The application allows customers or patients to login into the app, order medicines, and get it shipped to their doorsteps. This way medical store will generate business and customers will get full comfort while their purchase.

Advantages of E-Meds for their customers:

- Easy to use, easy to access and within range.
- Comfort of shipping medicines at home.
- Customers can order with full privacy.
- User can see nearby medical stores.
- User can also upload prescription.

Customer Relationships: Feedback and reviews of customers, Question and answer section, social media section.

Customer Segments: Customers can find an easy way to buy medicines online at their places.

Cost Involved: Ad cost, Mobile Application development cost, Employees Salary.

Revenue Streams: Drugs and other sales, delivery charges, on-page ad, sponsored Ad listings.

2.2 Stakeholders

Table 2.1: list of stakeholders

Stakeholder	Role in System
Admin	Manages all queries on the database.
Medical Store	Receives the medicines availability requests and responds accordingly. After that medical store confirms the orders that are generated by users then users receive a confirmation message.
User	Once the user creates the account and registered themselves, they can do the both (Check medicines availability on all medical stores and place an order.

This table describes the roles of stakeholders

2.3 Affected Groups with social or economic impact

Following are the list of people that affected socially or economically with this project

User:

We want our system to serve users in a fast-paced and most efficient way and will positioned itself as an attractive model in the healthcare space. User can easily purchase medicines from home and save their time in finding their desired medicines. User just tinker with the computer or smartphone and easily able to get the medicines they need. Users can easily switching tabs to visit one medical store to another medical store. It is much easier when it comes to comparing prices of medicines. Many medical stores are selling at much lower price than other medical store. Perhaps, it may be best that the user can check out many medical stores just to have a better overview of the average price. Especially it provides benefits to those users who are living far from pharmacies. For people who have a hard time to travel to a physical pharmacy.

Medical Store:

Pharmacies can easily connect with users and generate revenue by selling their products. With increasing demand for online purchasing, more and more businesses are moving to online system. Also pharmacies have a chance to introduce their company names on online platform.

2.4 Dependencies/ External Systems

No dependencies/ external systems, this project depends upon for its completion.

2.5 Reference Documents

Following are the reference documents and websites:

www.foodpanda.pk , www.dawaai.pk , www.sehat.com.pk , www.medicalstore.com.pk

2.5.1 Related Projects

1. Medical Store.com.pk:

Online Medical Store is an online pharmacy in Pakistan delivering health-care products. Online Medical Store in Pakistan delivers medicines through home delivery.

2. Sehat.com.pk:

Sehat.com.pk is a project by members from the 4th generation of Fazal Din family with many successful ventures into different aspects of the industry. It is an online pharmacy in Pakistan delivering health-care products.

3. Dawaai.pk:

Dawaai.pk is an innovative pharmacy-led healthcare group. Their mission is to provide pharmacy services with Ease and Quality Products. They work on the basis of retail pharmacy by taking authentic medicines to patient's doorstep as quickly as possible.

2.5.2 Feature Comparison

Table 2.2: Feature Comparison

Sr No.	Comparison Feature	Medical Store.com.pk	Sehat.com.pk	Dawaai.pk	Remarks
1	Proving products from different stores	Have their own database of different products	They have their own products. They don't interact with medical stores	They have their warehouses, they are not associated with any other companies	We are connected with all the stores and companies which are associated with our application
2	Home Delivery Option	Provide home delivery option	They have home delivery option	Providing the feature of home delivery	We improve our delivery option and provide fast and reliable delivery
3	Uploading Prescription Option	Provide uploading prescription option	Don't have uploading prescription feature	Have the feature of uploading prescription	Providing the feature of uploading prescription facilitate the user
4	Android Based System	Does come in android	Don't have android based system	Providing android application	We will add android based functionalities.

This table describes the comparison about the features of E-Meds as compared to the other medicine applications and websites in the market.

CHAPTER # 3

3. REQUIREMENTS ANALYSIS

3.1 Requirements

Table 3.1: User (Functional Requirements) this table describes the functional requirements that (E-Meds) provided to the users

Requirement ID	Description	Must/Want	Comments
FR1-01	The application shall provide access to the user so they can register themselves.	Must	User registered with Mobile Number and set Password.
FR1-02	The application shall provide an interface for Login/Logout, Signup and Forgot password for users.	Must	User Login with Mobile Number and Password.
FR1-03	The application shall provide an interface to check nearby medical stores.	Must	User can check nearby medical stores by entering their location.
FR1-04	The application shall provide an interface to search medicines.	Must	User can search medicines by typing the name of the medicine.
FR1-05	The application shall provide an interface to check medicine availability.	Must	User can check the availability of medicines from all medical stores.
FR1-06	The application shall provide an interface to Upload Prescription.	Must	User can upload prescription by taking picture or from gallery.
FR1-07	The application shall provide an interface to check Prescription availability.	Must	User can check the prescription availability from all medical stores.
FR1-08	The application shall provide an interface to update profile.	Must	User can update their profile by providing their general information.

Table 3.2: Medical Store (Functional Requirements) this table describes the functional requirements that (E-Meds) provided to the medical stores

Requirement ID	Description	Must/Want	Comments
FR2-01	The application shall provide Login access to medical stores.	Must	Medical store Login themselves.
FR2-02	The application shall provide an interface to confirm medicine requests.	Must	Medical stores can confirm requests which are requested by the users.
FR2-03	The application shall provide an interface to confirm order.	Must	Medical stores confirm the requested order.
FR2-04	The application shall provide an interface to confirm prescription request.	Must	Medical stores confirm prescription requests.
FR2-05	The application shall provide an interface to confirm prescription order.	Must	Medical stores can confirm prescription requests.

Table 3.3: Admin (Functional Requirements) this table describes the functional requirements that (E-Meds) provided to the Admin

Requirement ID	Description	Must/Want	Comments
FR3-01	The application shall provide an interface to Manage users.	Must	Admin can manage user database through online portal.
FR3-02	The application shall provide an interface to Manage medicines.	Must	Admin can manage medicines information.
FR3-03	The application shall provide an interface to Manage medical store users.	Must	Admin can manage medical store user database through online portal.

3.2 List of Actors

User: Customer using E-meds application.

Medical Store: Medical stores operator, which are associated with our application.

Admin: They monitor all the activities through backend (Firebase).

3.3 List of use cases

Register: User can create account by giving information to register.

Login: User can login by giving information.

Check Nearby Stores: User can see nearby medical stores to find their desired medicines.

Search Medicines: User can search medicines on their requirement or on their demand.

Check Medicines Availability: User can see all medicines and check availability of medicines, which they need.

Check Prescription Availability: User can check availability of medicines, which they need by their prescription.

Upload Prescription: User can upload prescription for medical store to check the availability of medicine.

Update Profile: User can update their details of his profile like: name, address and password.

Place Order: User can place order of medicines, which they need.

Confirm Medicines Request: Medical store user can give the response to the customers of their medicines request.

Confirm Prescription Request: Medical store user can give the response to the customers of their prescription.

Confirm Medicine Order: Medical store user can confirm the medicine order of their customers.

Confirm Prescription Order: Medical store user can confirm the prescription order of their customers.

Manage Medicines: Admin manage medicines (add, update and delete)

Manage Users: Admin manage users (add, update and delete)

Manage Medical Store Users: Admin manage medical store users (add, update and delete)

3.4 System use case diagram

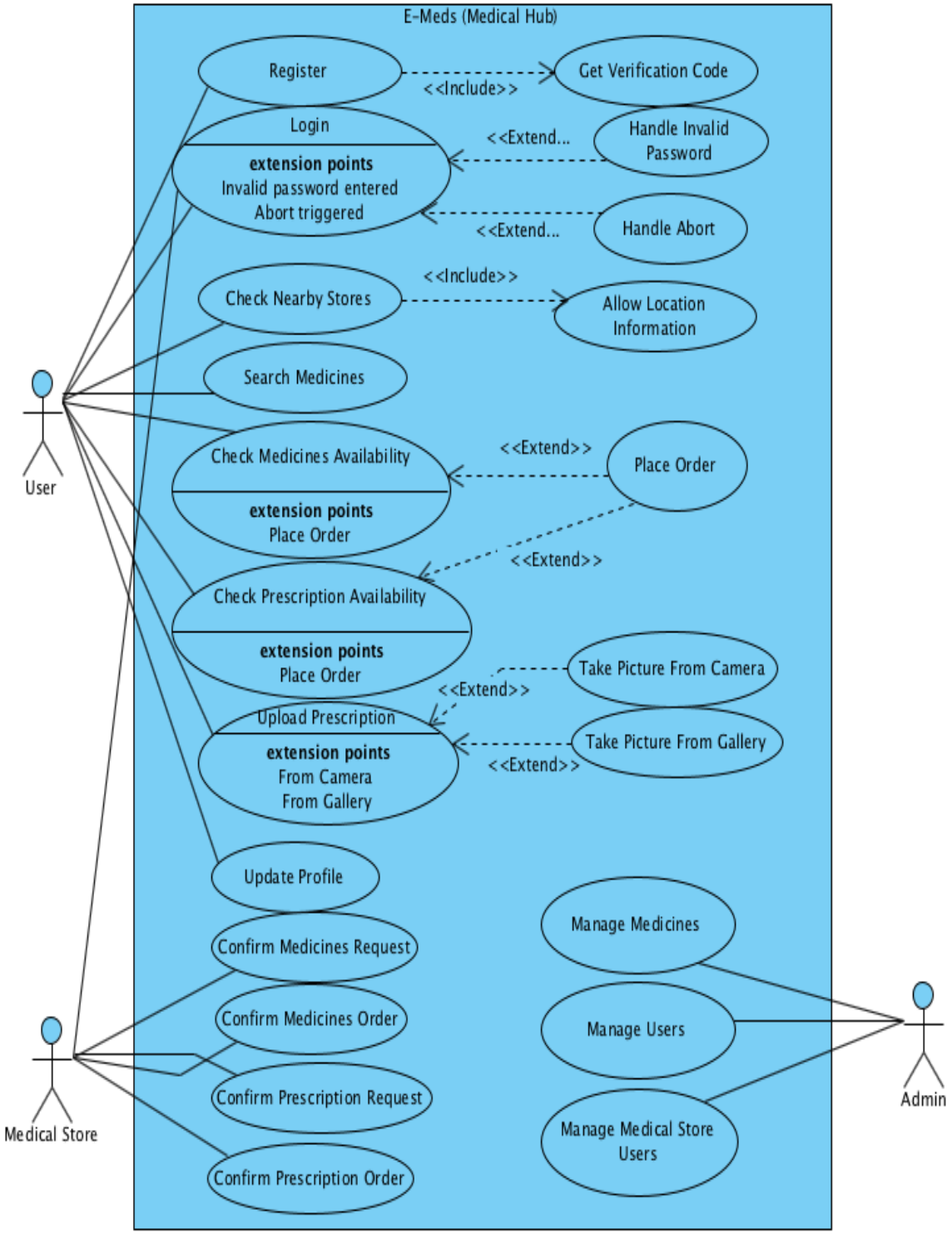

Figure 3.1: System Use Case Diagram - This figure describes the system usecase diagram, that how every usecase is related to every actor in the system

3.5 Extended use cases

Table 3.4: Extended Use Case (Register) this table describes the register extended use case

Use Case ID:	UC01		
Use Case Name:	Register		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can create their account by giving their information. Firstly user has to enter their mobile number, after that they will receive a message on their given mobile number. After verifying the mobile number user create a password, and the user account will created successfully.		
Trigger:	When user tap on register.		
Preconditions:	User must have valid phone number.		
Post conditions:	User can access the all features.		
Normal Flow:	1. User opens the app. 2. User taps on register button. 3. User enters phone number. 4. User enters his password. 5. User get verification code on his phone number with they entered. 6. User registration proceeds successfully.		
Alternative Flows:	5a. The 5th point of the flow, the user enter invalid verification code 1. System message user to resend verification code. 2. User taps the resend code button. 3. Verification code sends to user's phone again. 4. User enters verification code. 5. User registers successfully if the entered code is correct.		
Exceptions:	3a. The 3rd point of the flow, the user enter invalid phone number 1. System message user that they type wrong phone number. 2. Use Case resumes to the 2nd point. 3b. The 3rd point of the flow, if customer enter phone number 1. System prompt user that this phone number is already registered. 2. System prompt user that please enter other phone number. 3. Use Case resumes to step 2.		
Includes:	N/A		
Frequency of Use:	Whenever user interacts the app for first time.		
Special Requirements:	No		
Assumptions:	User uses the new phone number, which is not already registered.		
Notes and Issues:	No		

Table 3.5: Extended Use Case (Login) this table describes the login extended use case

Use Case ID:	UC02		
Use Case Name:	Login		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User, Medical Store		
Description:	The extended use-case is for log-in on the application. The user has to enter the valid phone number, which he/she used for creating the account, and the password. User login successfully after pressing the enter button.		
Trigger:	User tap on Login.		
Preconditions:	User must have valid phone number.		
Post conditions:	User can access the all features.		
Normal Flow:	1. Customer opens the app. 2. Customer types his phone number. 3. Customer enters his code. 4. Customer login in the account. 5. Customer logged in successfully.		
Alternative Flows:	5a. 5a. The 5th point of the flow, the user is invalid 1. User enters his phone number. 2. User enters his password. 3. Customer login error or user is not valid. 4. System sends message to enter his phone no and password again. 5. Customer achieves the goal.		
Exceptions:	2a. 2nd point of the flow, the customer enter incorrect phone number 1. Application send message that they enter wrong phone number. 2. Usecase return on step 2. 3a. Point 3, if customer enters invalid code. 1. Application messages that they type wrong code. 2. Usecase return to 3rd step.		
Includes:	N/A		
Frequency of Use:	Whenever user interacts the app for sign-in.		
Special Requirements:	No		
Assumptions:	User must be registered.		
Notes and Issues:	No		

Table 3.6: Extended Use Case (Check Nearby Stores) this table describes the check nearby stores extended use case

Use Case ID:	UC03		
Use Case Name:	Check Nearby Stores		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can see all medical stores, or also provide their current location to check nearby medical stores. User can also checks the ratings of different medical stores. User also checks the distance of the medical stores from their current location. User can check the delivery details of different medical stores.		
Trigger:	When user check nearby medical store.		
Preconditions:	1. User has valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User checks the nearby medical store. 2. User checks the rating of different medical stores. 3. User checks the location and distance from their destination. 4. User checks the delivery details or minimum order details of nearby stores.		
Normal Flow:	7. User login its account. 8. User taps on nearby store. 9. System gets permission to allow location. 10. Nearby store list will be appearing. 11. User can see the distance from their destination by clicking on store.		
Alternative Flows:	2a. Second point of the flow, perhaps the location not pin on your device, 6. Application messages the user that your location is off please turn on your location first 7. Application will show to location option to turn on 8. Usecase continuous on step 2		
Exceptions:	2a. 2nd point, perhaps the location not pin on your device, 1. Application message the user that your location is off please turn on your location first 2. App will show to location option to turn on 3. Usecase continuous on step 2		
Includes:	Login		
Frequency of Use:	Whenever user wants to see the nearby medical stores location.		
Special Requirements:	No		
Assumptions:	User Is Valid		
Notes and Issues:	NO		

Table 3.7: Extended Use Case (Search Medicine) this table describes the search medicine extended use case

Use Case ID:	UC04		
Use Case Name:	Search Medicines		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User taps the search option in the navigation bar. Search page will appear. User can search their desired medicines, after that the medicines will appear on the search page. Searching medicines will provide ease to the users, so the user not faces any difficulty in searching the medicines.		
Trigger:	When user wants to search the medicines.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User can search medicine successfully on their demand.		
Normal Flow:	1. User login its account. 2. User taps the search options in navigation bar. 3. Search page will be appearing. 4. User can search their desired medicine. 5. Medicines will be appearing in search page.		
Alternative Flows:	4a. 4th point in the flow, the customer enters incorrect medicine name 1. Application will not show any thing in the search page because of wrong spelling. 2. Usecase continuous on step 4.		
Exceptions:	N/A		
Includes:	Login		
Frequency of Use:	Whenever user want to search medicines.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.8: Extended Use Case (Check Medicine Availability) this table describes the check medicine availability extended use case

Use Case ID:	UC05		
Use Case Name:	Check Medicines Availability		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	This extended use case is for check medicine availability. From this feature the user can check that their desired medicines are available on which medical stores. From this key feature user can easily find their medicines and place online order.		
Trigger:	When user wants to see the availability of medicines.		
Preconditions:	1. User has a valid unique Id and password. 2. User has to be logged in.		
Post conditions:	1. User can search and check the availability of medicine on their demand.		
Normal Flow:	1. User login its account. 2. User taps the search options in navigation bar. 3. Search page will be appearing. 4. User can search their desired medicine. 5. User search medicines will be appearing in search page. 6. User adds medicine to the cart by selecting them on their demand. 7. User tap on request button. 8. User medicines request response will be appearing in notification tab.		
Alternative Flows:	4a. Point 4 of the flow, the customer enter invalid medicine name. 1. Application will not show any thing in the search page because of wrong spelling. 2. Usecase continuous on step 4.		
Exceptions:	7a. Point 7 of the flow, whether user cart is empty 1. Application will prompt user the message of empty cart. 2. Usecase continuous on step 7.		
Includes:	Login		
Frequency of Use:	When user want to check the availability of medicine on different medical stores.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.9: Extended Use Case (Check Prescription Availability) this table describes the check prescription availability extended use case

Use Case ID:	UC06		
Use Case Name:	Check Prescription Availability		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can also upload prescription to the medical stores and can check their prescription availability response from medical stores.		
Trigger:	When user wants to see the availability of medicines.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User can check the availability of medicine by their prescription.		
Normal Flow:	1. User login its account. 2. User taps the upload icon in navigation bar. 3. Upload prescription page will be appearing. 4. User can tab on their required method to upload prescription.		
Alternative Flows:	4a. Point 4 of the flow, customer select gallery option 1. Application will send message to allow the access of media files to upload prescription. 2. Usecase continuous on step five. 4b. Fourth point in the flow, the consumer choose camera option 1. Application will prompt user that please allow the access of camera to upload prescription. 2. Usecase continuous on point 5.		
Exceptions:	4a. Point 4 of the flow, the customer choose gallery option and don't give permission 1. Application will message customer that please allow the access of media files to upload prescription. 2. Usecase resumes on step 4. 4b. Step 4 of the flow, the customer select camera option 1. System will show user that please allow the access of camera to upload prescription. 2. Usecase continuous to step 4.		
Includes:	Login		
Frequency of Use:	When user want to check the availability of medicine on different medical stores by their prescription.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.10: Extended Use Case (Upload Prescription) this table describes the upload prescription extended use case

Use Case ID:	UC07		
Use Case Name:	Upload Prescription		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can upload prescription from gallery or by taking prescription photo from their camera		
Trigger:	When user wants to upload prescription for medical stores.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User uploads prescription successfully.		
Normal Flow:	1. User login its account. 2. User taps the upload icon in navigation bar. 3. Upload prescription page will be appearing. 4. User can tab on their required method to upload prescription. 5. User taps on request button.		
Alternative Flows:	4a. Fourth step of the flow, the customer select gallery option 1. Application show message to the customer that please allow the access of media files to upload prescription. 2. Usecase continuous to step five. 4b. 4th point of the normal flow, the consumer select camera option 1. App will show the customer that please allows the access of camera to upload prescription. 2. Usecase continuous on step five.		
Exceptions:	4a. Step four of the flow, the consumer select gallery option but don't give permission 1. App will show the customer that please allows the access of media files to upload prescription. 2. Usecase continuous on point 4. 4b. In point four of the flow, the consumer select camera option 1. App will show the customer that please allows the access of camera to upload prescription. 2. Usecase continuous on step 4.		
Includes:	Login		
Frequency of Use:	Whenever user want to upload prescription.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.11: Extended Use Case (Update Profile) this table describes the update profile extended use case

Use Case ID:	UC08		
Use Case Name:	Update Profile		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can update their profile, their names, contact number so the medical store can easily communicate with the customer. User can also change their request where they want to deliver the medicines. User can also change their account password; if they forget their password then they also reset the password by entering the verification code.		
Trigger:	When user request to update the information of their profile.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User updates their profile successfully.		
Normal Flow:	1. Customer login its account. 2. User taps the profile icon in navigation bar. 3. User profile page will be appearing. 4. User can tab on edit profile button. 5. User profile page is open and it is edit able. 6. System has shown the fields of its name, address and password which containing their current data. 7. User updates the field. 8. User confirms update changes and tab on update button. 9. Profile updated successfully.		
Alternative Flows:	Nil		
Exceptions:	N/A		
Includes:	Login.		
Frequency of Use:	Whenever user wants to update their profile information.		
Special Requirements:	Performance requirements: the response time should be fast the profile should be updated in real time.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.12: Extended Use Case (Confirm Medicine Request) this table describes the confirm medicine request extended use case

Use Case ID:	UC9		
Use Case Name:	Confirm medicine request		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	Firstly the user has to send the required medicine availability request to the medical store. Then the medical store after receiving the request, confirm the request and send the response to the user that the required medicines are available or not.		
Trigger:	When medical store user give response of their customers medicines request.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. Medical store user can check the requests of their customers. 2. Medical store user gives them response of their medicine request.		
Normal Flow:	1. Medical store user login its account. 2. Medical store user tab request button on navigation. 3. Request page will be appearing. 4. Medical store user can check the customer request of medicines. 5. Medical store user must be response them such as: available or not available.		
Alternative Flows:	Nil		
Exceptions:	4a. Step 4 of the flow, the customer request nothing showing 1. Medical store customer tab on request sign. 2. Medical store customer can repeat the process 1-4 time Use case resume on step 4		
Includes:	Login		
Frequency of Use:	Whenever user wants to response customer medicine request availability.		
Special Requirements:	Performance requirements: the response time should be fast to response customer.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.13: Extended Use Case (Confirm Medicine Order) this table describes the confirm medicine order extended use case

Use Case ID:	UC10		
Use Case Name:	Confirm Medicine Orders		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	After sending the medicine availability request to the user, the user give the order to their desired medical stores, and after that the medical store confirm the order.		
Trigger:	When medical store user confirmed order of their customer medicines request.		
Preconditions:	1. Medical store user has a valid phone number and password. 2. Medical store user has to be logged in.		
Post conditions:	1. Medical store user confirms the order of their customer medicine.		
Normal Flow:	1. Medical store user login its account. 2. Medical store user tab new order button on navigation. 3. New order page will be appearing. 4. Medical store user can check the customer order request of medicines. 5. Medical store user must be response them such as by: accepting or rejecting order request.		
Alternative Flows:	Nil		
Exceptions:	4a. Point four of the flow, the consumer order request of nothing showing 1. Medical store customer tab on request sign. 2. Medical store customer can repeat the process 1-4 time Use case resume on step 4		
Includes:	Login		
Frequency of Use:	Whenever user wants to confirm customer confirmed order request.		
Special Requirements:	Performance requirements: the response time should be fast to response customer.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.14: Extended Use Case (Confirm Prescription Request) this table describes the confirm prescription request extended use case

Use Case ID:	UC11		
Use Case Name:	Confirmation Prescription Request		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	The medical store confirms the prescription request, when the user sends the request to the medical store. Sending the prescription requests mean that the user wants to know the availability of the medicine from the medical store. The request pop-up to the medical store and the medical store confirm the prescription request.		
Trigger:	When medical store user wants to give response of their customer medicines request.		
Preconditions:	1. Medical store user has a valid phone number and password. 2. Medical store user has to be logged in.		
Post conditions:	1. Medical store user confirms the request of their customer prescription medicine.		
Normal Flow:	1. Medical store user login its account. 2. Medical store user tab prescription request button on navigation. 3. Prescription request page will be appearing. 4. Medical store user can check the customer prescription request of medicines. 5. Medical store user must be response them such as: available or not available.		
Alternative Flows:	Nil		
Exceptions:	4a. Step 4 of the flow, nothing is showing in the prescription request. 1. Medical store customer tab on request button. 2. Medical store customer can repeat the process 1-4 time Use case resume on step 4		
Includes:	Login		
Frequency of Use:	Whenever user wants to response customer prescription request availability.		
Special Requirements:	Performance requirements: the response time should be fast to response customer.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.15: Extended Use Case (Confirm Prescription Order) this table describes the confirm prescription order extended use case

Use Case ID:	UC12		
Use Case Name:	Confirm Prescription Orders		
Created By:	Hassan	**Last Updated By:**	Fahad
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	Confirm prescription order means, after the medical store confirm the prescription request and confirm that the medicines are available then the user can give order to their desired medical store and the medical store confirm the prescription order.		
Trigger:	When medical store user confirmed prescription order of their customer medicines request.		
Preconditions:	1. Medical store user has a valid phone number and password. 2. Medical store user has to be logged in.		
Post conditions:	1. Medical store user confirms the prescription order of their customer medicine.		
Normal Flow:	1. Medical store user login its account. 2. Medical store user tab new prescription order button on navigation. 3. New prescription order page will be appearing. 4. Medical store user can check the customer prescription order request. 5. Medical store user must be response them such as by: accepting or rejecting order request.		
Alternative Flows:	Nil		
Exceptions:	4a. Step 4 of the flow, the customer orders request nothing showing 1. Medical store customer tab on request button. 2. Medical store customer repeat the process 1-4 time. Usecase remains on step 4		
Includes:	Login		
Frequency of Use:	Whenever user wants to confirm customer confirmed prescription order request.		
Special Requirements:	Performance requirements: the response time should be fast to response customer.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.16: Extended Use Case (Allow Location Information) this table describes the allow location information extended use case

Use Case ID:	UC13		
Use Case Name:	Allow Location Information		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	Allow location information means that, when the user wants to search nearby location, then they have to give permission to the device to locate the current location. After give the permission only the nearby medical stores will appear at the user interface.		
Trigger:	When user wants to see nearby medical store.		
Preconditions:	1. User has valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User will allow location information.		
Normal Flow:	1. User login its account. 2. User taps on nearby store. 3. Nearby Store page will appear. 4. Nearby location allows permission message prompt. 5. Allow the permission to access your location.		
Alternative Flows:	Nil		
Exceptions:	2a. Step 2 of the flow, perhaps the location is not provided by the system 1. System will prompt user that your location is off please turn on your location first. 2. System will show to location option to turn on. 3. Use case resume on step 2.		
Includes:	Login		
Frequency of Use:	Whenever user wants to see the nearby medical stores location.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.17: Extended Use Case (Place Orders) this table describes the place order extended use case

Use Case ID:	UC14		
Use Case Name:	Place Order		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	After searching the required medicines the user can place order of the required medicines. User adds medicines to the cart, and the total bill of the purchase will show to the user.		
Trigger:	When user wants to place order.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	1. User can place order of medicine on their demand.		
Normal Flow:	1. User login its account. 2. User taps the search options in navigation bar. 3. Search page will be appearing. 4. User can search their desired medicine. 5. User search medicines will be appearing in search page. 6. User adds medicine to the cart by selecting them on their demand. 7. User tap on request button. 8. User order will be placed.		
Alternative Flows:	4a. In point 4 of the flow, the consumer enters invalid medicines names 1. Application not shows anything in the search page because of wrong spelling. 2. Usecase continuous on point four.		
Exceptions:	7a. Point seven of the flow, the consumer cart is empty 1. App will show the customer that cart is empty user cannot send the empty cart order will not be placed 2. Usecase continuous to step 7		
Includes:	Login		
Frequency of Use:	When user want to check the availability of medicine on different medical stores.		
Special Requirements:	No		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.18: Extended Use Case (Upload Prescription From Camera) this table describes the upload prescription from camera extended use case

Use Case ID:	UC15		
Use Case Name:	Upload Prescription from Camera		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can upload prescription from the camera by clicking on the upload prescription button. After that the application prompt the user to what you want camera or gallery, and the user choose camera option and take clear picture of the prescription and upload the prescription.		
Trigger:	When user upload prescription picture for medical stores.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	User upload prescription picture from camera successfully.		
Normal Flow:	1. User login its account. 2. User taps the upload icon in navigation bar. 3. Upload prescription page will be appearing. 4. User can tab on take picture from camera. 5. Camera will be open to take picture. 6. User tap on request button..		
Alternative Flows:	4a. Point four of the flow, the customer select camera option 1. App will show the customer that please allows the access of camera to upload prescription. Usecase continuous on point 5		
Exceptions:	4a. Point four of the flow, the consumer select camera option 1. Application will show the customer that please allows the access of camera to upload prescription. Usecase continuous on step 4. 6a. In point 6 of flow, if consumer not take picture 1. Application will show that picture is not taken user cannot send empty request. Use Case resumes on step 4.		
Includes:	Login		
Frequency of Use:	Whenever user want to upload prescription via camera.		
Special Requirements:	Special requirement. User should know to use camera with good techniques.		
Assumptions:	User Is Valid.		
Notes and Issues:	NO		

Table 3.19: Extended Use Case (Upload Prescription From Gallery) this table describes the upload prescription from gallery extended use case

Use Case ID:	UC16		
Use Case Name:	Upload Prescription from gallery		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	User		
Description:	User can upload prescription from the gallery by clicking on the upload prescription button. After that the application prompt the user to what you want camera or gallery, and the user choose gallery option and choose the prescription photo and upload it.		
Trigger:	When user upload prescription picture from gallery for medical stores.		
Preconditions:	1. User has a valid phone number and password. 2. User has to be logged in.		
Post conditions:	User upload prescription picture from gallery successfully.		
Normal Flow:	1. User login its account. 2. User taps the upload icon in navigation bar. 3. Upload prescription page will be appearing. 4. User can tab on choose from gallery option. 5. Gallery will be open for selecting picture. 6. Select picture and tab on ok button. 7. User taps on request button.		
Alternative Flows:	4a. Step 4 of the flow, the customer select gallery option 1. Application will show the customer that that please allows the access of media file to upload prescription. Usecase continuous on step 5.		
Exceptions:	4a. Point 4 of the flow, the consumer will don't give permission 1. Application will show the customer that please allows the access of media files to upload prescription. 2. Usecase continuous on point 4. 6a. Step 6 of the flow, the customer not selecting picture. 1. Application will show the customer that picture is not selected user cannot send empty request. Usecase continuous on step 4.		
Includes:	Login.		
Frequency of Use:	Whenever user want to upload prescription via gallery.		
Special Requirements:	Special requirement. User should know to search picture from media files.		
Assumptions:	User Is Valid		
Notes and Issues:	NO		

Table 3.20: Extended Use Case (Manage Medicines) this table describes the manage medicines extended use case

Use Case ID:	UC17		
Use Case Name:	Manage Medicines		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	Admin add medicines on the application and can also remove the medicines. The added medicines will show to the user, and the user can place order from these medicines which are added on the application.		
Trigger:	When user wants to change the information of their profile		
Preconditions:	1. User has a valid ID and password. 2. User has to be logged in.		
Post conditions:	1. Admin see the list of medicines. 2. Admin can manage medicines 3. Admin can add medicines. 4. Admin can delete medicines.		
Normal Flow:	1. Admin login its account in firebase. 2. Admin go into console 3. Admin click on database 4. Admin click on product object 5. Admin edit products object		
Alternative Flows:	5a. Point 5 of the flow, the admin edit products. 1. System will allow admin to update medicine product information like: name, price, description. 2. System will allow admin to add medicine product information like: name, price, description. 3. System will allow admin to delete medicine products. Usecase continuous to step 5.		
Exceptions:	N/A		
Includes:	Login		
Frequency of Use:	Whenever user wants to manage medicines.		
Special Requirements:	No		
Assumptions:	User Is Valid		
Notes and Issues:	NO		

Table 3.21: Extended Use Case (Manage Users) this table describes the manage users extended use case

Use Case ID:	UC18		
Use Case Name:	Manage Users		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Medical store		
Description:	The admin can manage all the users through firebase account. The admin have the access to see all the information of the user, like their names, address and phone number. Admin can also delete the user account.		
Trigger:	When user wants to change the information of their profile		
Preconditions:	1. User has a valid ID and password. 2. User has to be logged in.		
Post conditions:	1. Admin see the list of users. 2. Admin can manage users of medical stores and their customers.		
Normal Flow:	1. Admin login its account in firebase. 2. Admin go into console 3. Admin click on database 4. Admin click on user object 5. Admin edit user object		
Alternative Flows: [Alternative Flow 1 – Not in Network]	5a. In point 5 of the flow, the controller can do management of the user. 1. System will allow admin to add user information manually like: name, phone number, password and address. 2. System will allow admin to delete user. Usecase resumes to step 5.		
Exceptions:	N/A		
Includes:	Login		
Frequency of Use:	Whenever user wants to manage users.		
Special Requirements:	No		
Assumptions:	User Is Valid		
Notes and Issues:	NO		

Table 3.22: Extended Use Case (Manage Medical Store Users) this table describes the manage medical store users extended use case

Use Case ID:	UC19		
Use Case Name:	Manage Medical Store Users		
Created By:	Raza	**Last Updated By:**	Raza
Date Created:	11-April-2020	**Last Revision Date:**	11-April-2020
Actors:	Admin		
Description:	Admin have access to see all the lists of medical stores. Admin manage medicine store account by giving them id and password. Admin can also add new medical stores. Admin can also delete the medical stores.		
Trigger:	When admin wants to manage medical stores		
Preconditions:	1. User has a valid ID and password. 2. User has to be logged in.		
Post conditions:	1. Admin see the list of medical stores. 2. Admin managed medical stores. 3. Admin can assign medical store id and password. 4. Admin can add medical store users. 5. Admin can delete medical store users.		
Normal Flow:	1. Admin login its account in firebase. 2. Admin go into console 3. Admin click on database 4. Admin click on medical store user object 5. Admin edit medical store user object		
Alternative Flows:	5a. In point 5 of the flow, controller edit products. 1. System will allow admin to update medical store information like: name, id, and password. 2. System will allow admin to add medical store information like: name, id and password. 3. System will allow admin to delete medicine products. Usecase continuous on step 5.		
Exceptions:	N/A		
Includes:	Login		
Frequency of Use:	Whenever user wants to manage medical store users.		
Special Requirements:	No		
Assumptions:	User Is Valid		
Notes and Issues:	NO		

3.6 User interfaces (mock screens)

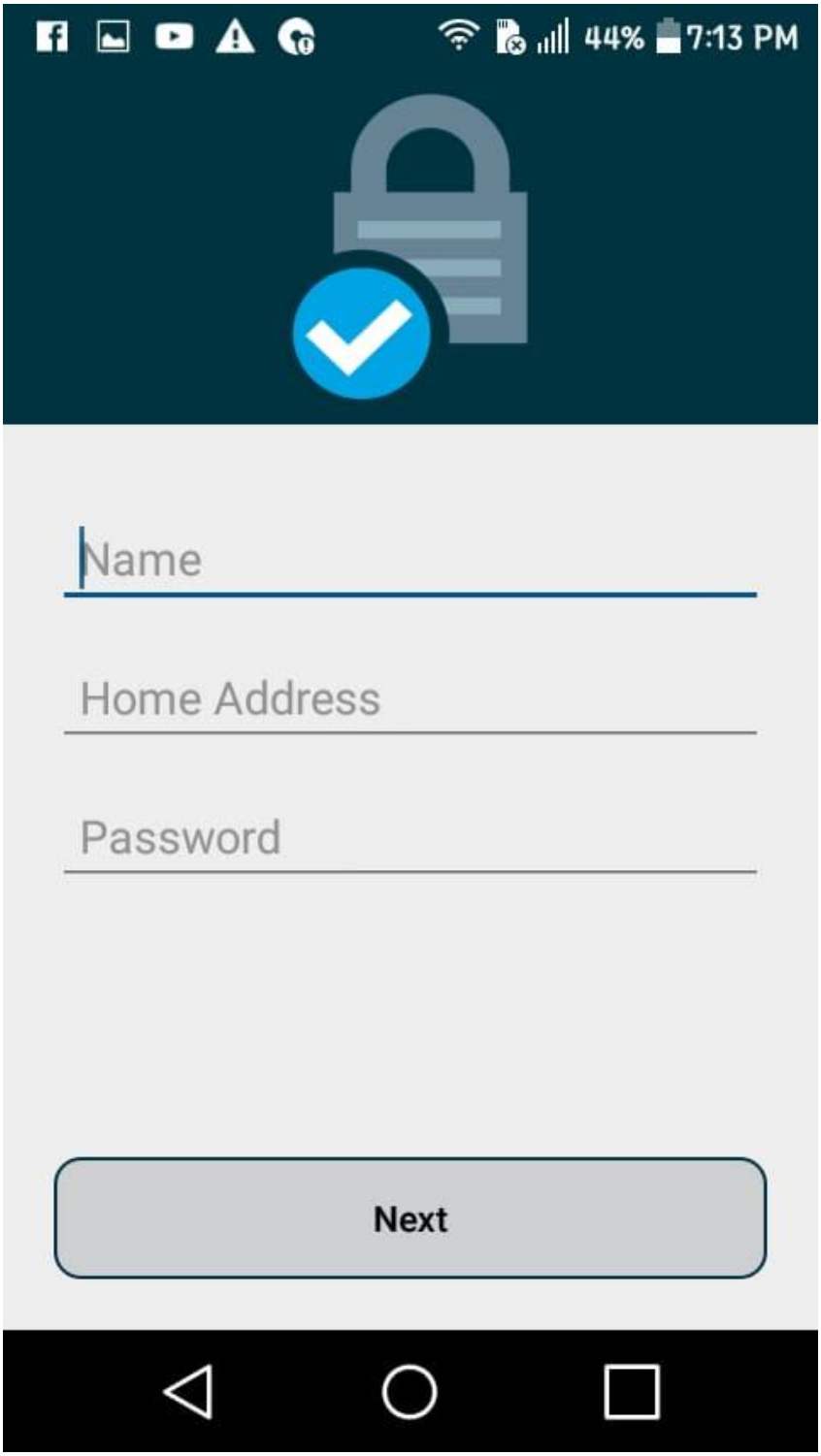

Figure 3.2: Prototype-1: This is a register screen to register user in our system

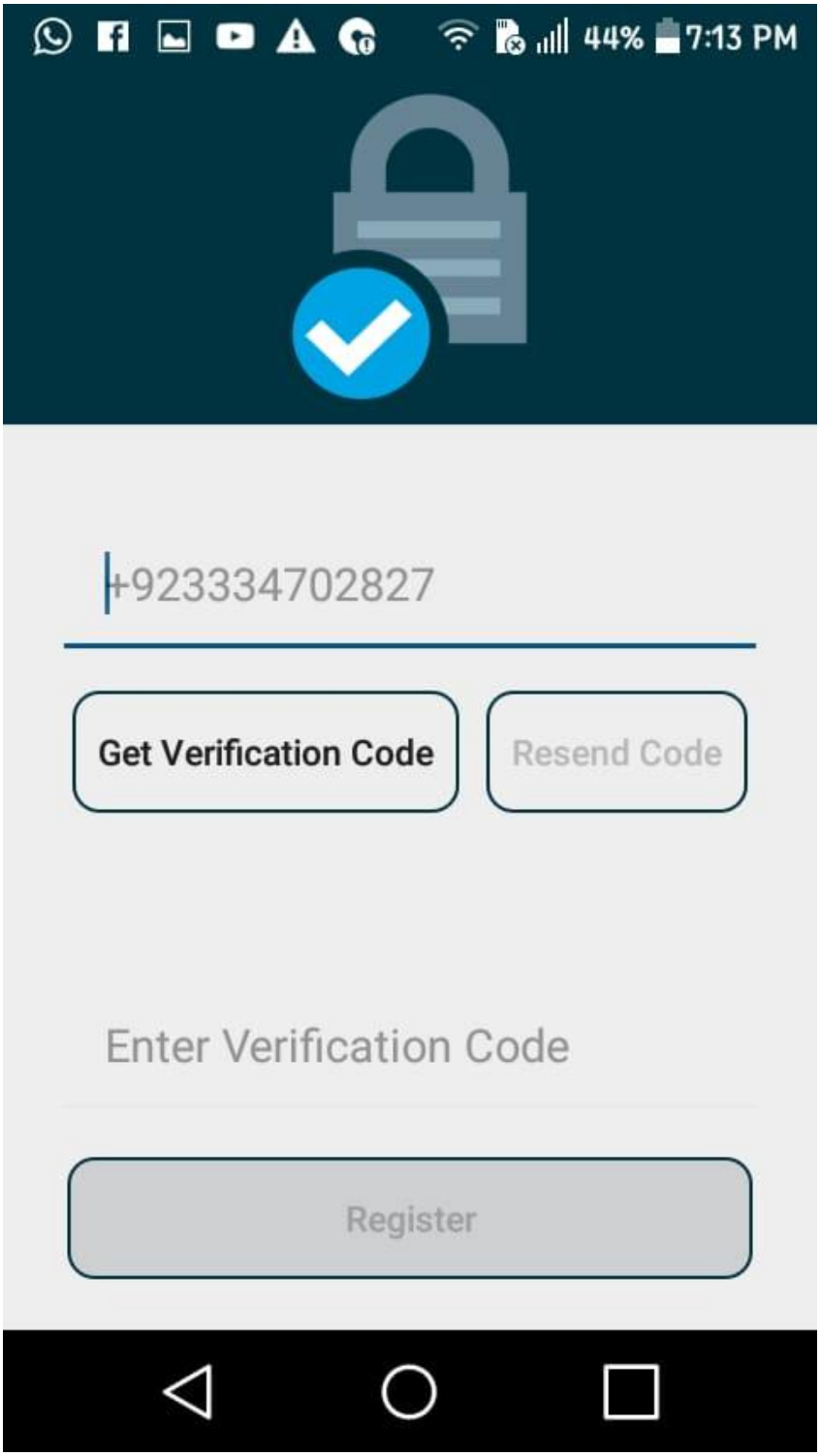

Figure 3.3: Prototype-2: After pressing the next button user will get onto the next requirement which is fill the blank with their contact no and press the button as shown in the prototype

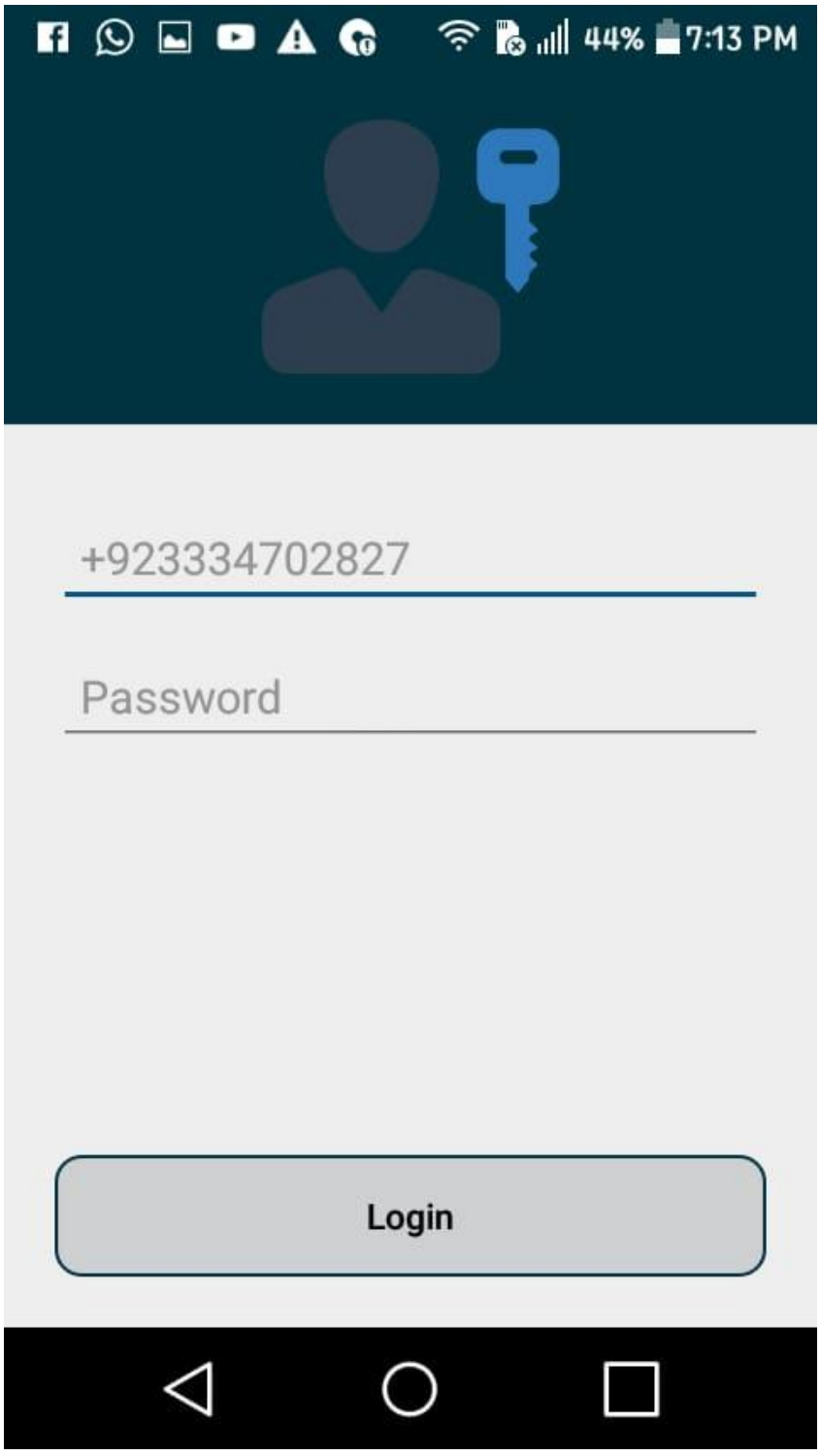

Figure 3.4: Prototype-3: In this Screen user simply enter the Username and Password and press the login button if user created an account.

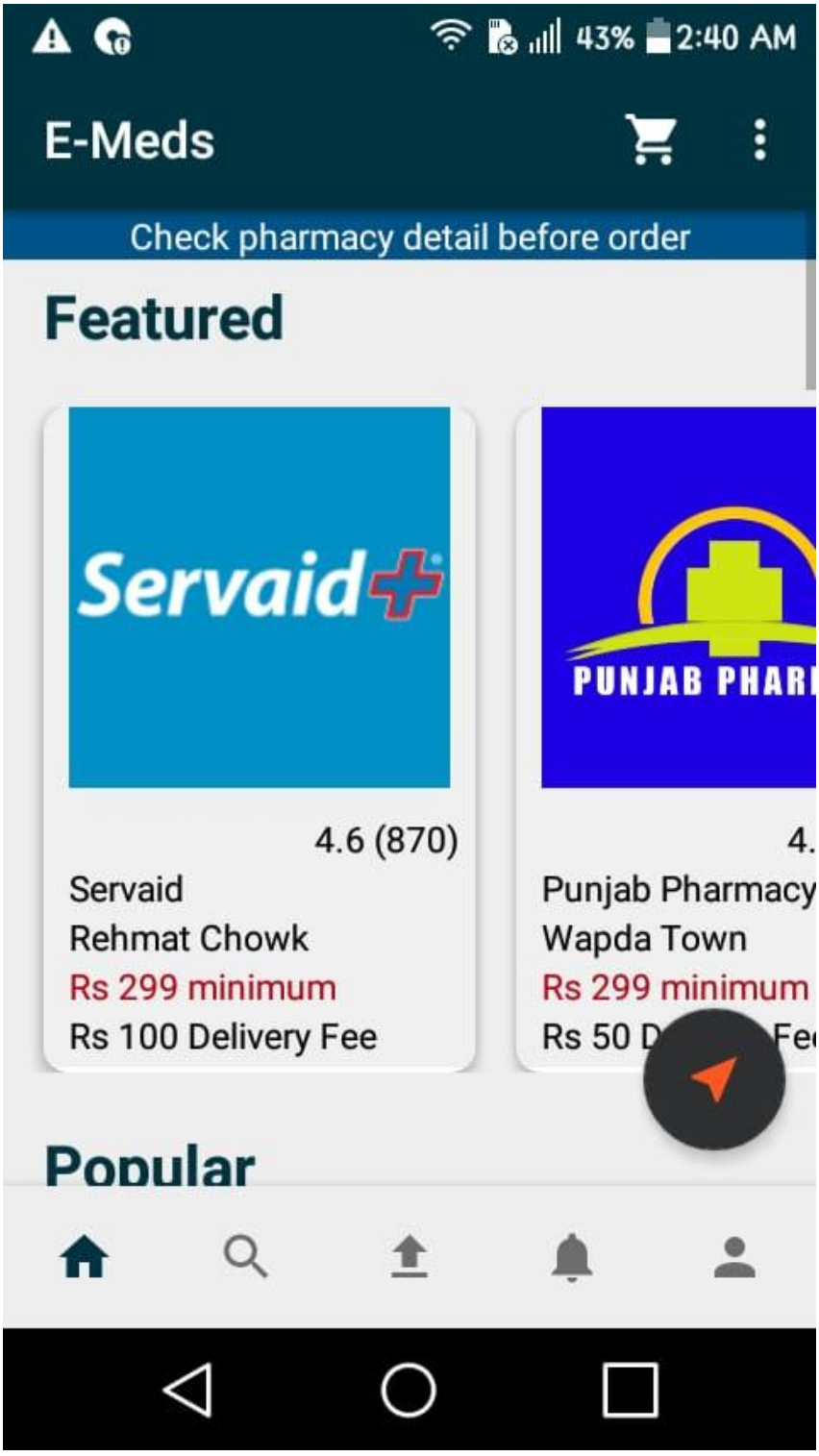

Figure 3.5: Prototype-4: After clicking the login button user will get the interface of the application where various medical store shown to the user where user can check the details about all the medical stores.

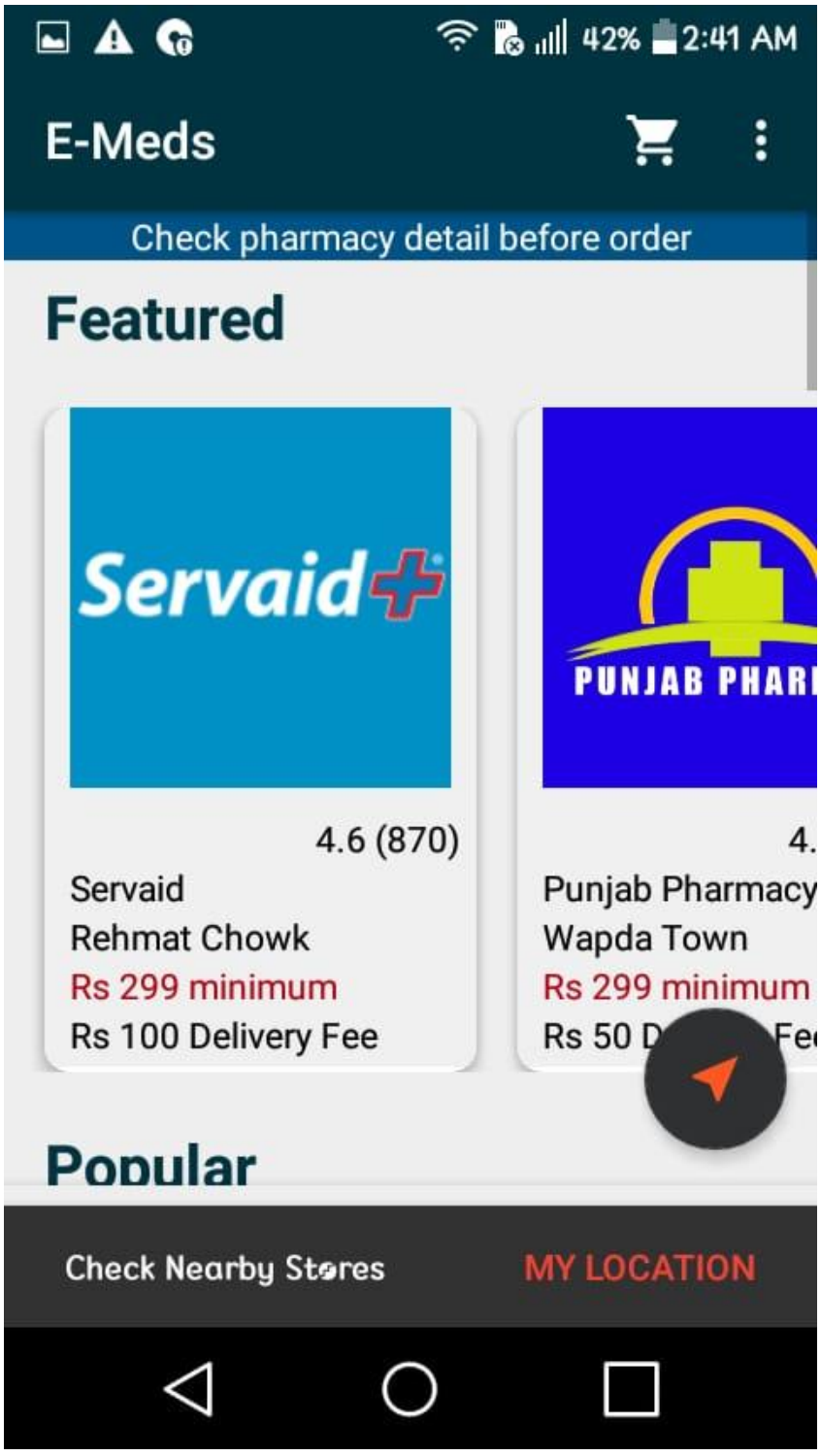

Figure 3.6: Prototype-5: In this prototype basically it is linked with prototype-4 where user can check details about medical stores on the other hand user can also check the nearby medical stores which can access easily by user.

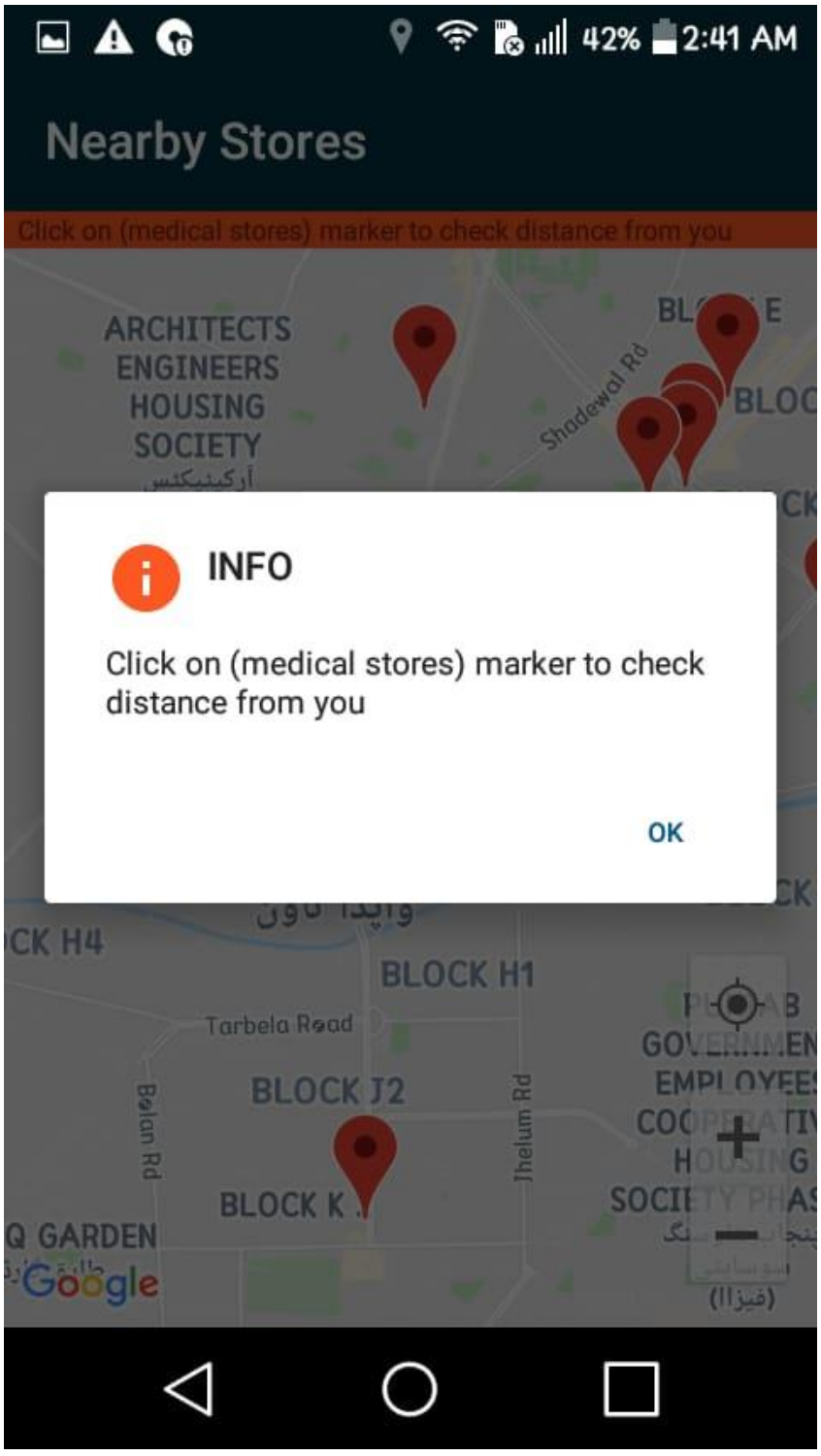

Figure 3.7: Prototype-6: In this prototype after selecting the medical store by user which is easily access by the user, user click the location button the check box where user get some info about location.

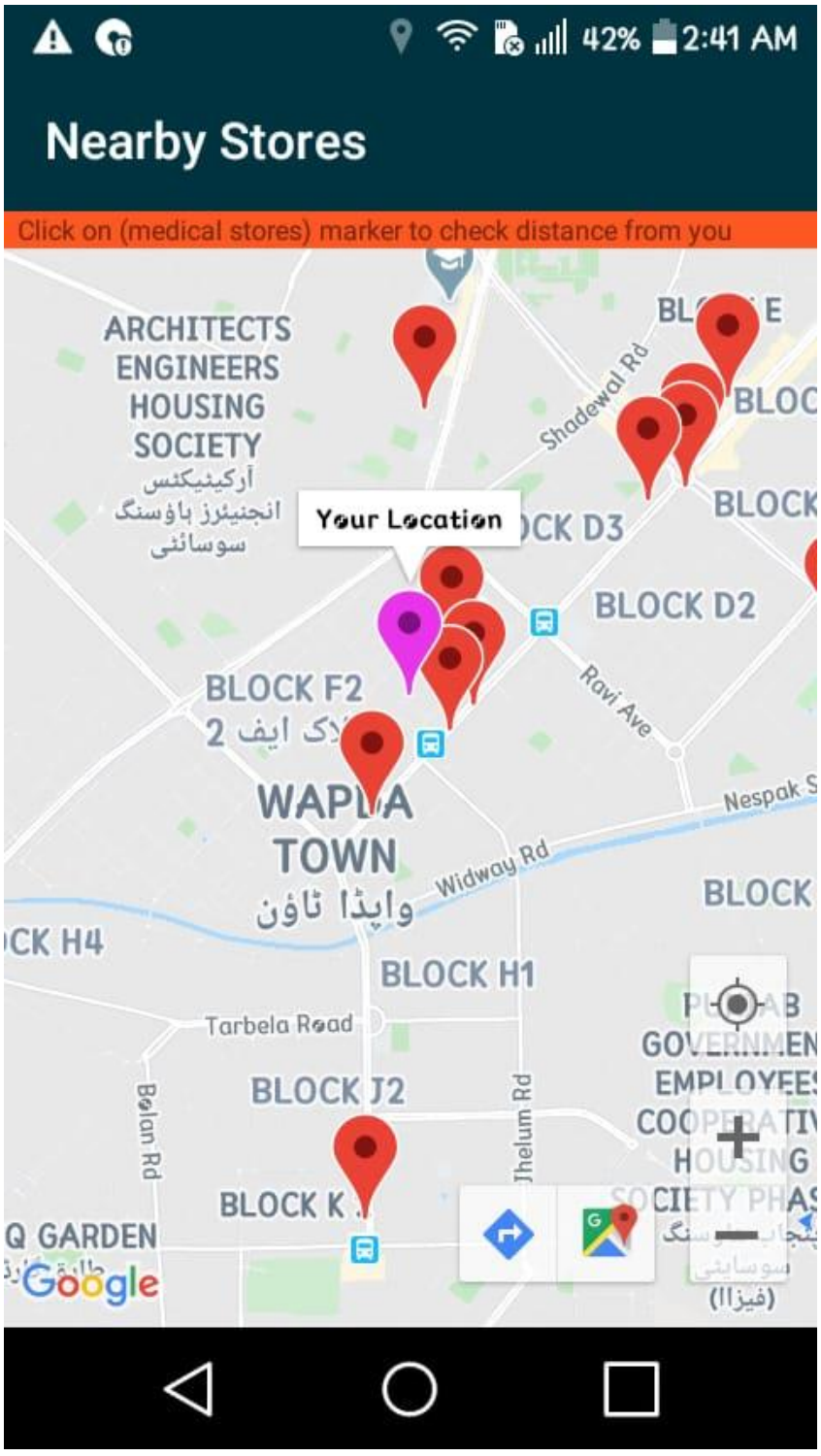

Figure 3.8: Prototype-7: In this prototype the screen is showing to the user the nearby stores where user can get their medicine easily.

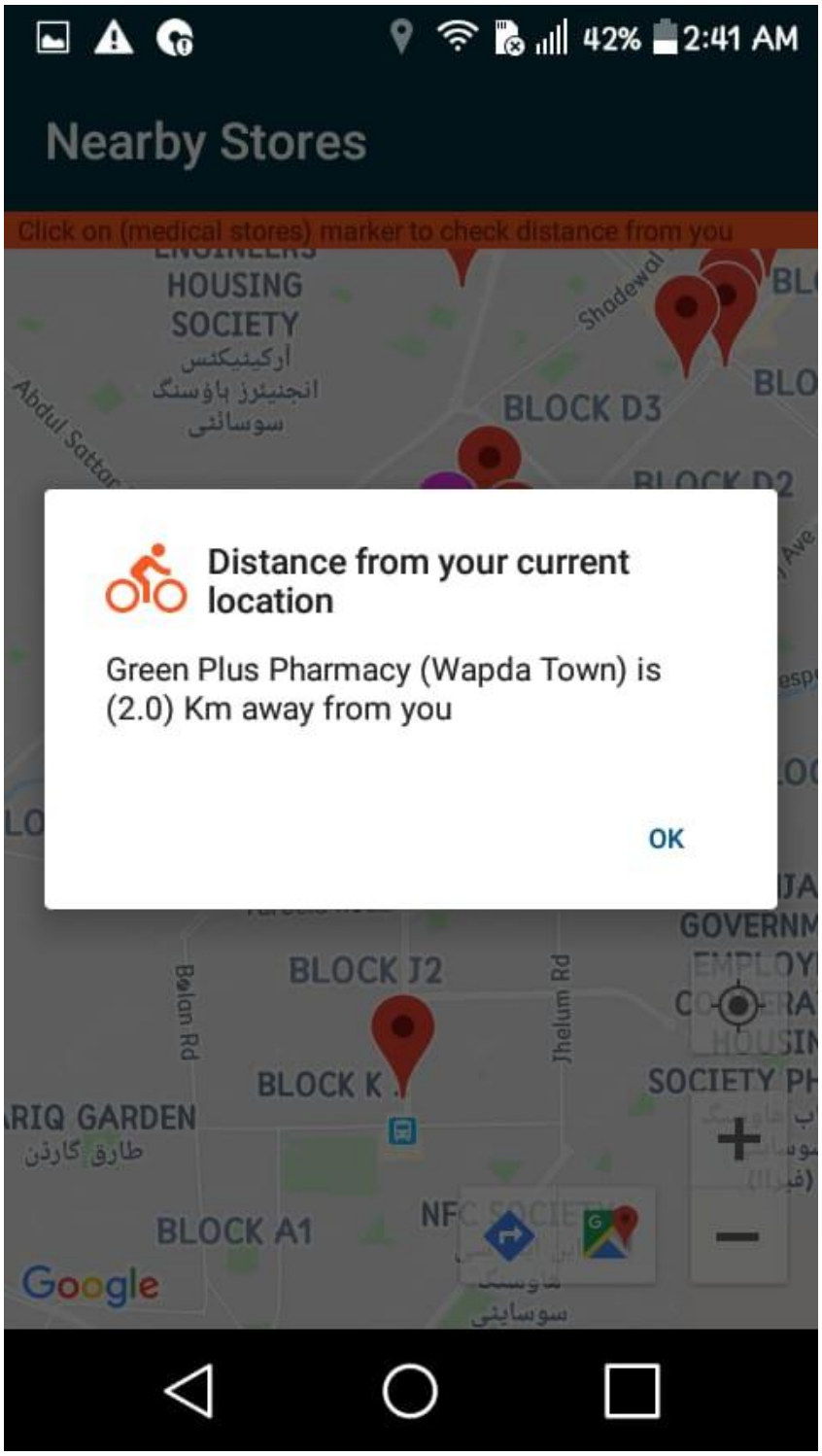

Figure 3.9: Prototype-8: In the same screen of prototype-7 after selecting the medical store the check box shown to the user is briefed that how much that selecting medical is far away from your destination.

Figure 3.10: Prototype-9: In this prototype after selecting nearby medical store user will get the screen where user can search their desire medicine or search about the medicine that the searched medicine is available on that medical store or not, after selecting medicine price and description of that medicine is also mention.

Figure 3.11: Prototype-10: In this prototype the quantity option and add to cart option shown to user, user can select the quantity as many as he want and select add to cart.

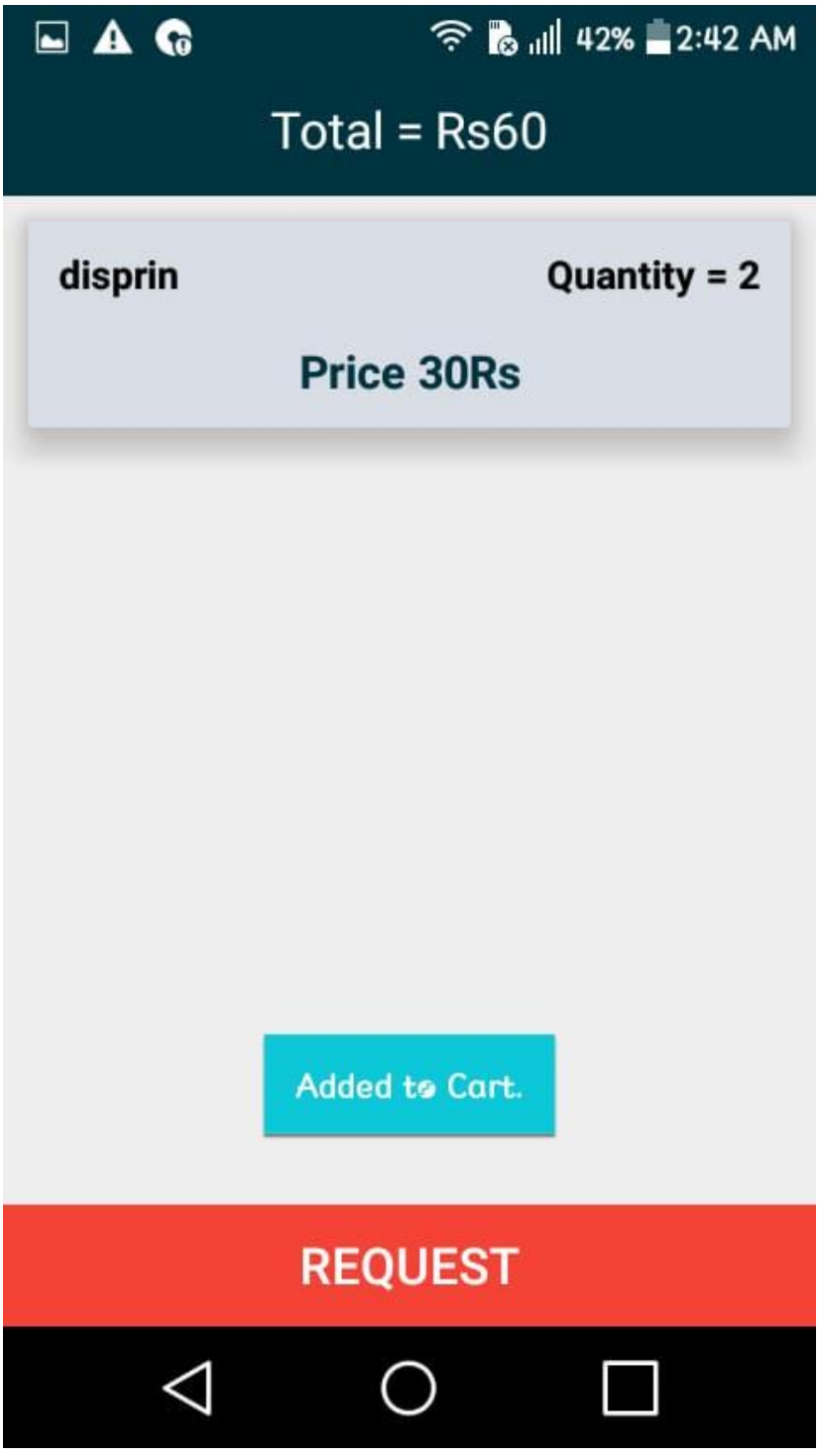

Figure 3.12: Prototype-11: In this prototype the screen is showing the total price and quantity which user selected to the user and press add to cart.

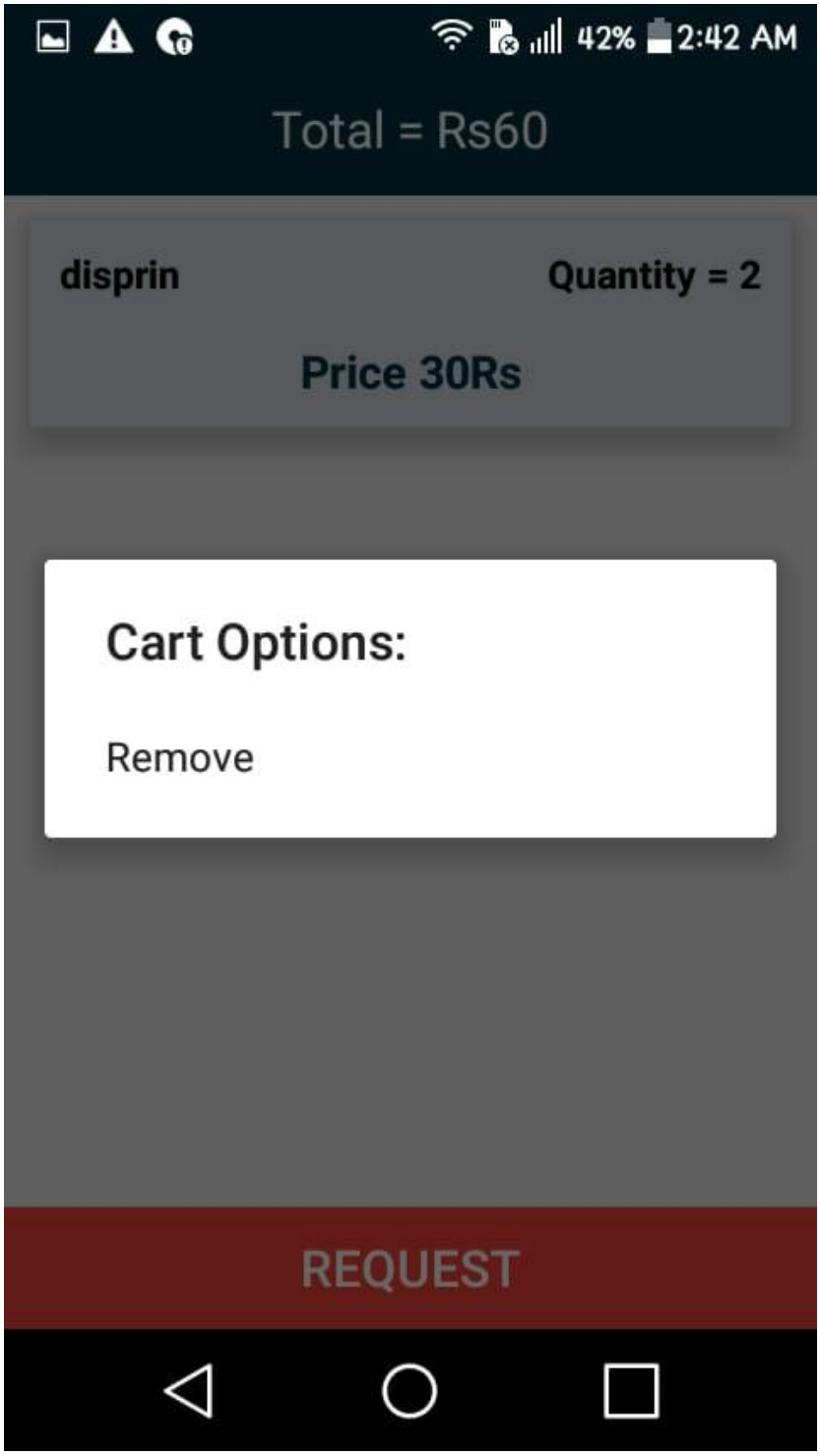

Figure 3.13: Prototype-12: In this prototype after clicking add to cart button the cart option is shown to user that you want remove the selected medicine from cart after ordering it.

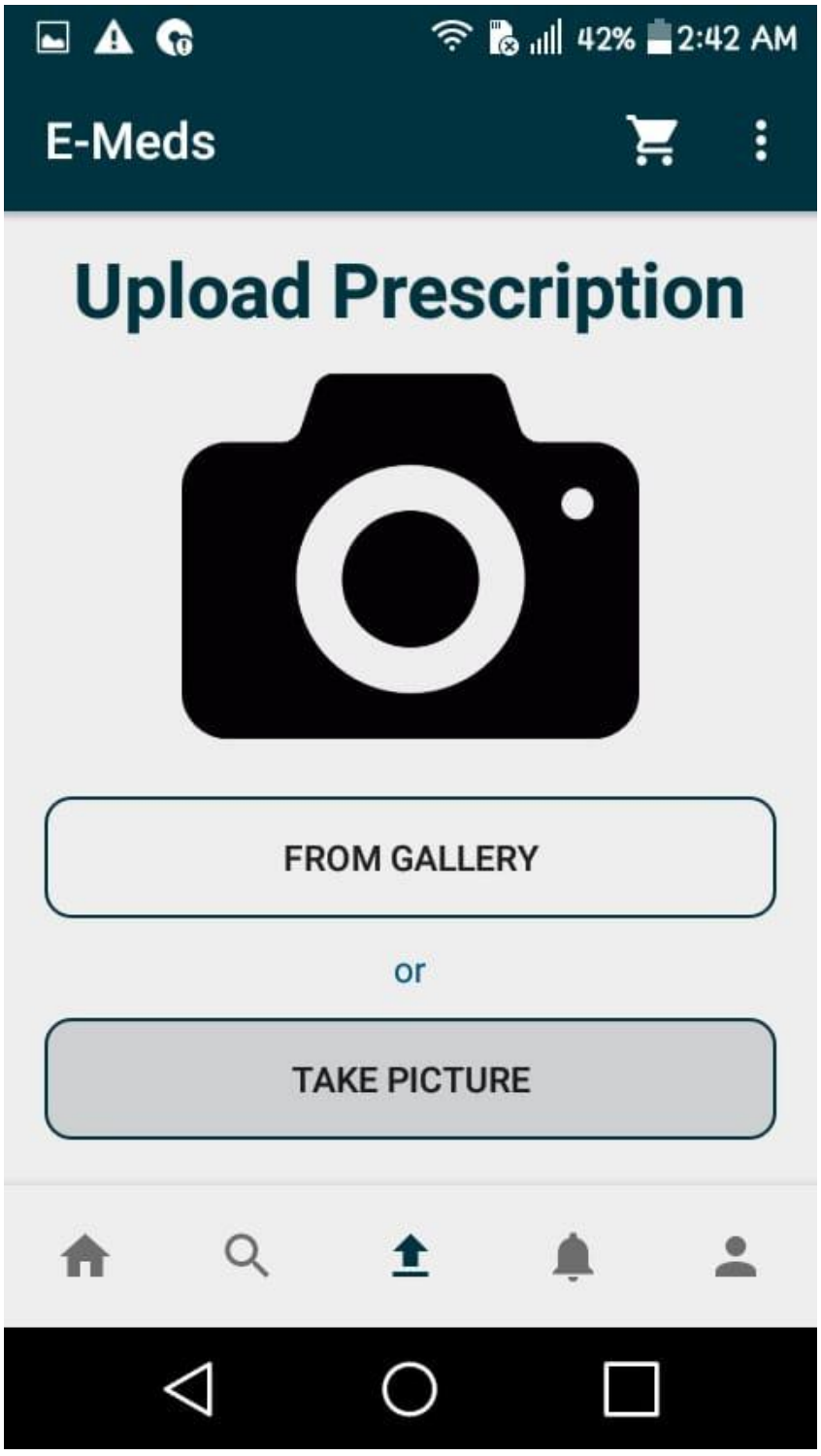

Figure 3.14: Prototype-13: In this prototype app is asking to user to select picture from gallery or take picture of your prescription.

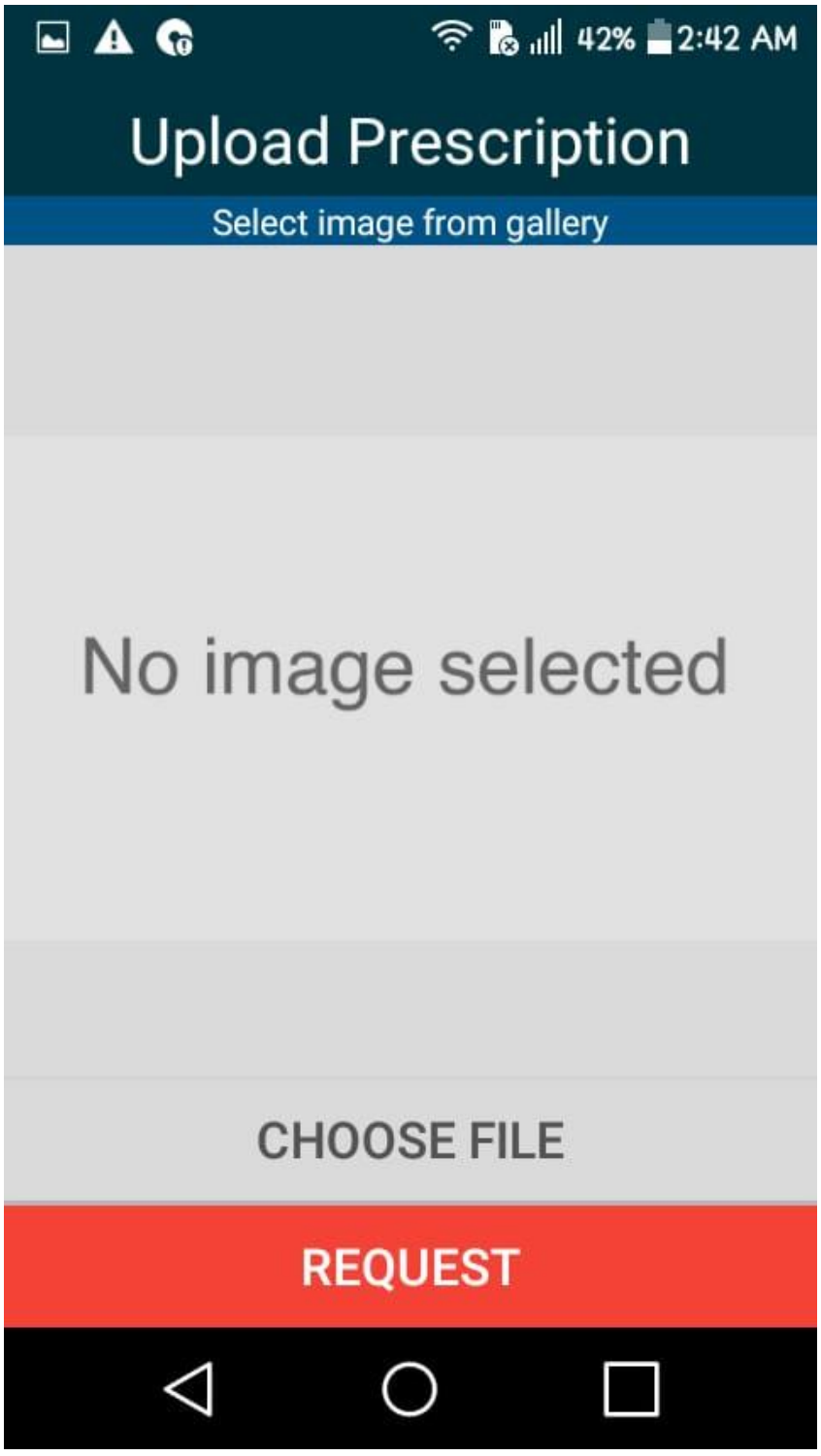

Figure 3.15: Prototype-14: In this prototype after clicking the button from gallery from prototype-13 the screen is shown to user is that no image selected or chooses a file and press request to medical store.

Figure 3.16: Prototype-15: In this prototype the same screen is showing to user except option, take a picture option is shown to user rather than choose a file after clicking the button take a picture from prototype-13 and send request to selected medical store.

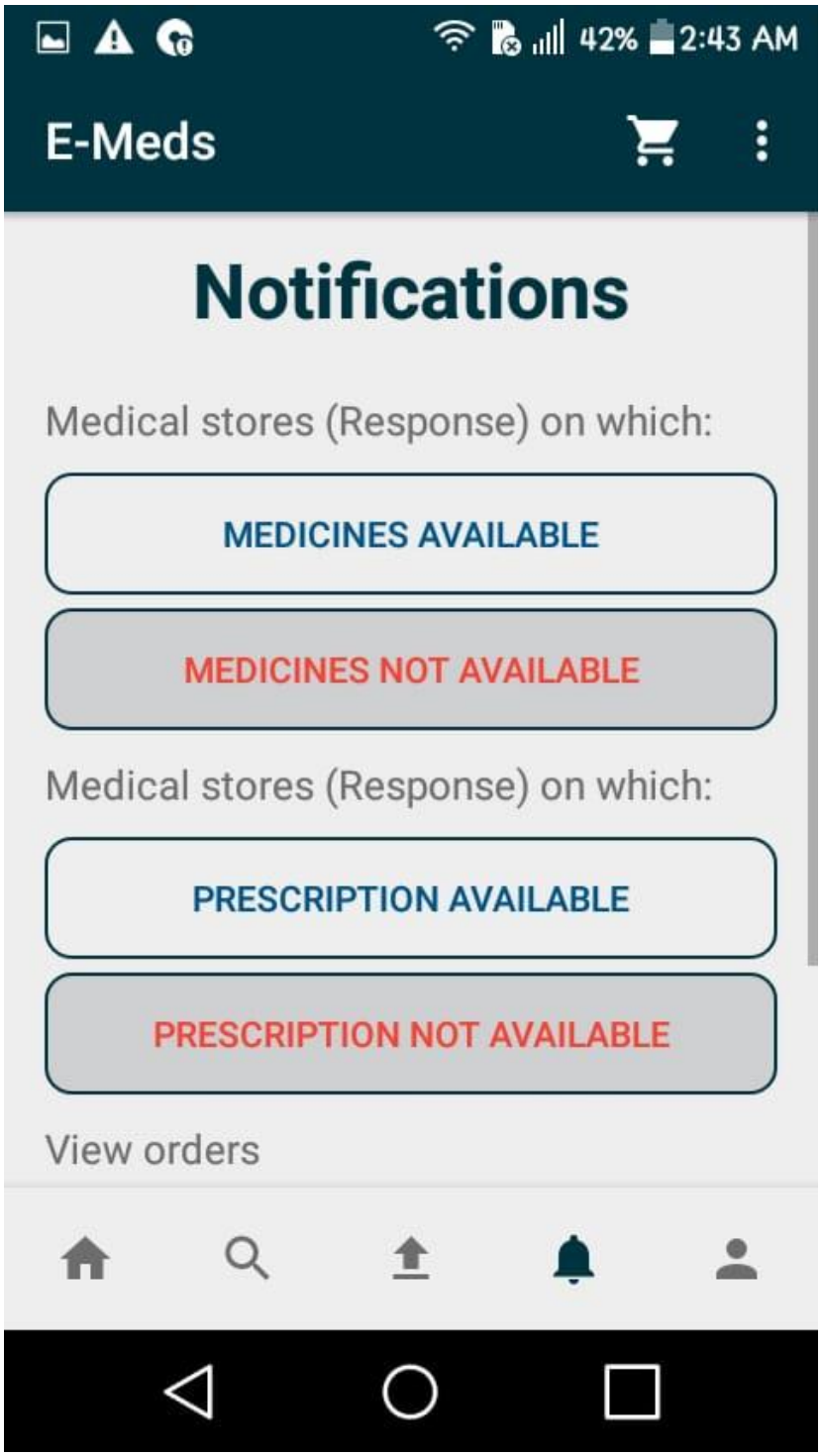

Figure 3.17: Prototype-16: In this prototype the medical response against the order that your medicine or prescribed medicine is available or not basically it shows the availability of medicine or prescription to user.

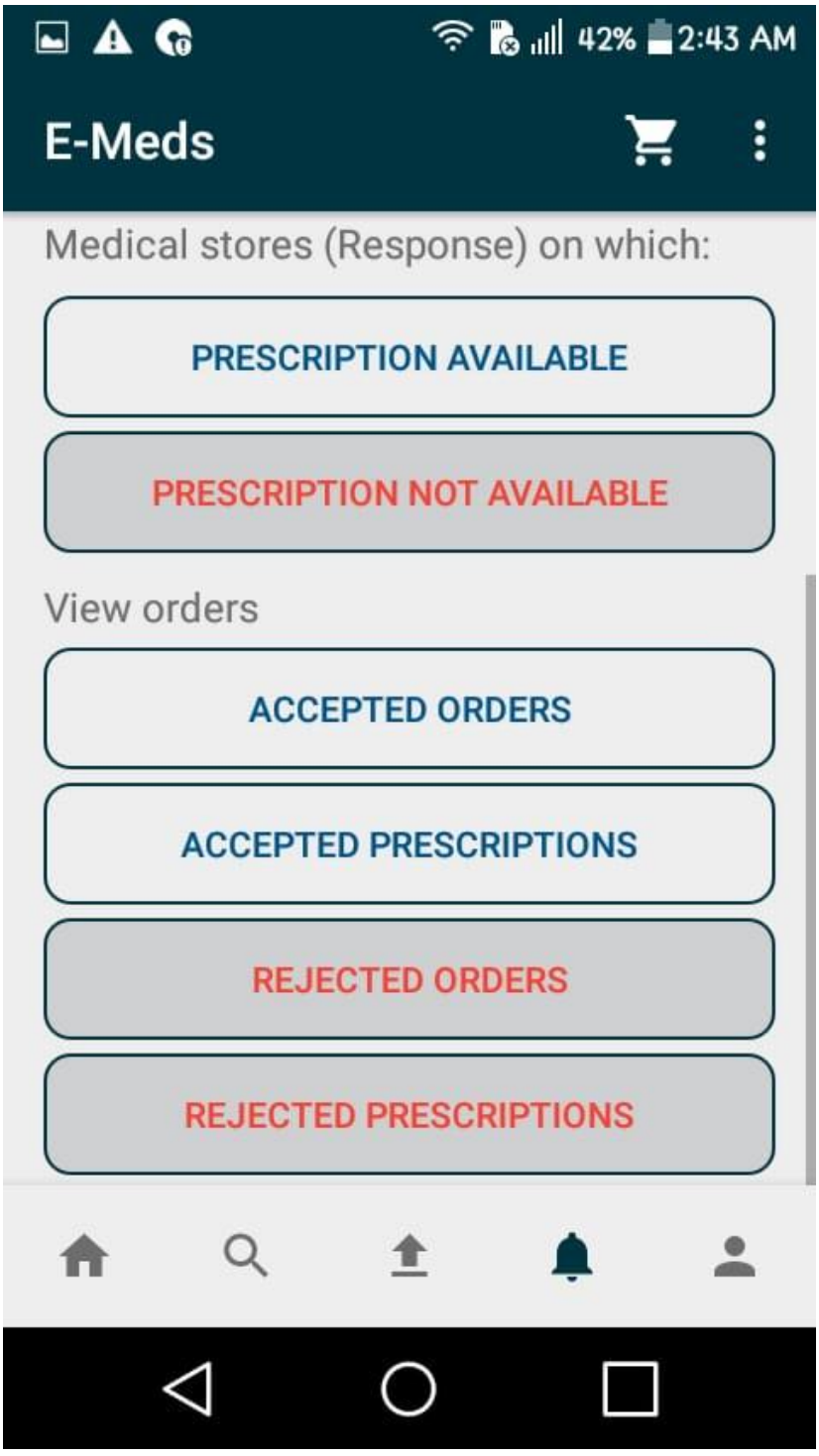

Figure 3.18: Prototype-17: In this prototype the medical response against the order that your medicine or prescribed medicine is available or not basically it shows the availability of medicine or prescription to user.

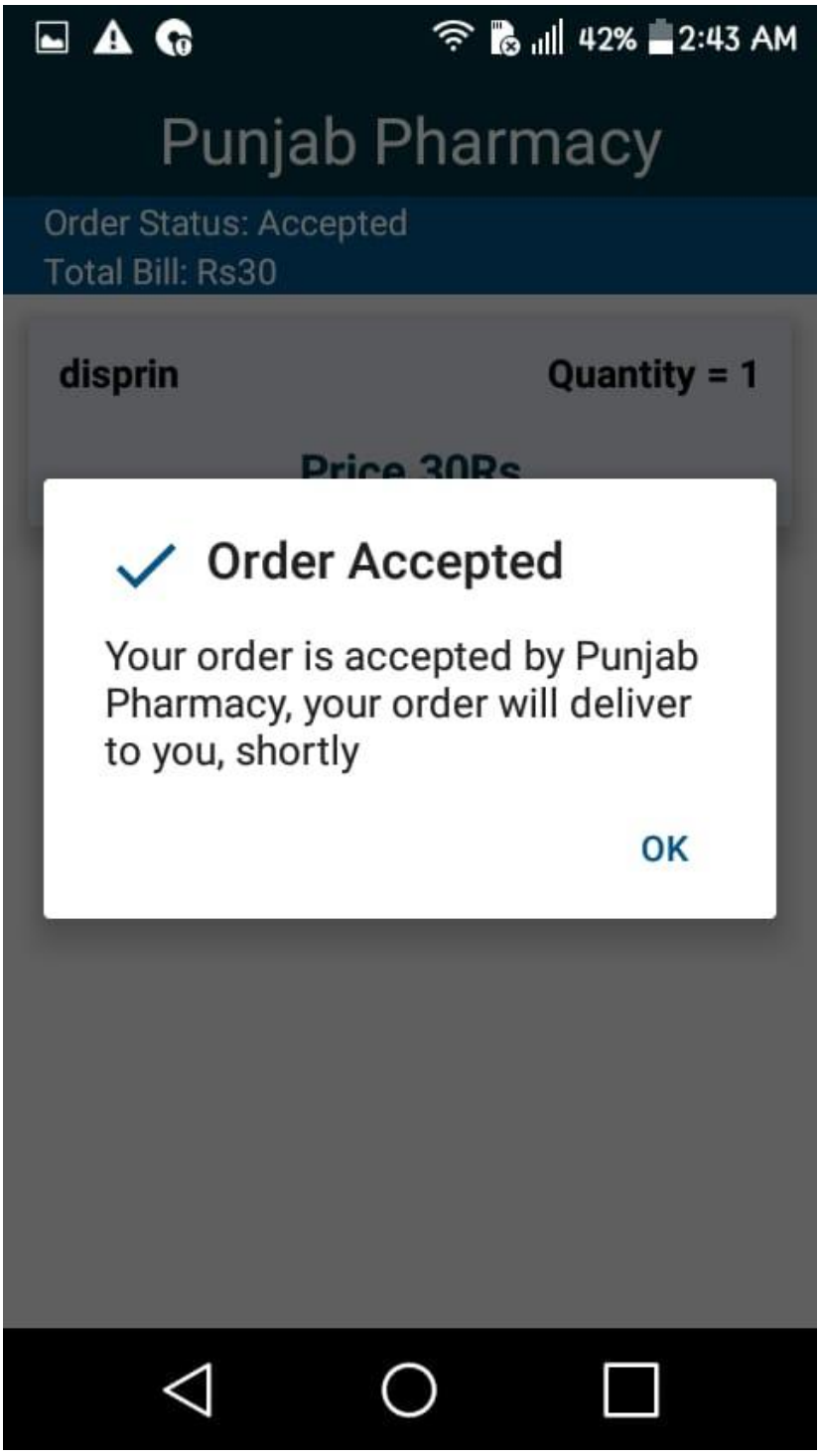

Figure 3.19: Prototype-18: In this prototype after all the ordering procedure of medicine the app show up with the message box that your order is accepted by the selected medical store and it'll shortly delivered to your destination.

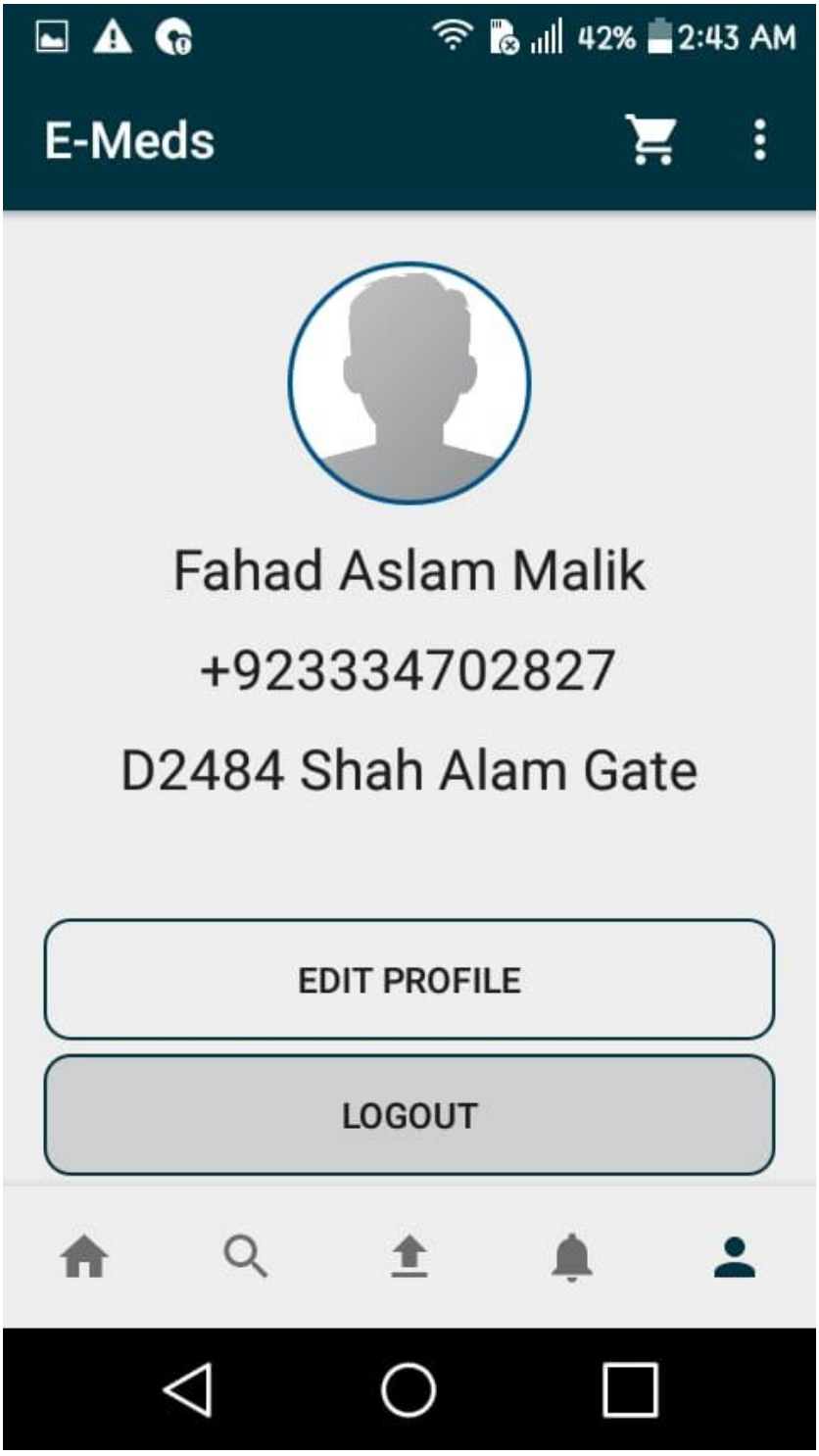

Figure 3.20: Prototype-19: In this prototype the user profile is shown his/her name, contact# and address user can also edit their profile information or log out like othe applications.

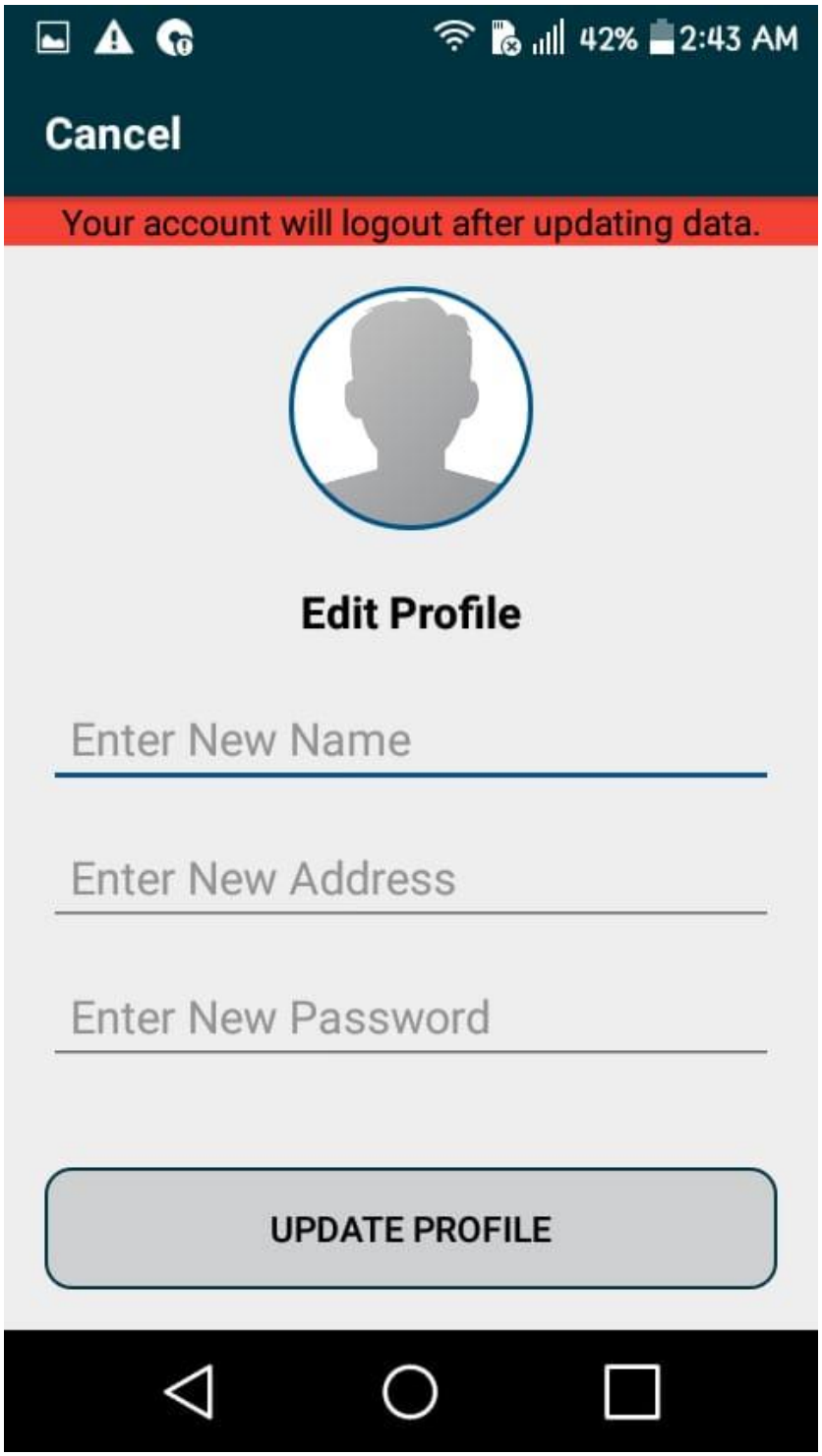

Figure 3.21: Prtotype-20: in this prototype the edit form is shown to user that after editing the profile the user automatically logged out from the app in case if he clicked the edit button in prototype-19.

CHAPTER # 4

4. DATA FLOW DIAGRAM

4.1 Data Flow Diagram Level 0

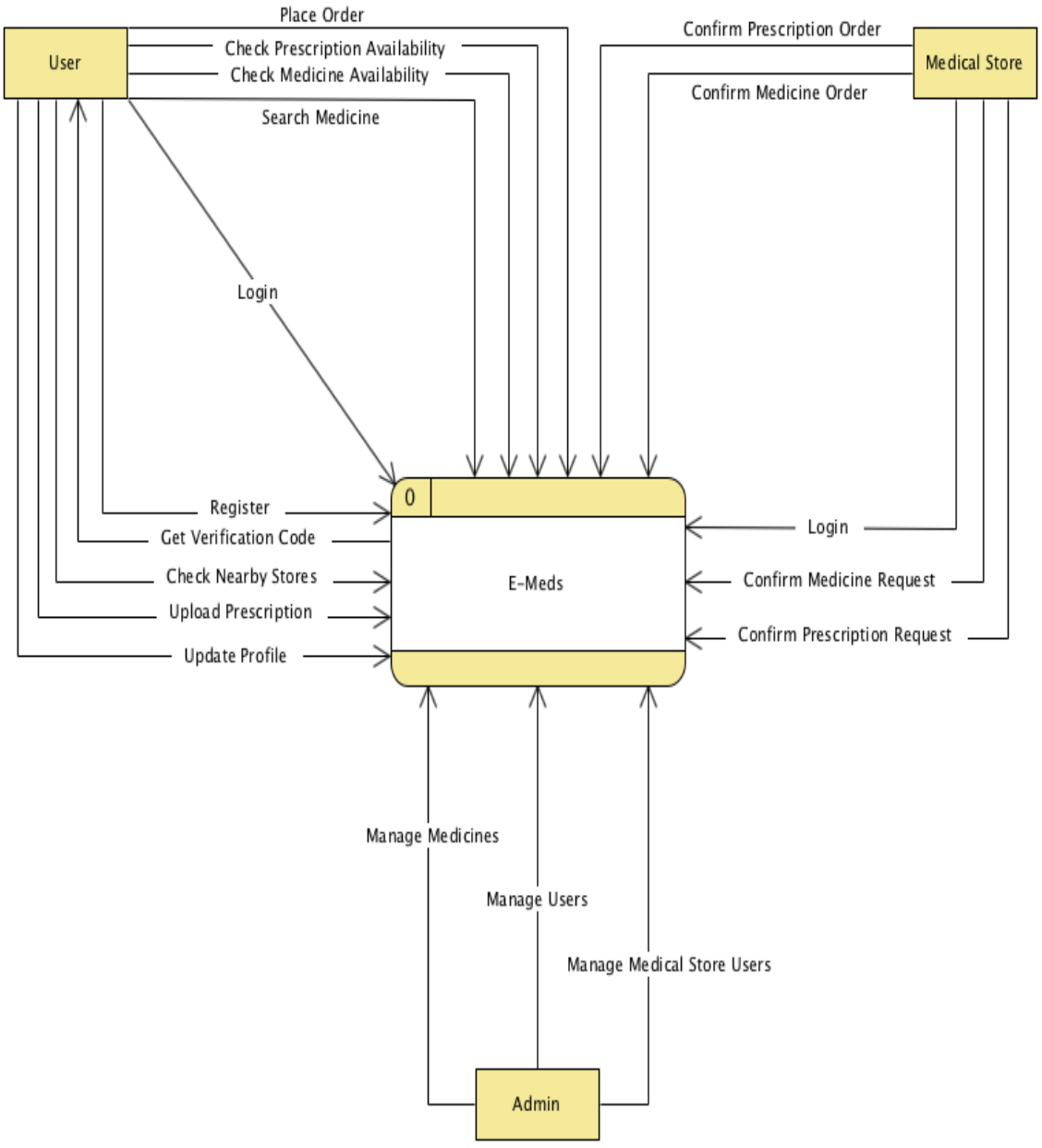

Figure 4.1: DFD (Level-0) - This figure describes the overall flow of data in our system at very basic level, that define the role of every stakeholder in the system

4.2 Data Flow Diagram Level 1

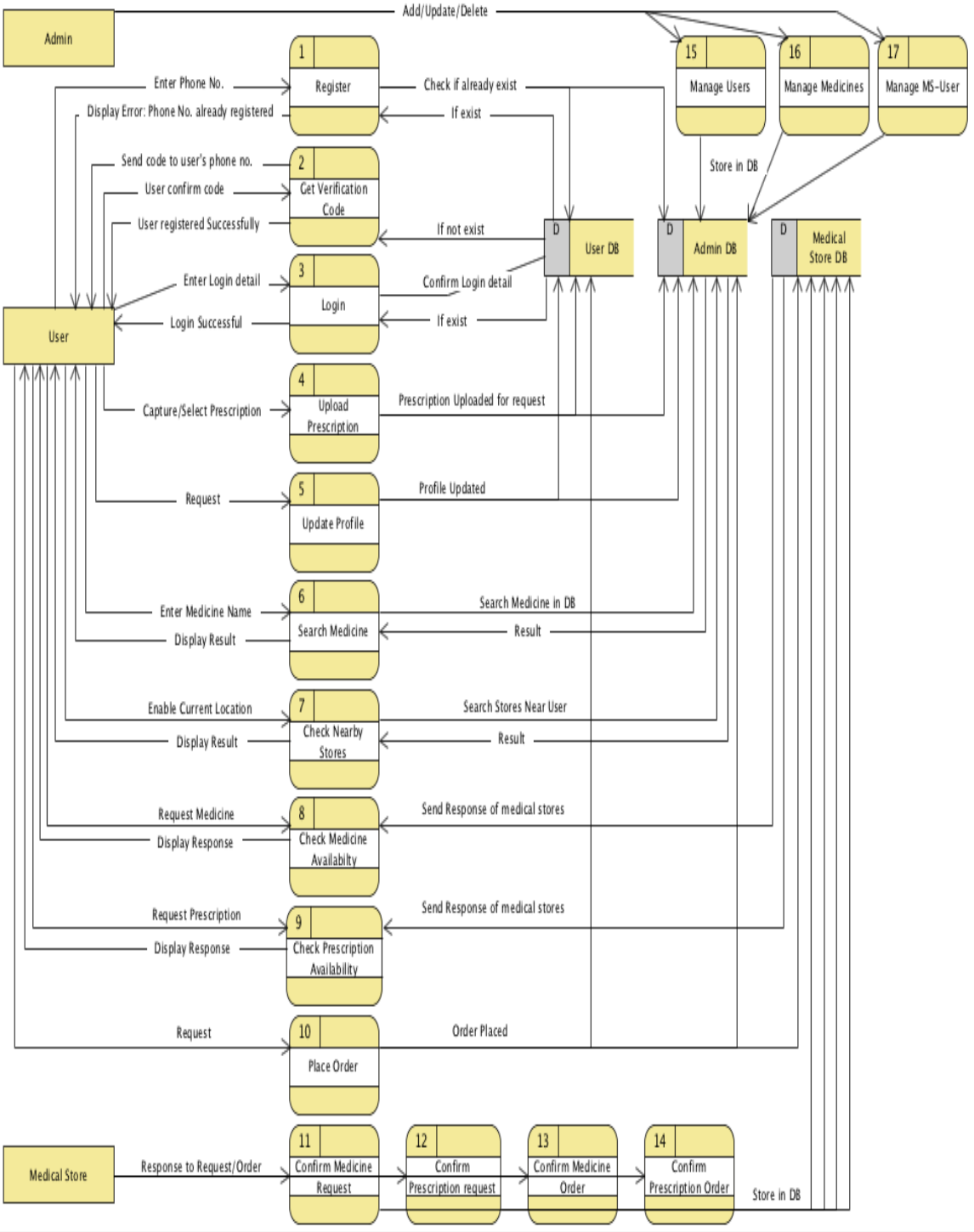

Figure 4.2: DFD (Level-1) - This figure describes the overall flow of data in our system at some advance level, that define every process in the system

4.3 Data Flow Diagram Level 2

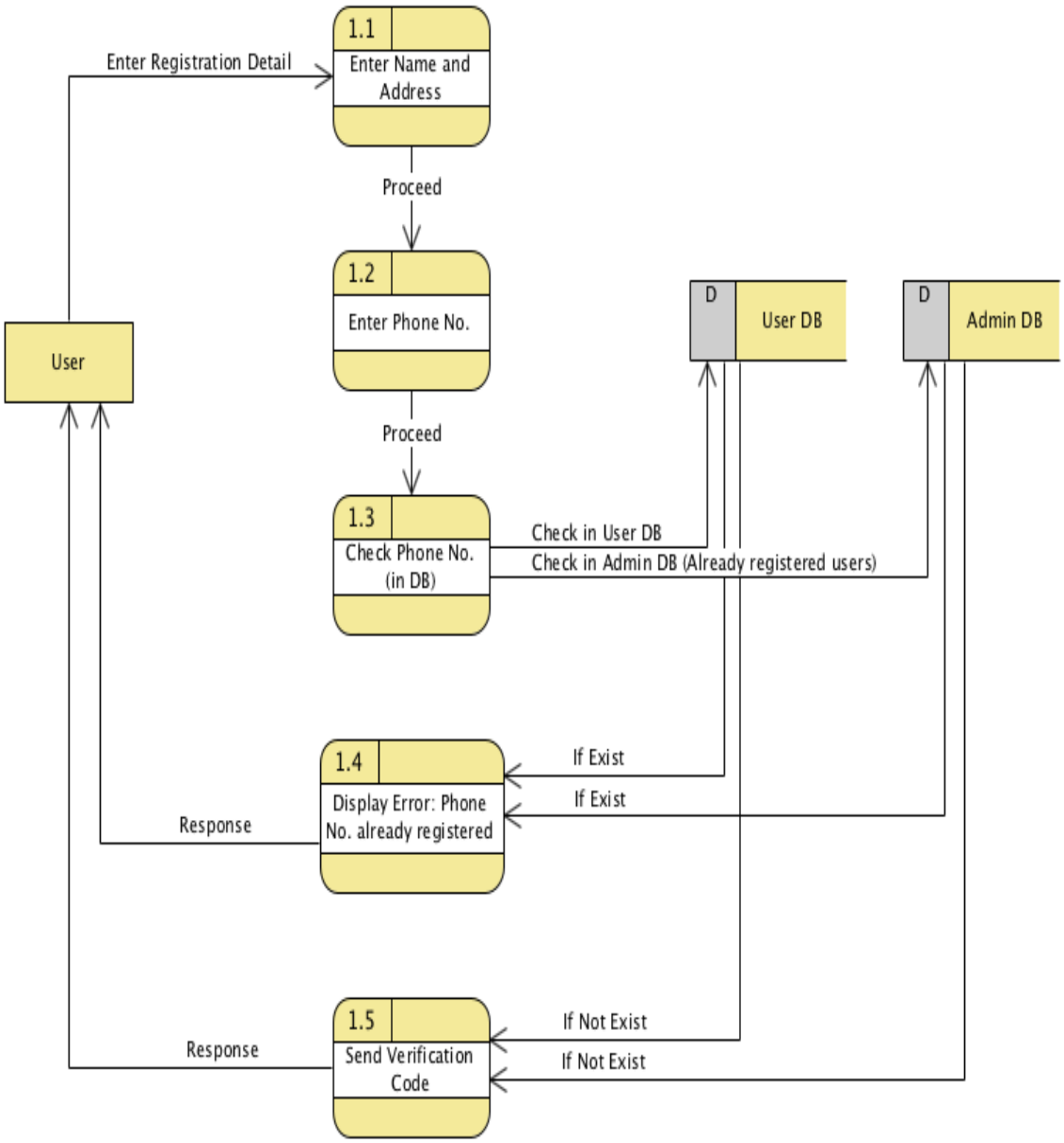

Figure 4.3: DFD (Level-2) - Register - This figure describes the register process in the system, that how data flow when registeration process is done in the system

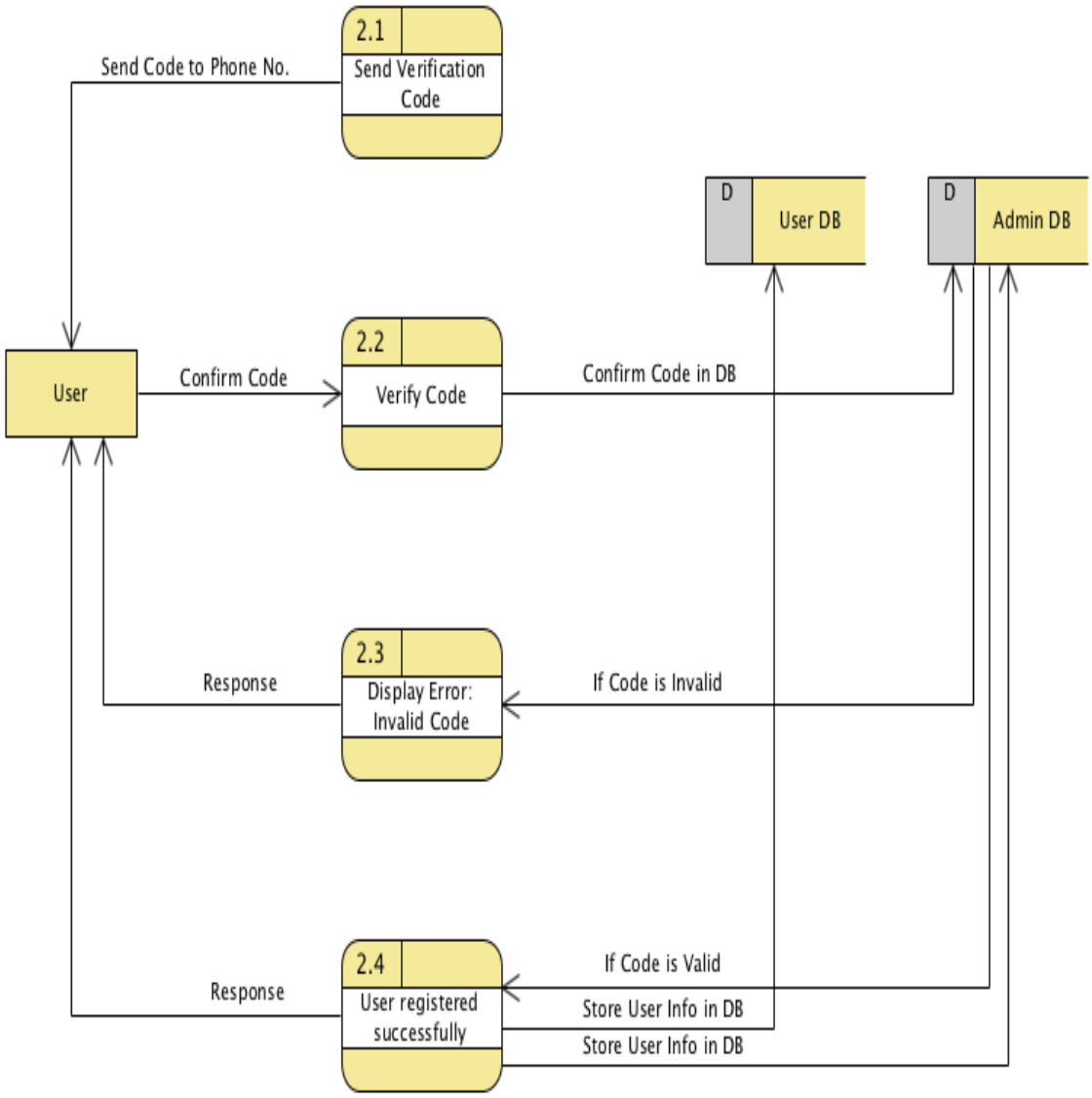

Figure 4.4: DFD (Level-2) – Get Verification Code - This figure describes the get verification code process in the system, that how data flow when user get verification code through OTP in the system

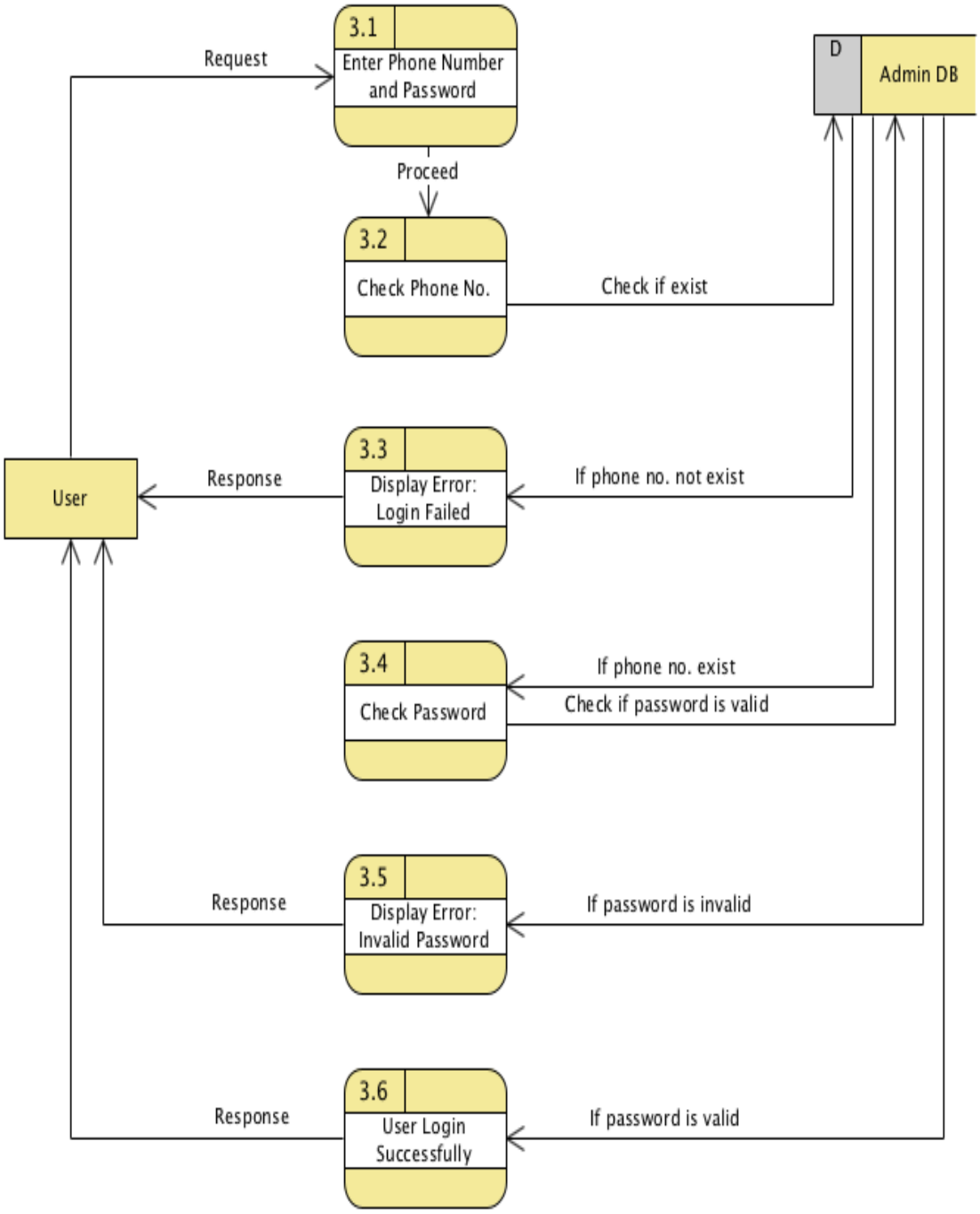

Figure 4.5: DFD (Level-2) – Login - This figure describes the login process in the system, that how data flow when login process is done in the system

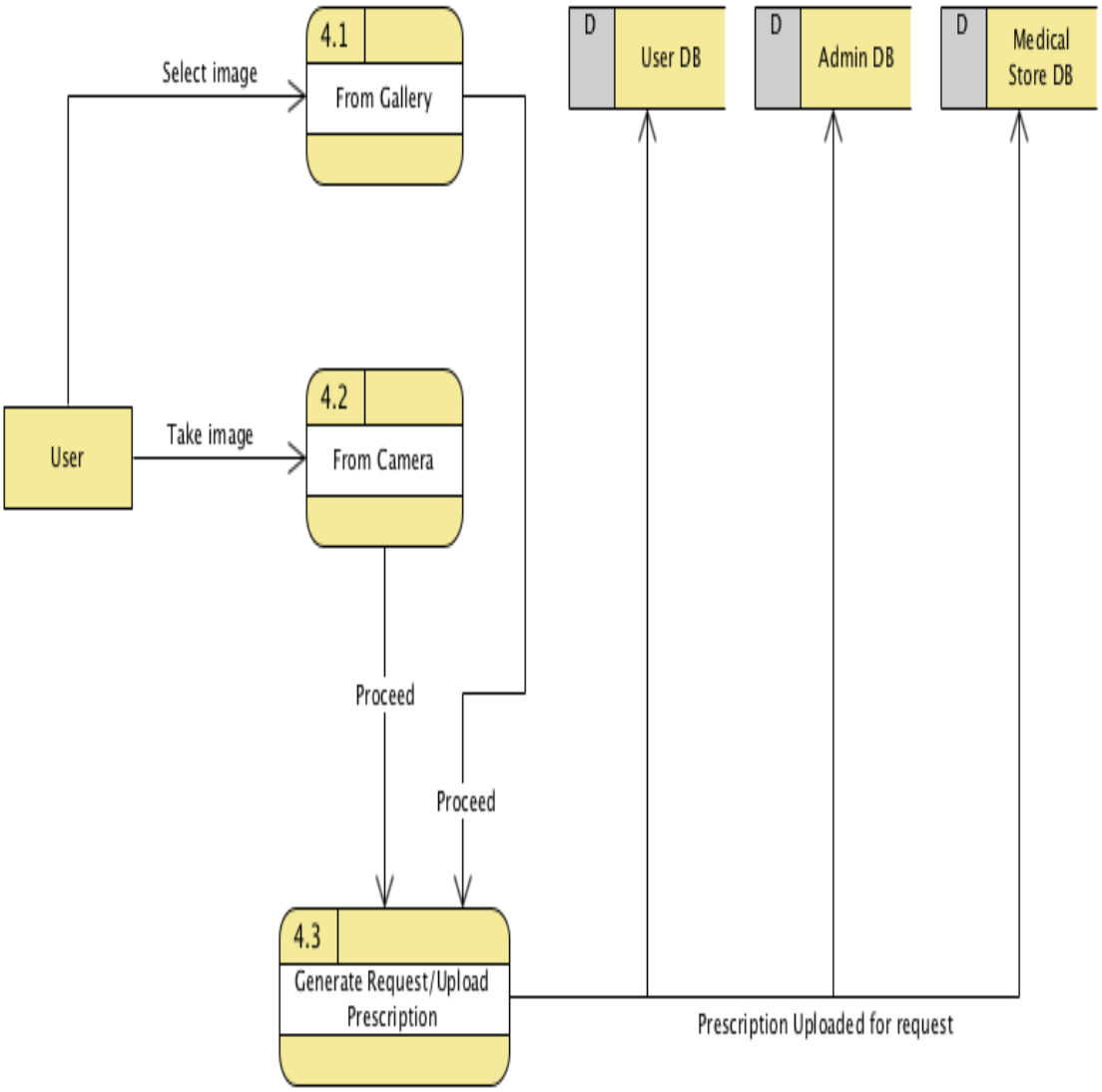

Figure 4.6: DFD (Level-2) – Upload Prescription - This figure describes the upload prescription process in the system, that how data flow when user upload prescription in the system

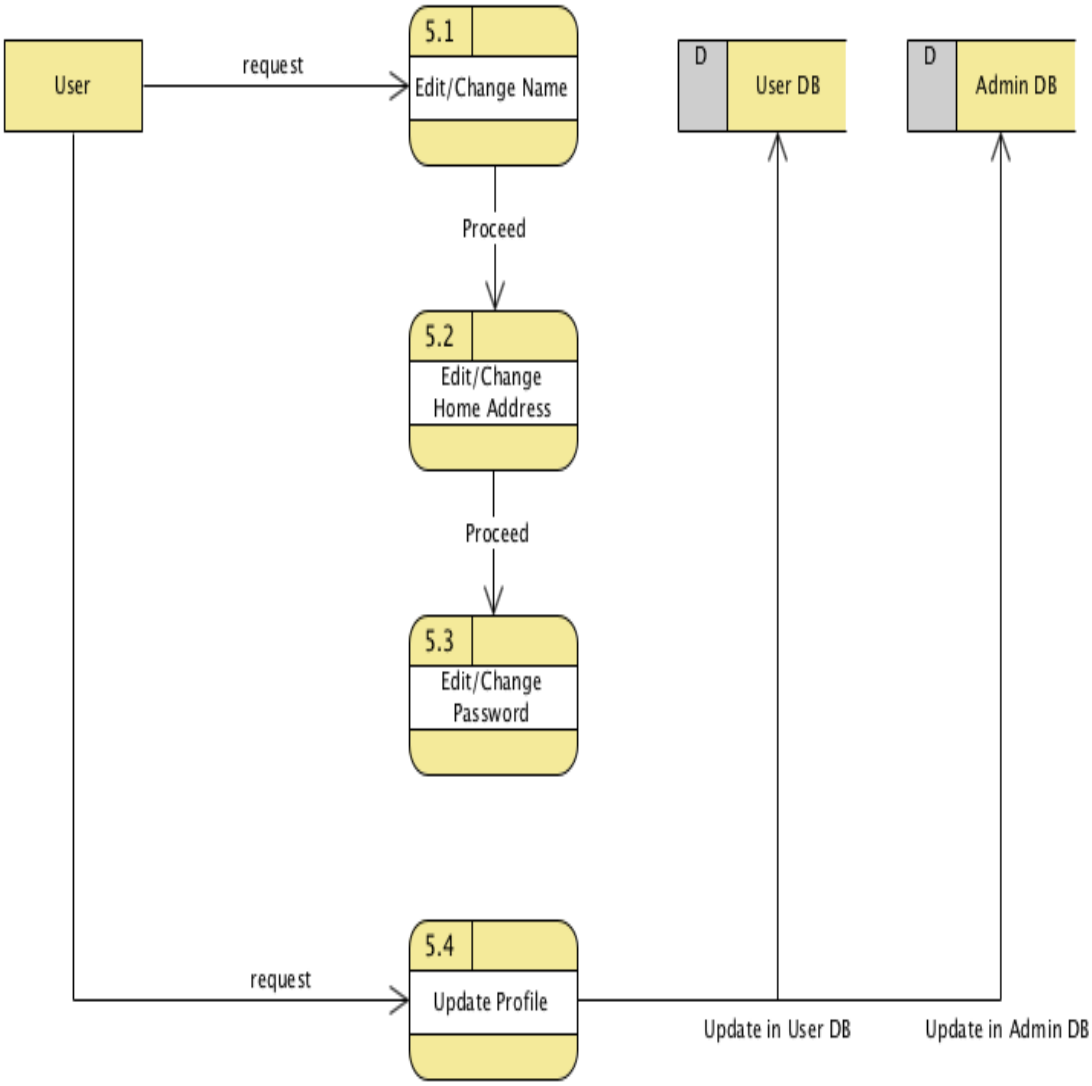

Figure 4.7: DFD (Level-2) – Update Profile - This figure describes the update profile process in the system, that how data flow when user update his/her profile in the system

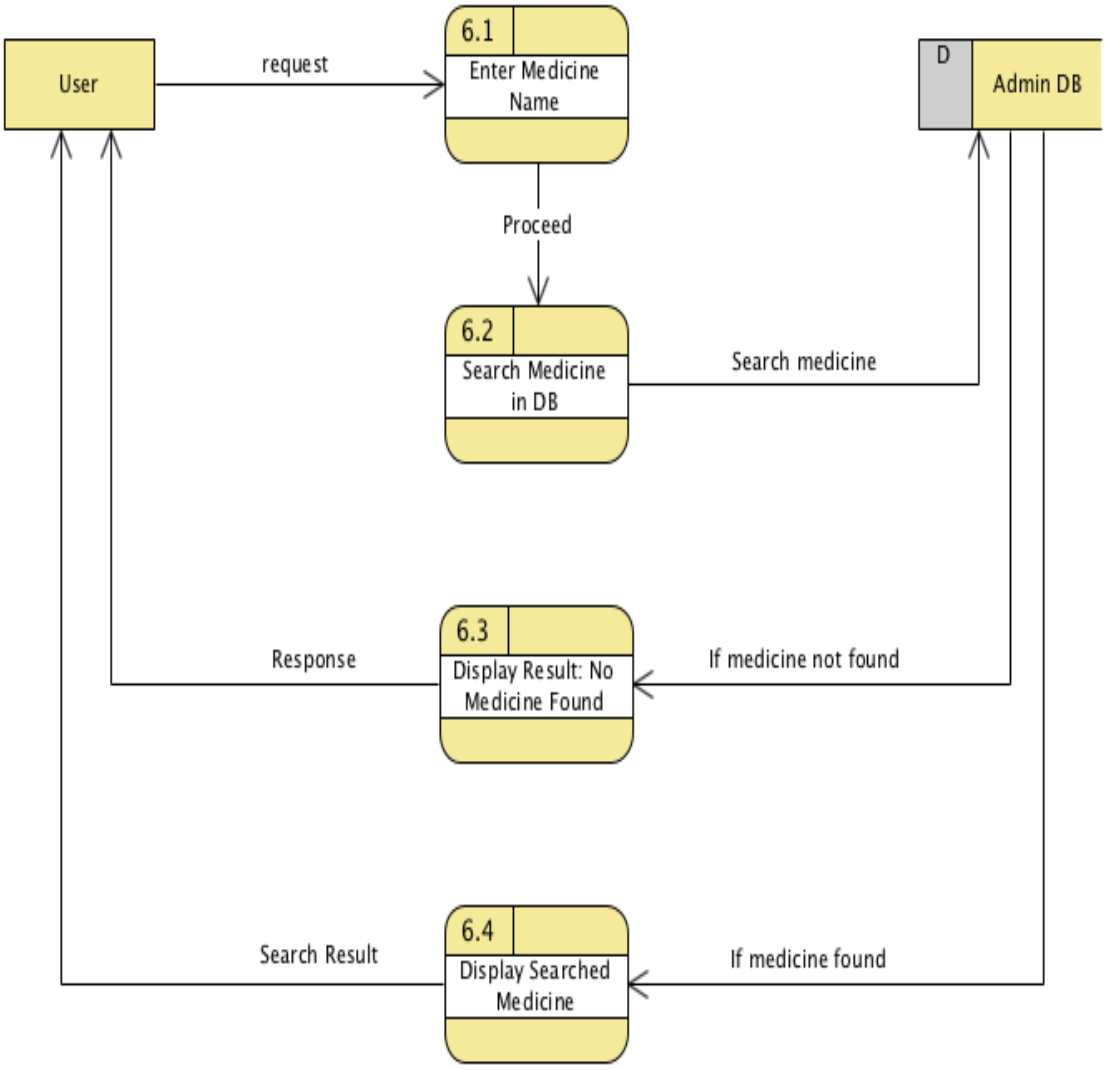

Figure 4.8: DFD (Level-2) – Search Medicine - This figure describes the search medicine process in the system, that how data flow when user search medicine in the system

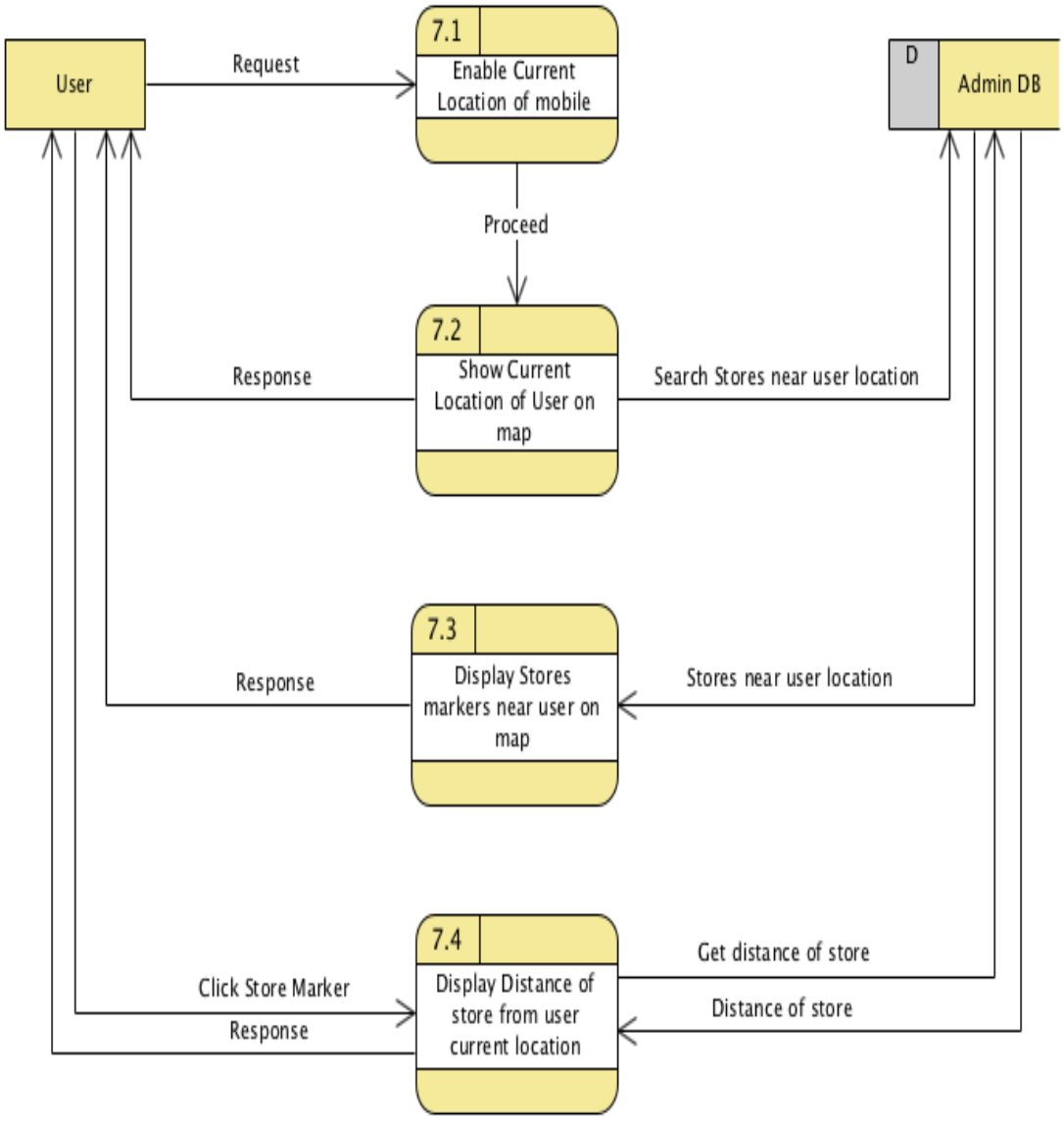

Figure 4.9: DFD (Level-2) – Check Nearby Store - This figure describes the check nearby store process in the system, that how data flow when user want to see nearby store at his/her location

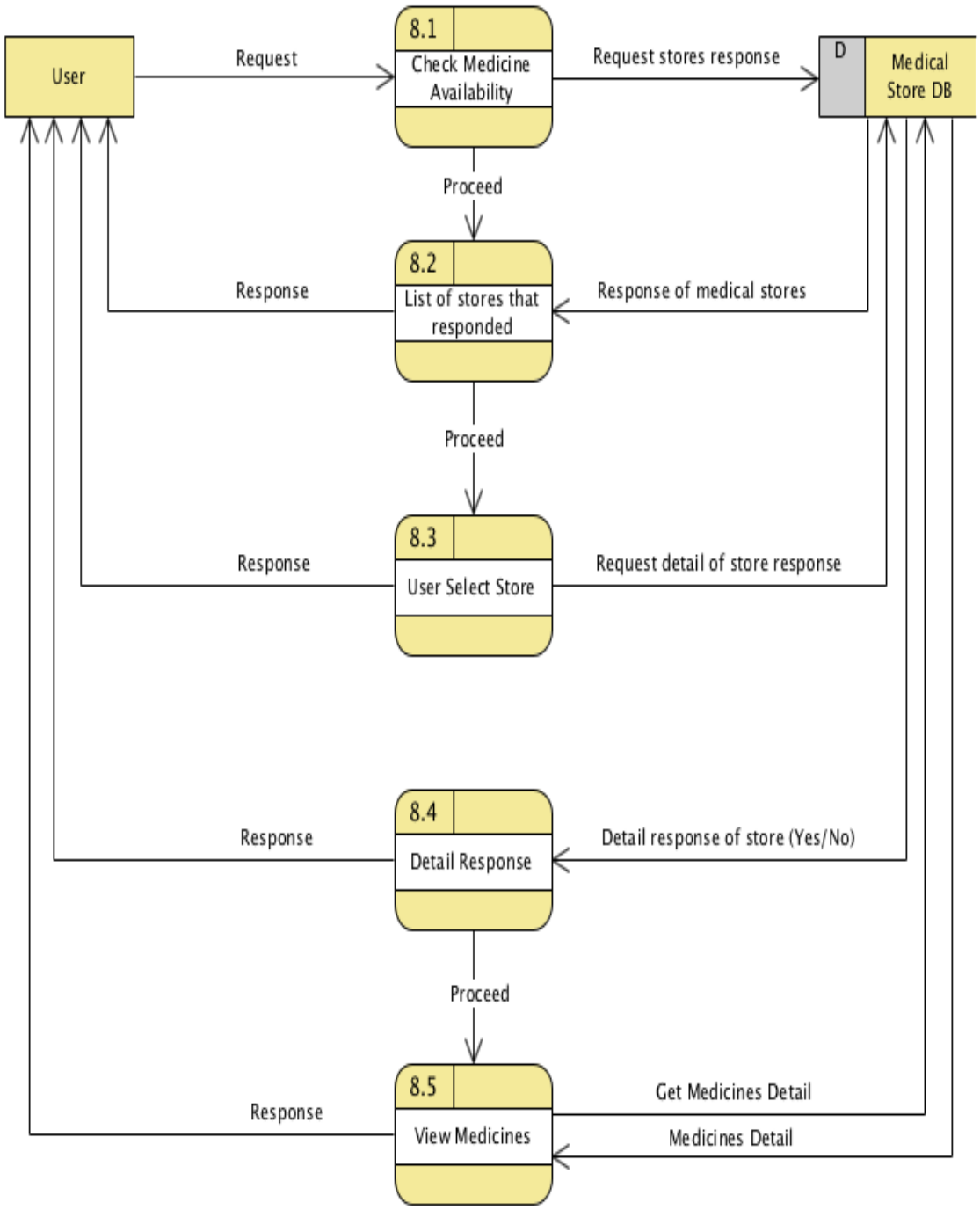

Figure 4.10: DFD (Level-2) – Check Medicine Availability - This figure describes the check medicine availability process in the system, that how data flow when user want to check the medicine availability at their desired medical store

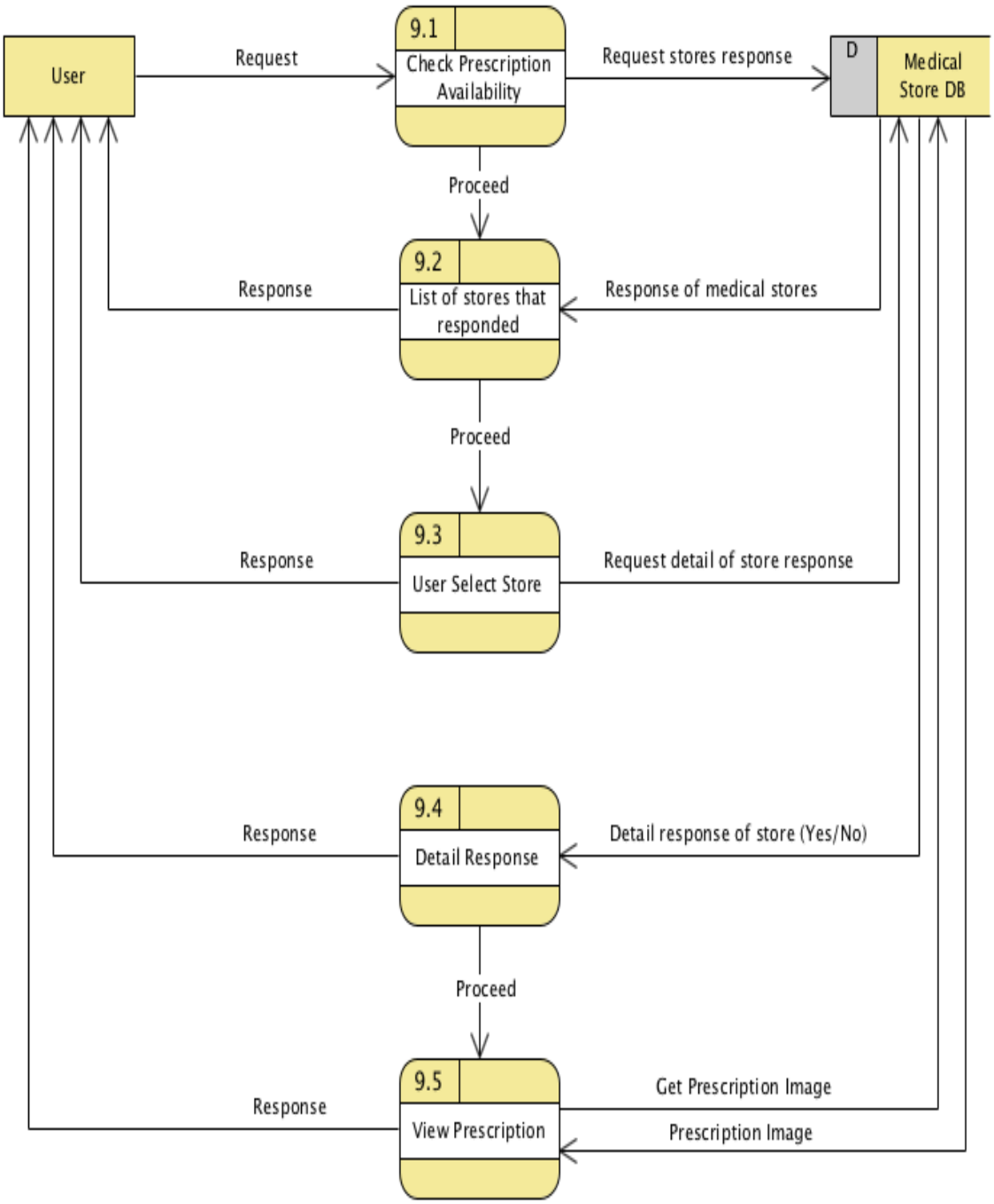

Figure 4.11: DFD (Level-2) – Check Prescription Availability - This figure describes the check prescription availability process in the system, that how data flow when user want to check the prescription availability at their desired medical store

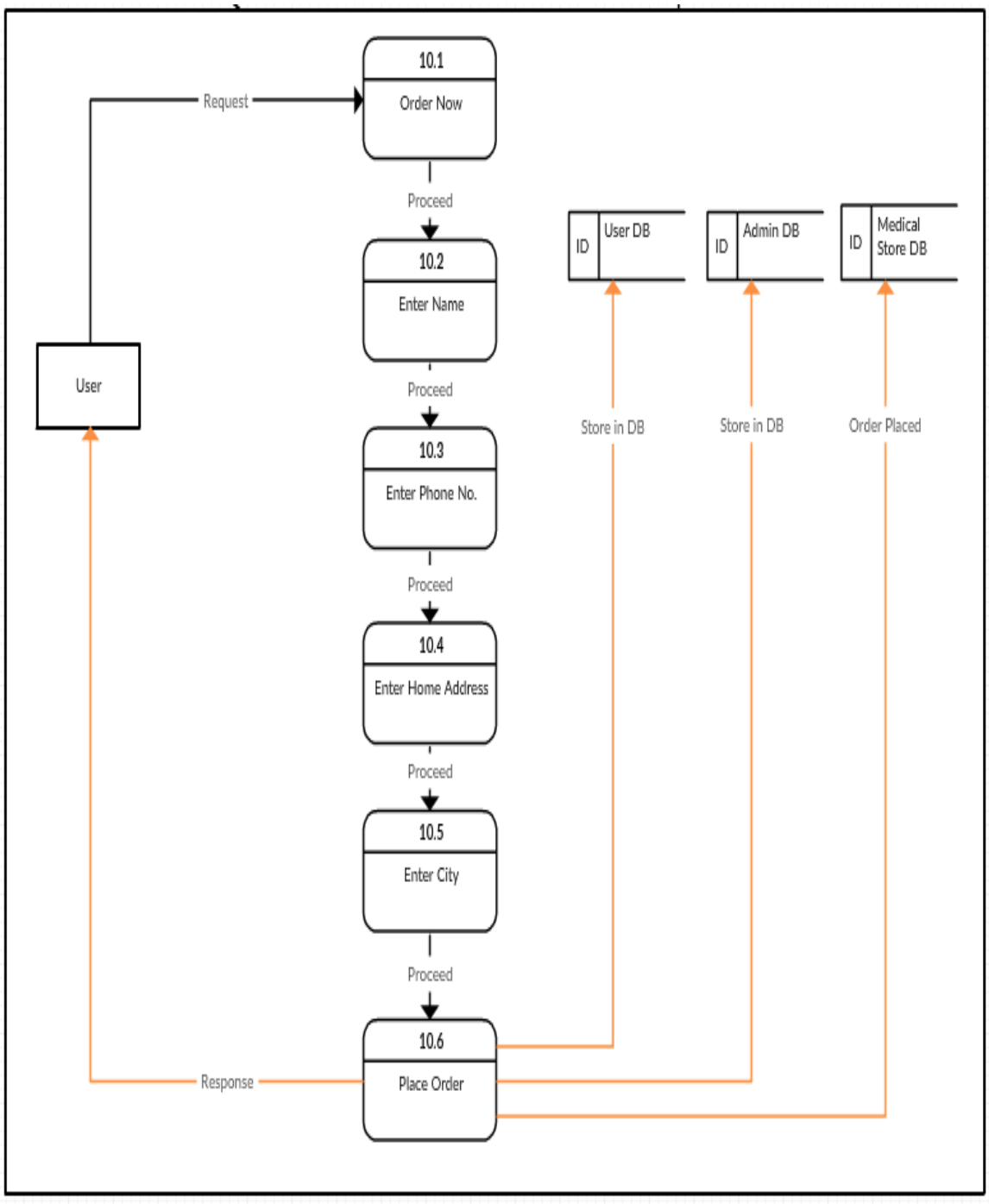

Figure 4.12: DFD (Level-2) – Place Order - This figure describes the place order process in the system, that how data flow when user place an order to their desired medical store

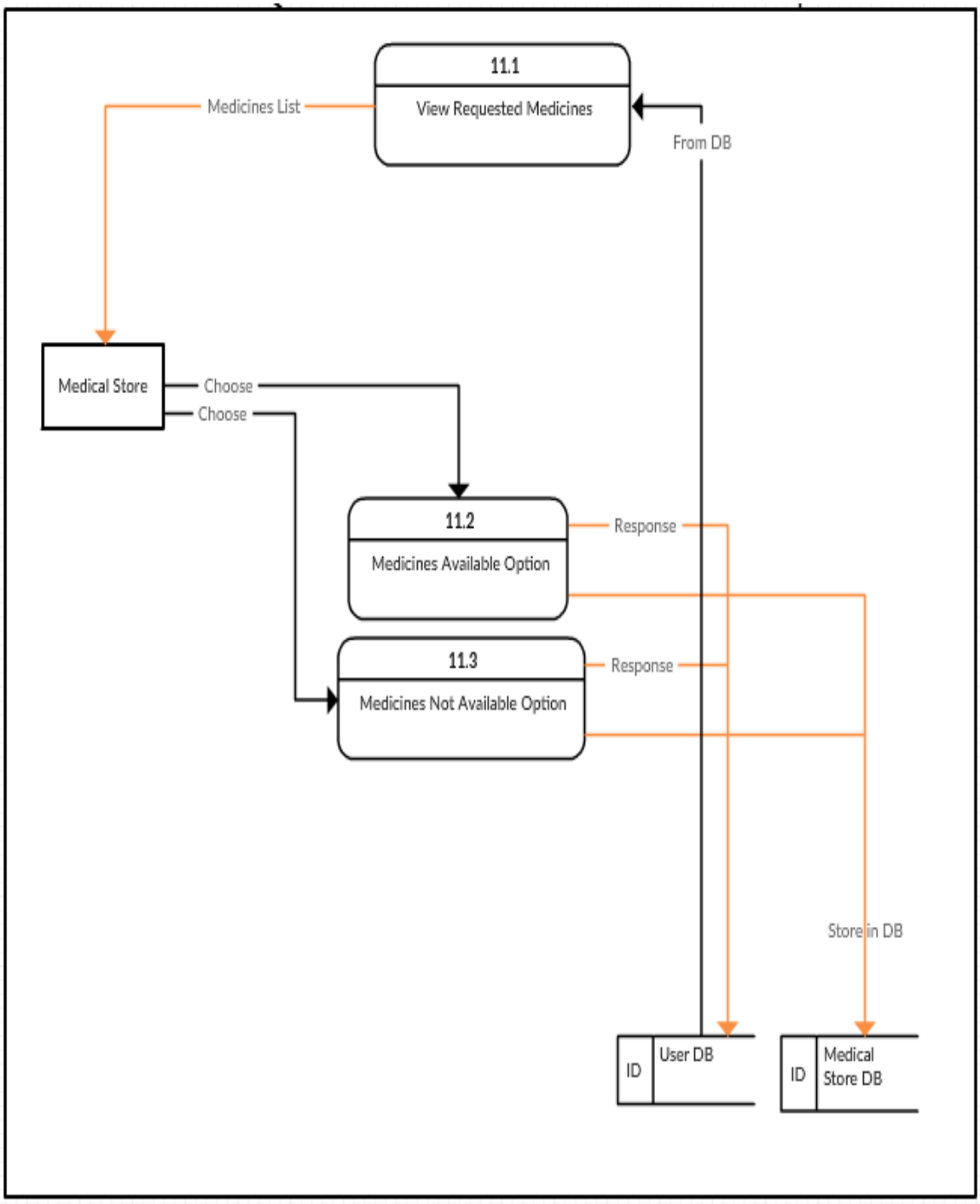

Figure 4.13: DFD (Level-2) – Confirm Medicines Request - This figure describes the confirm medicine request process in the system, that how data flow when medical store respond to medicine requests

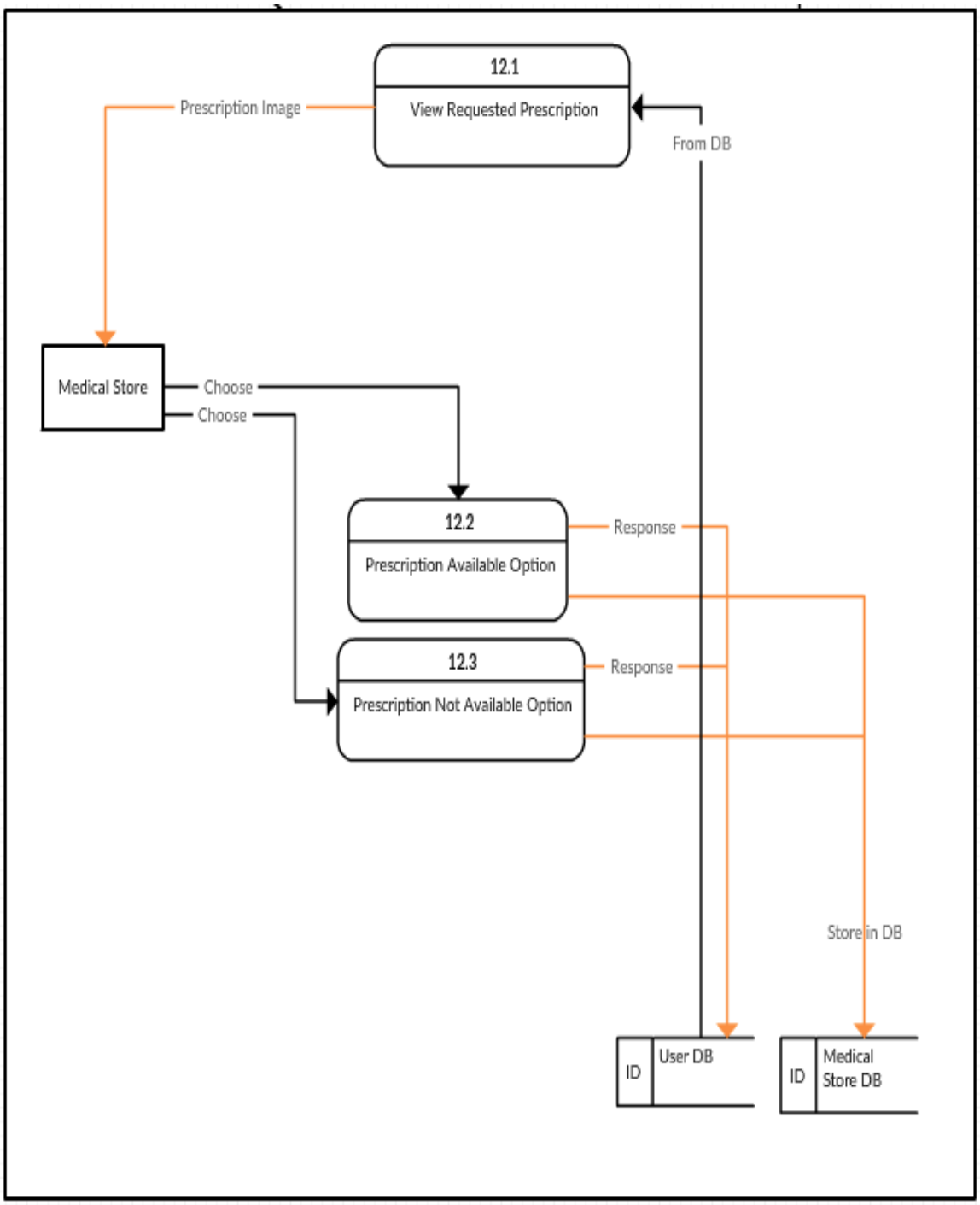

Figure 4.14: DFD (Level-2) – Confirm Prescription Request - This figure describes the confirm prescription request process in the system, that how data flow when medical store respond to prescription requests

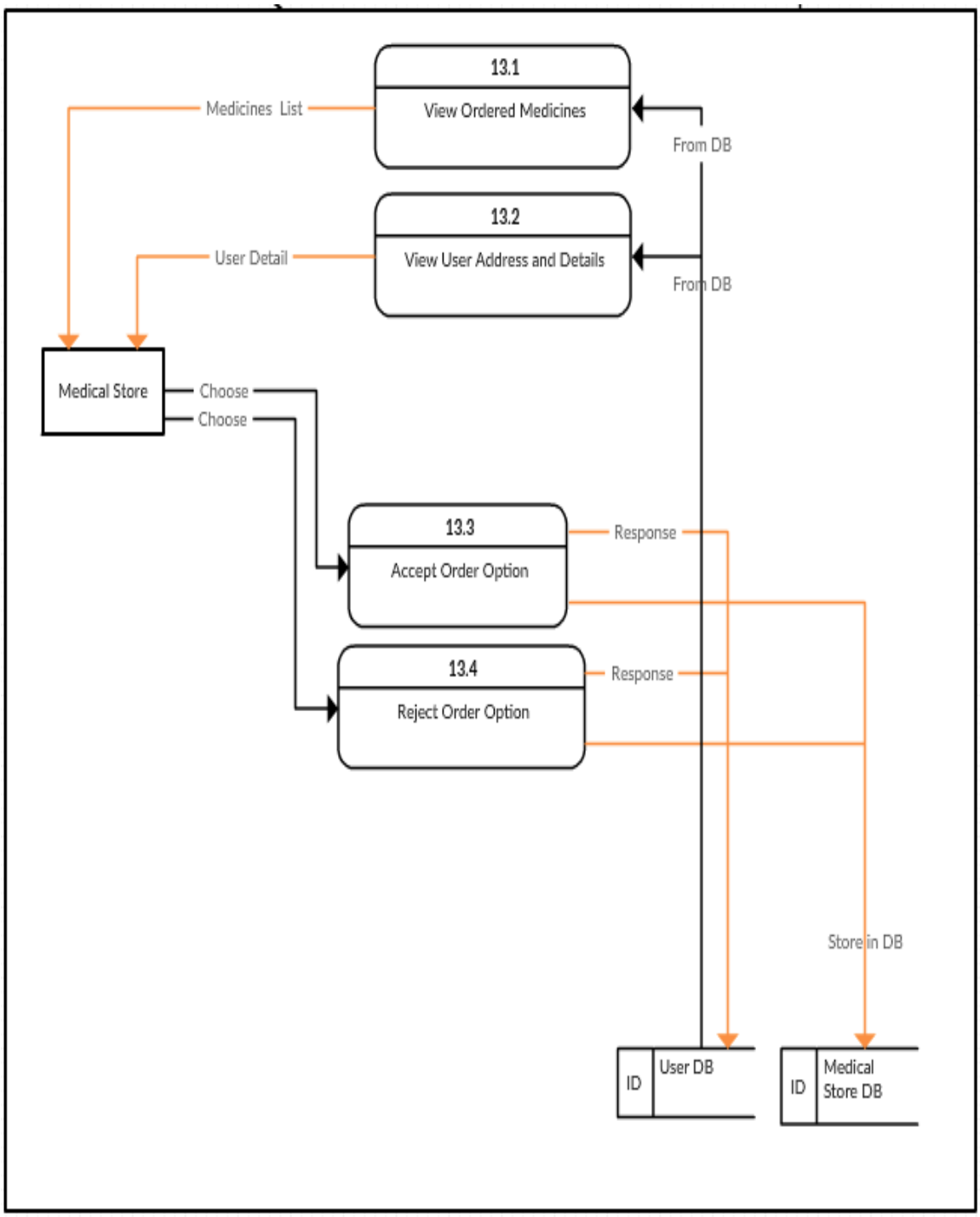

Figure 4.15: DFD (Level-2) – Confirm Medicine Order - This figure describes the confirm medicine order process in the system, that how data flow when medical store respond to medicine order

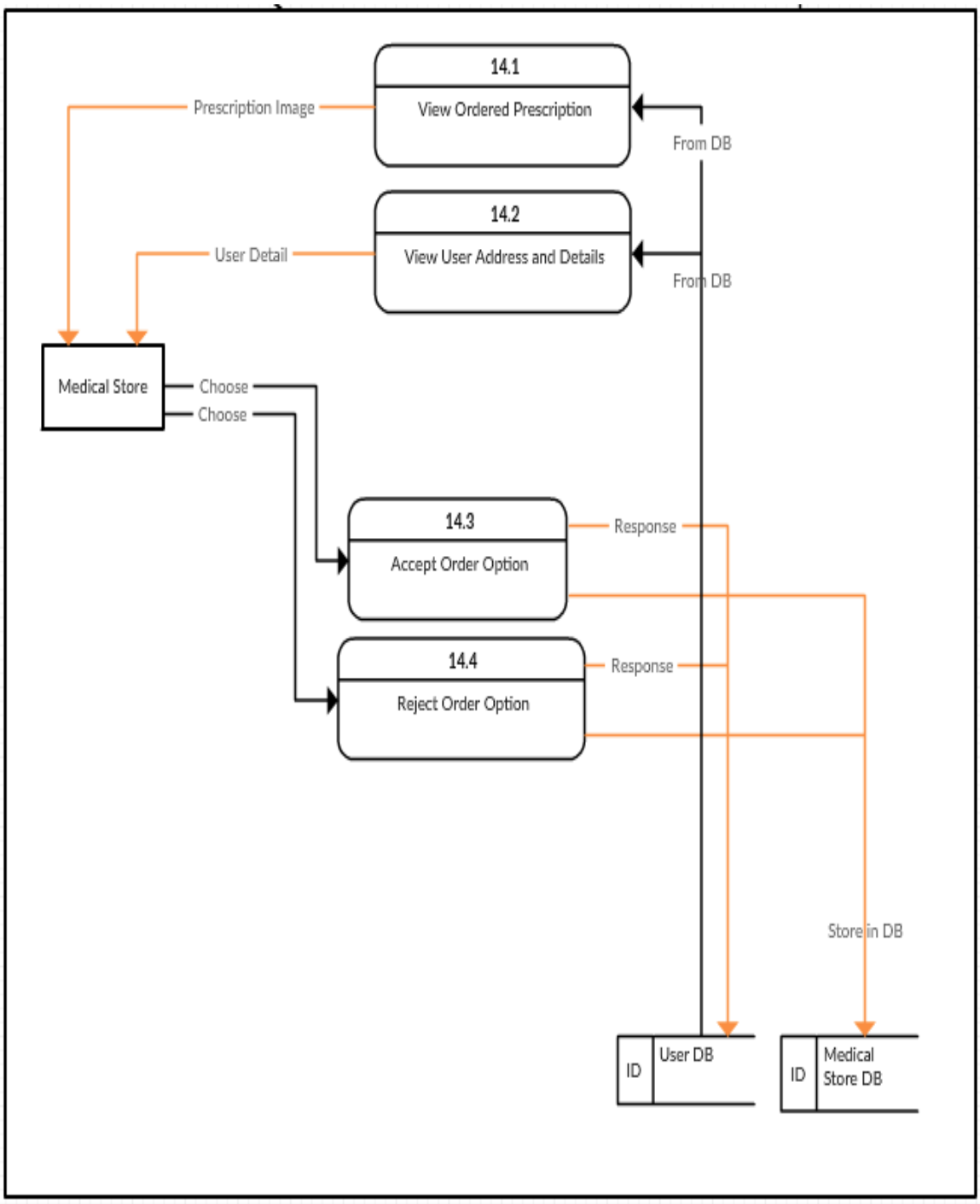

Figure 4.16: DFD (Level-2) – Confirm Prescription Order - This figure describes the confirm prescription order process in the system, that how data flow when medical store respond to prescription order

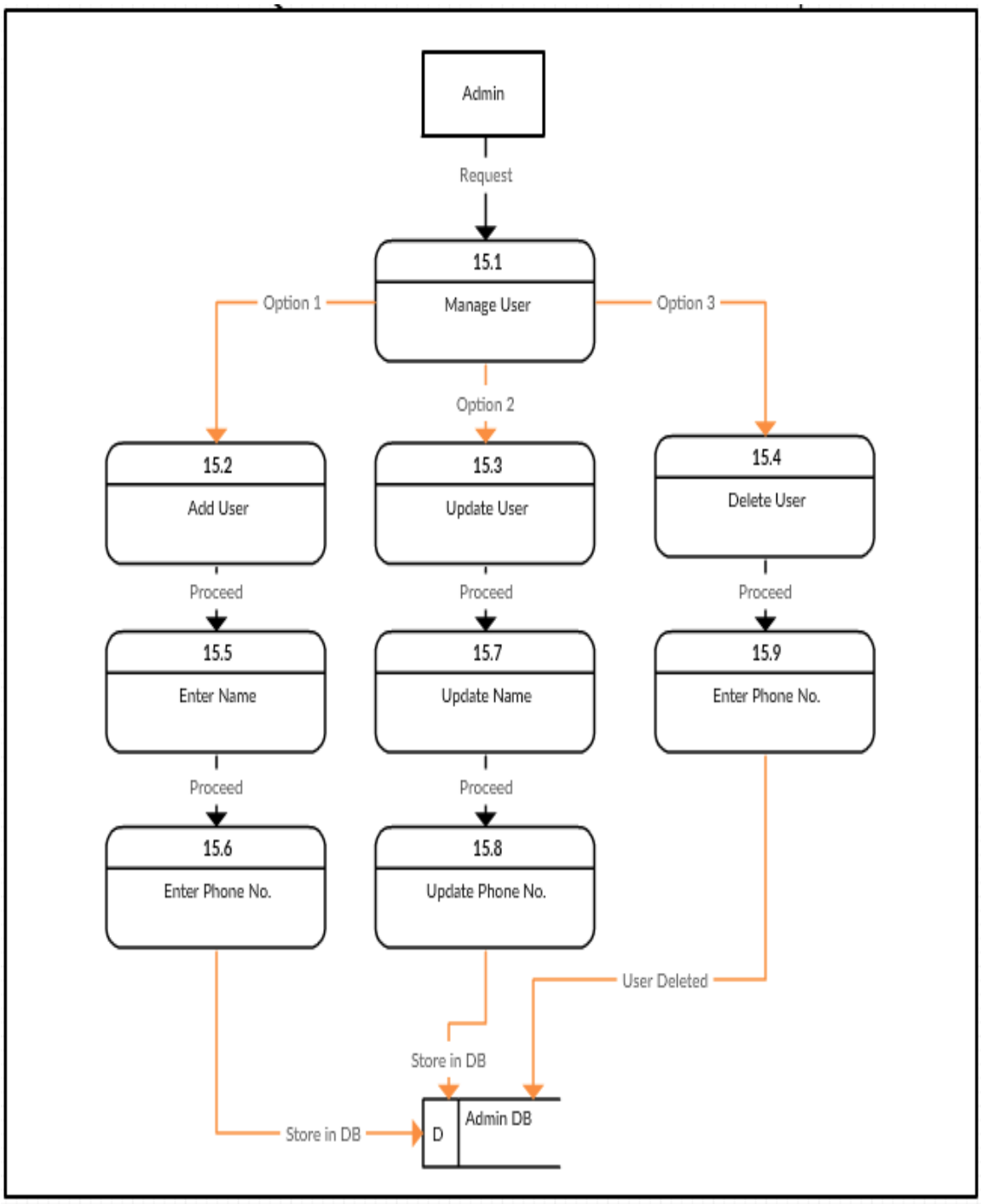

Figure 4.17: DFD (Level-2) – Manage Users - This figure describes the manage user process in the system, that how data flow when admin manage the users in the system

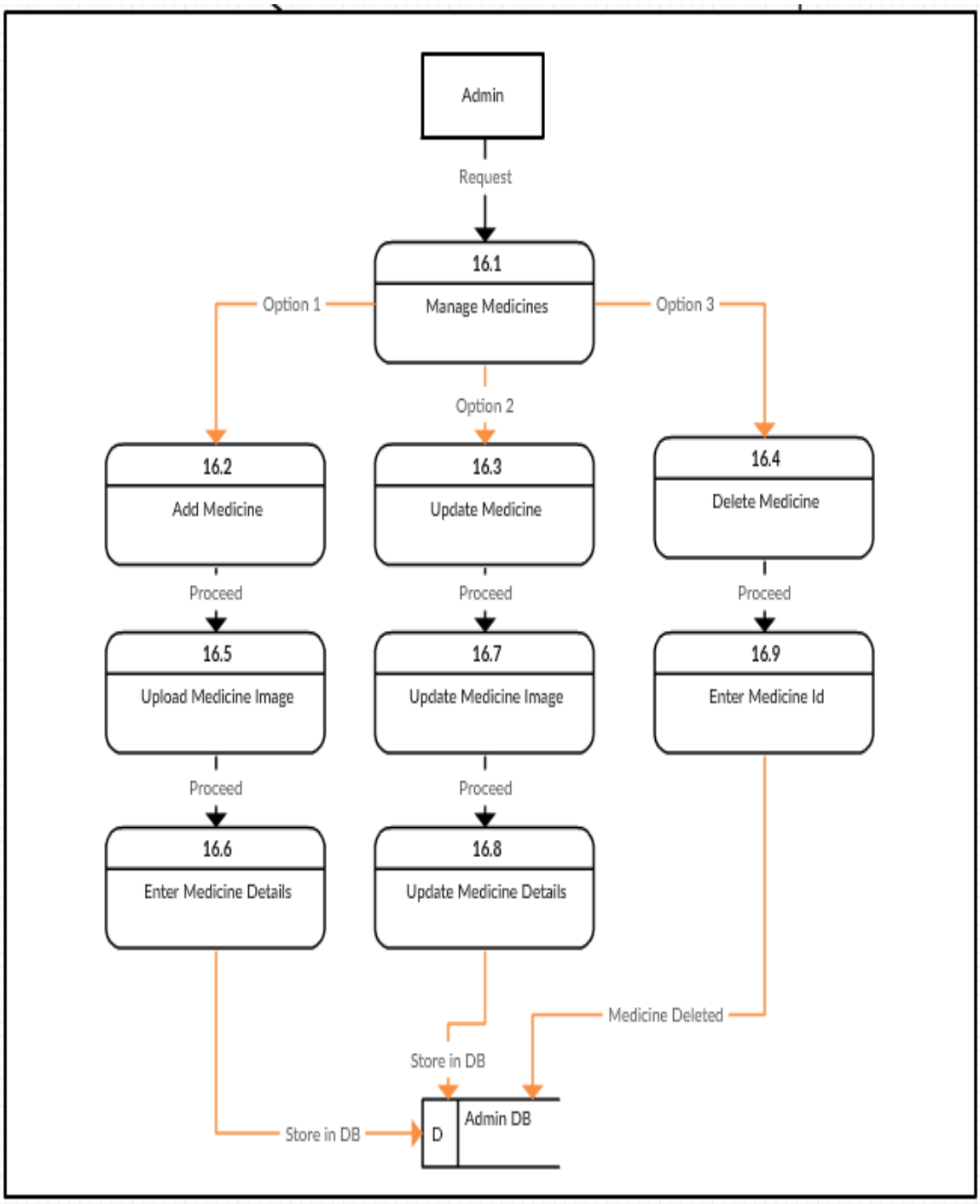

Figure 4.18: DFD (Level-2) – Manage Medicines - This figure describes the manage medicines process in the system, that how data flow when admin manage the medicines in the system

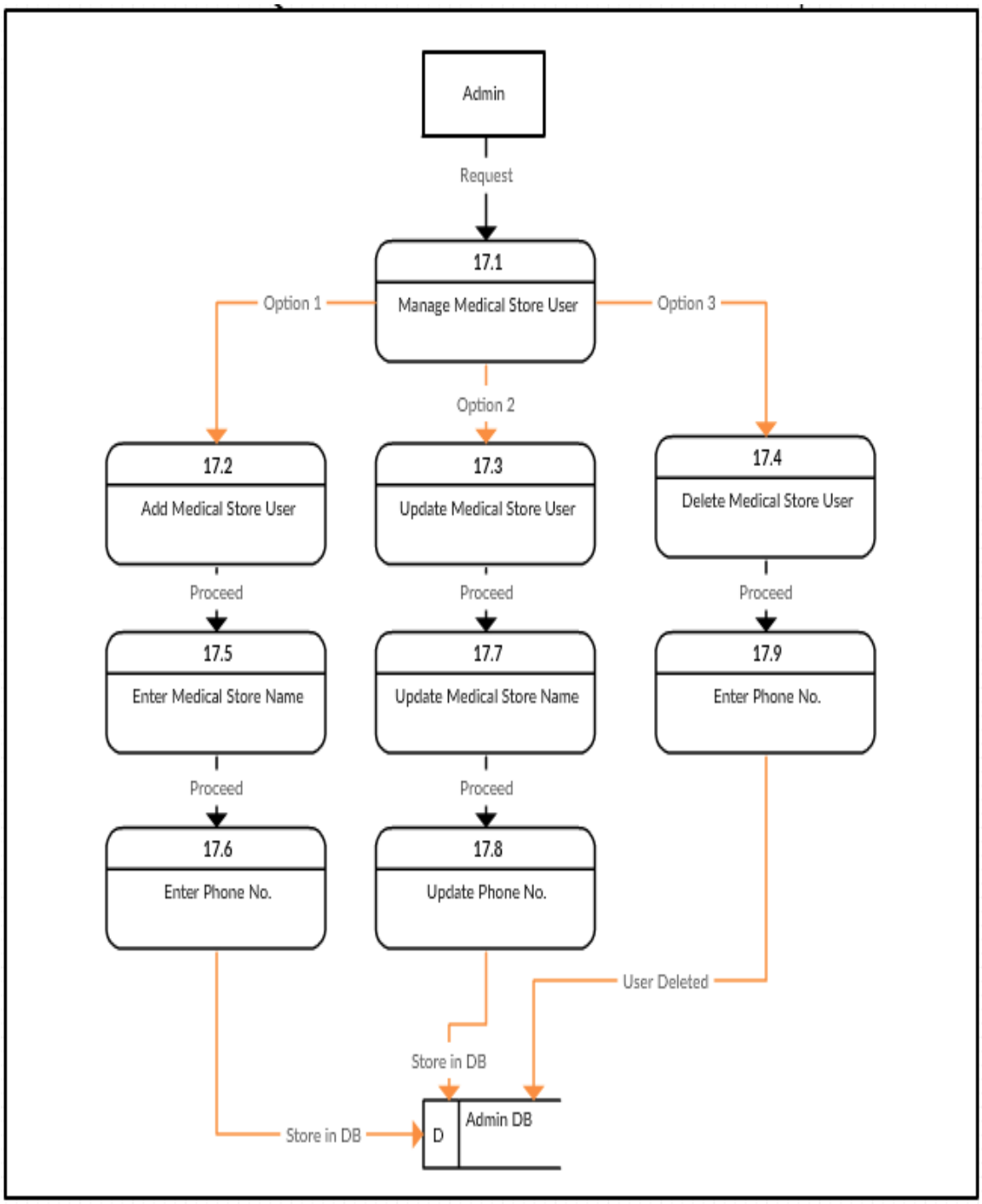

Figure 4.19: DFD (Level-2) – Manage Medical Store Users - This figure describes the manage medical store users process in the system, that how data flow when admin manage the medical store users in the system

CHAPTER # 5

5. SYSTEM DESIGN

5.1 System Architecture Diagram

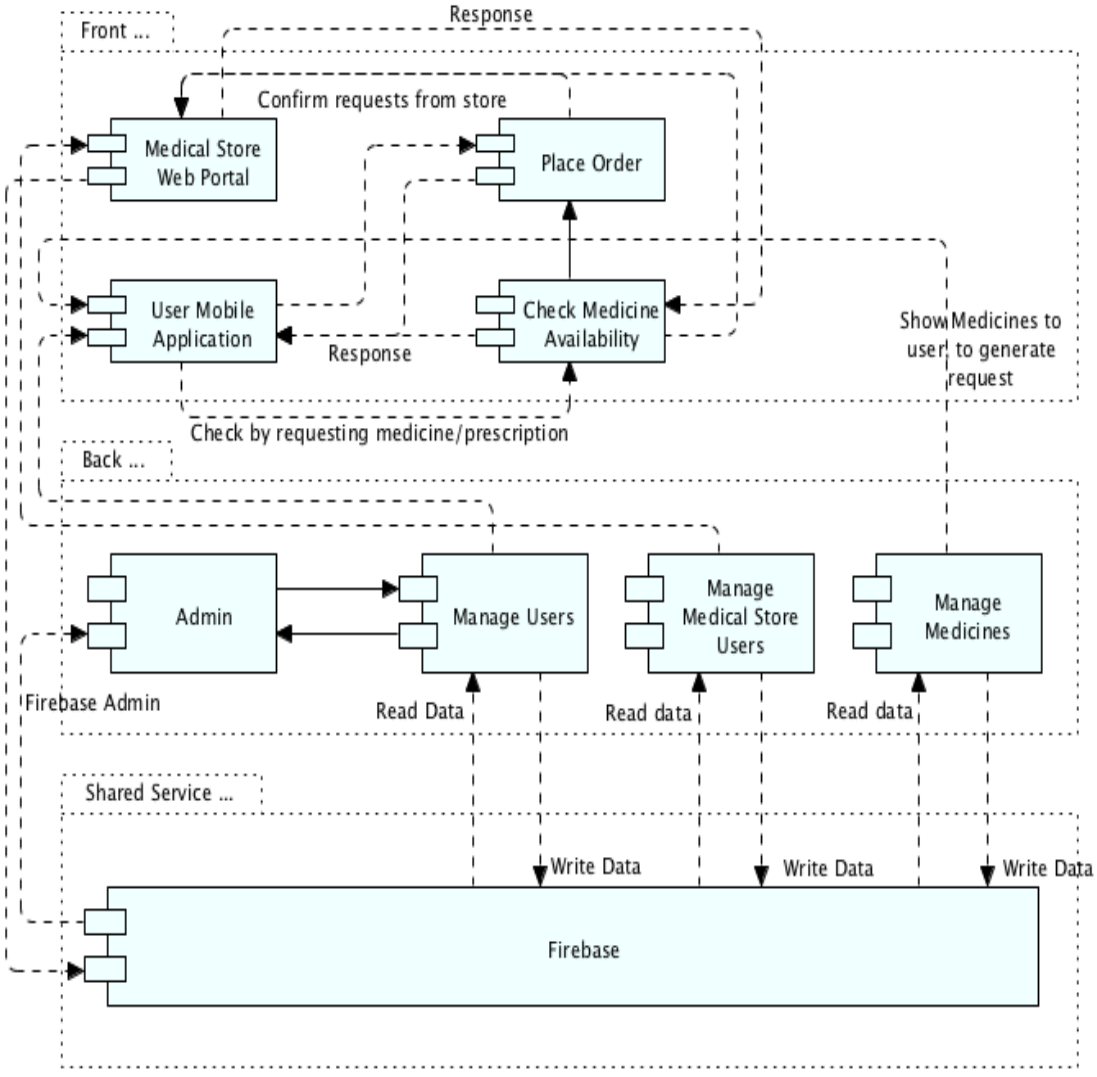

Figure 5.1: System Architecture Diagram - This figure describes the front-end, back-end & shared services in system. There are two users, which are interacting with this system. User connects with the system via mobile app and the medical store user connects with web portal. Mobile user generates a request availability of medicine, this request response by medical store user. After getting response user place an order. Admin manages user in firebase. In this section there are two types of user one is customer user and other one is medical store user.

5.2 Class Diagram

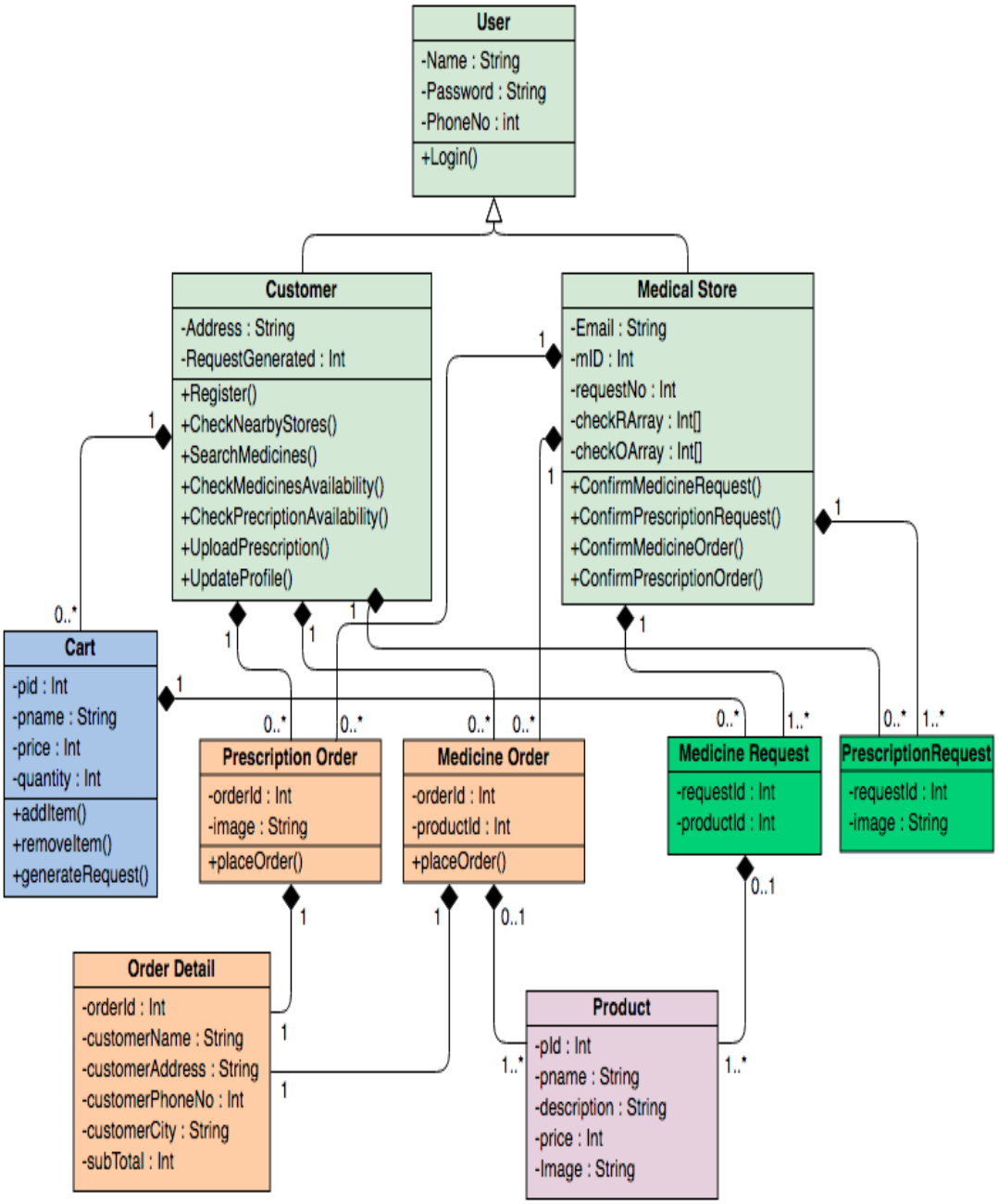

Figure 5.2: Class Diagram - This figure describes the classes in our system. The customer user has one to many relations with cart object. Cart has also one to many relations with the medicine request whereas prescription request object has many to one relation with the customer object. The medical store user has one to many relations with prescription request, prescription order, and medicine request and medicine order. Prescription order object has one to many relations with the customer user and medical store user whereas the other relation is one to one relation with the order details object.

5.3 Sequence Diagrams

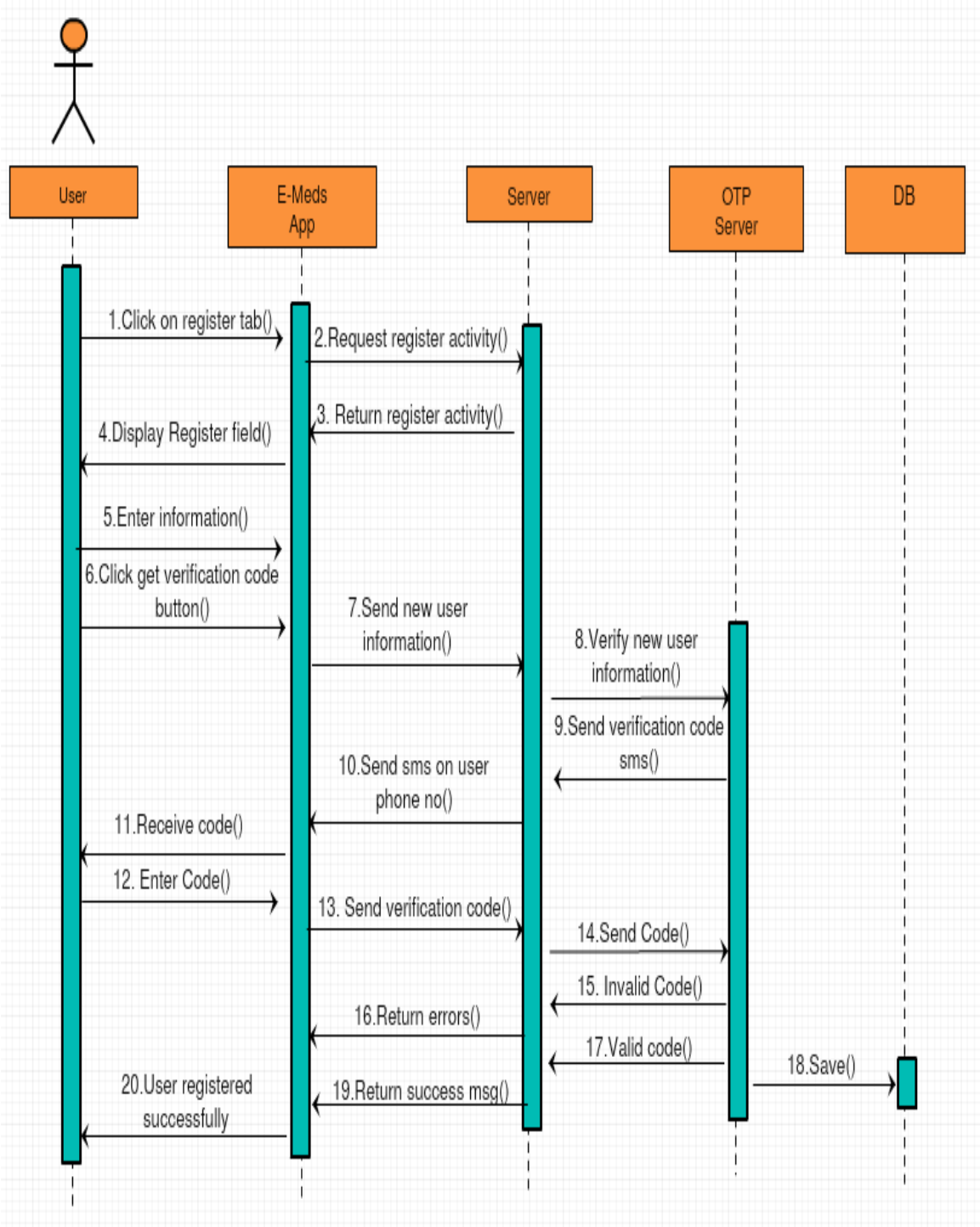

Figure 5.3: Sequence Diagram - Register - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this register sequence user, E-meds app, server, opt server and database are involved.

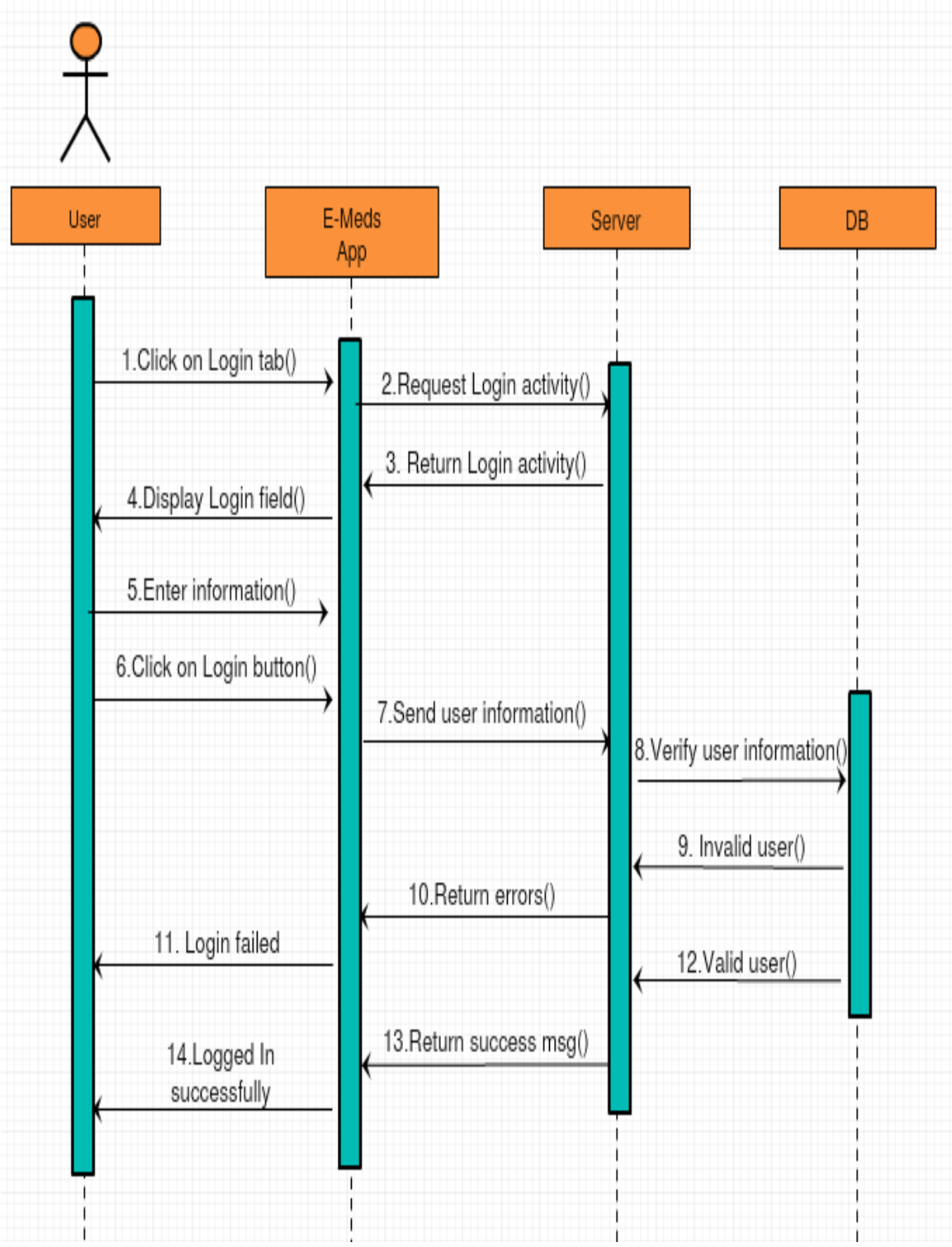

Figure 5.4: Sequence Diagram - Login - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this login sequence user, E-meds app, server and database are involved.

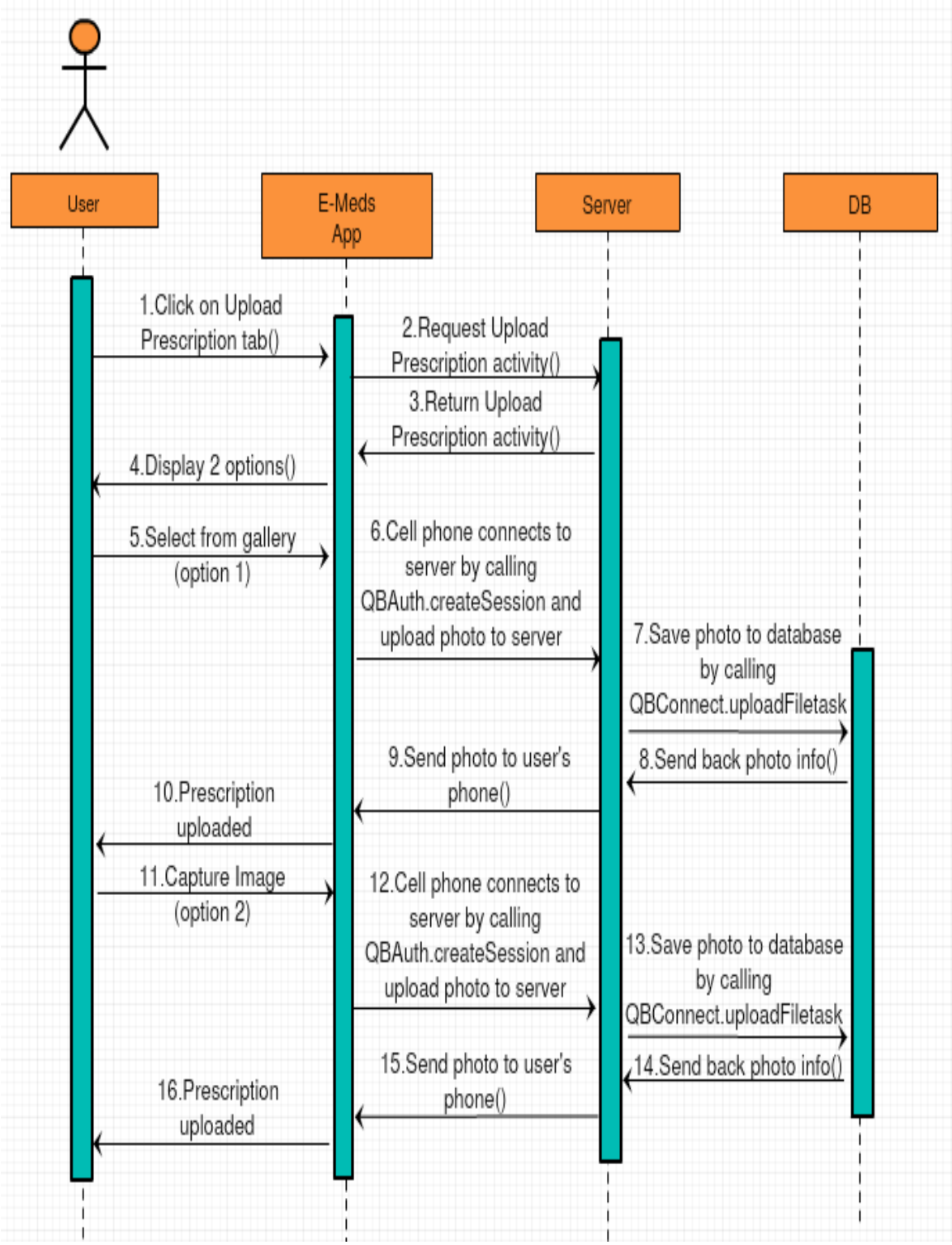

Figure 5.5: Sequence Diagram – Upload Prescription - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this uploading sequence user, E-meds app, server and database are involved.

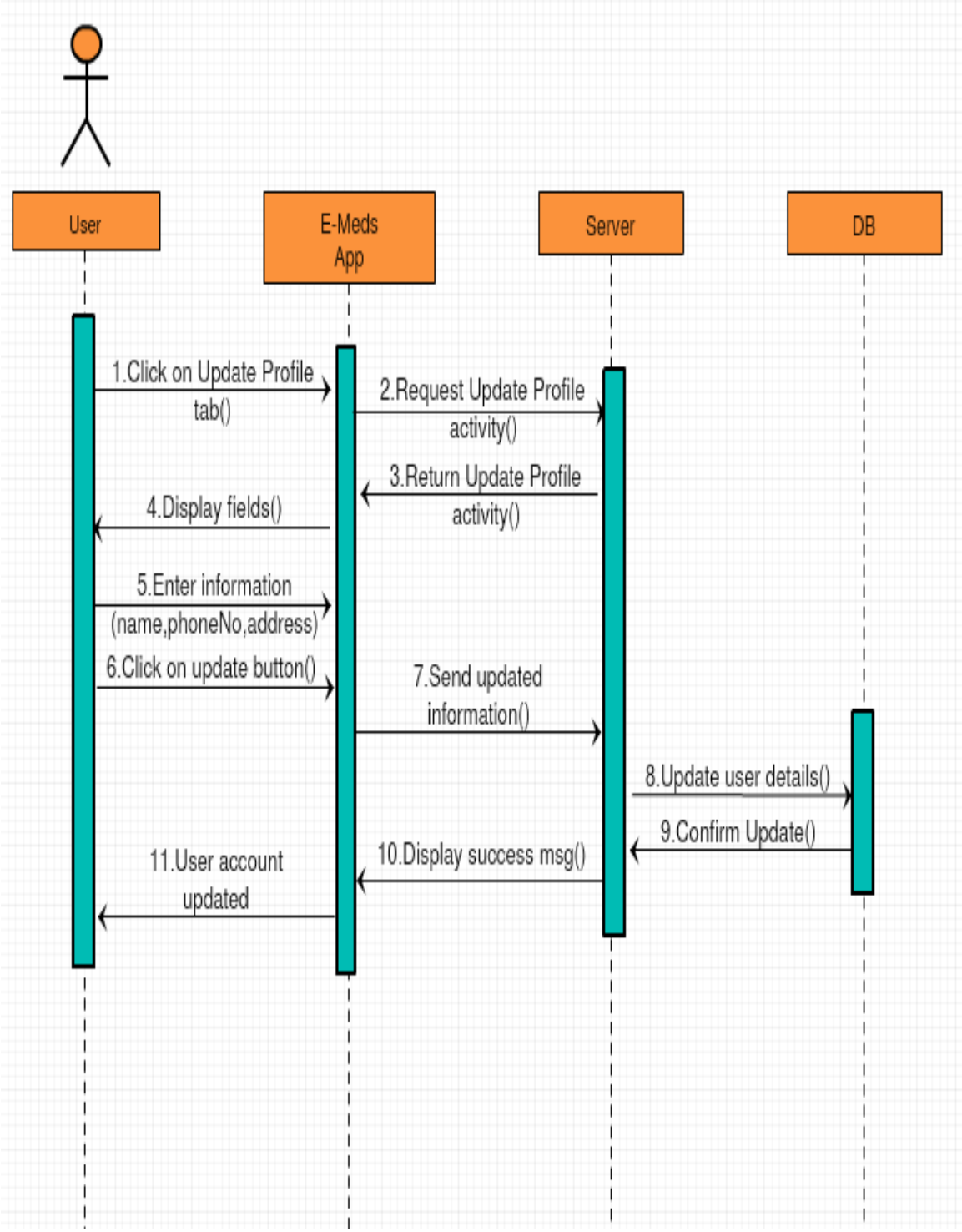

Figure 5.6: Sequence Diagram – Update Profile - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this update profile sequence user, E-meds app, server and database are involved.

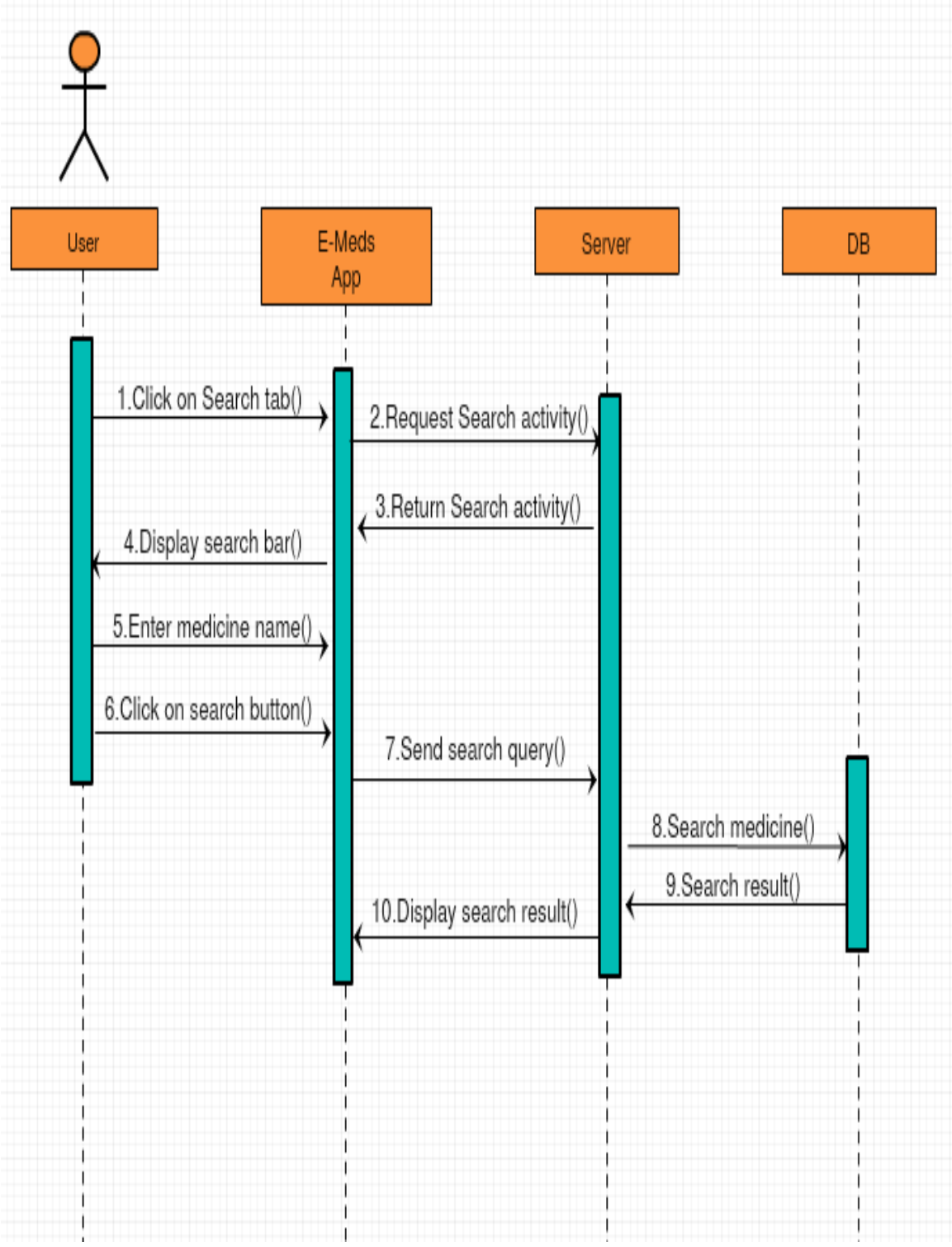

Figure 5.7: Sequence Diagram – Search Medicines - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this search medicines sequence user, E-meds app, server and database are involved.

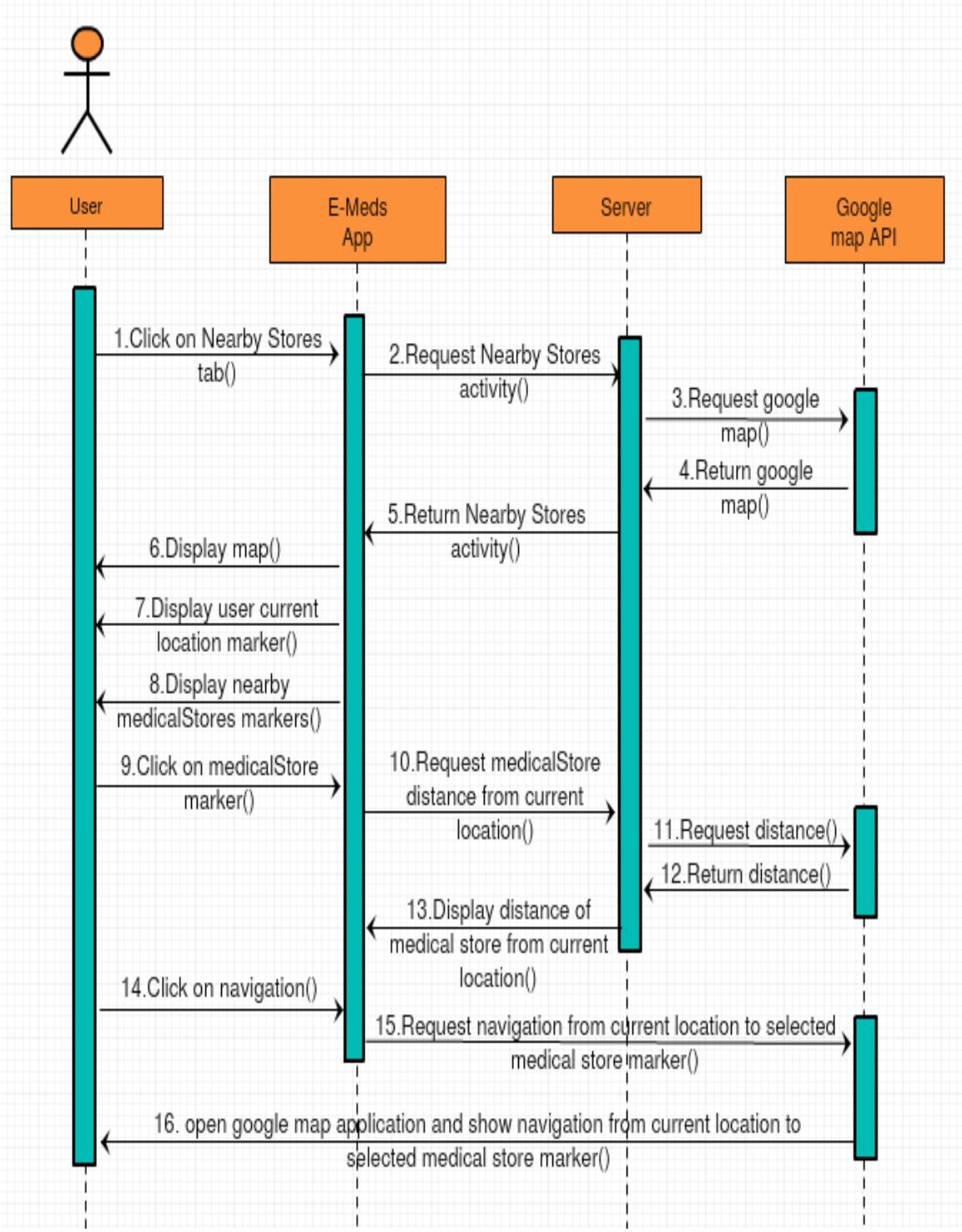

Figure 5.8: Sequence Diagram – Check Nearby Stores - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this check nearby store sequence user, E-meds app, server and Google Map API are involved.

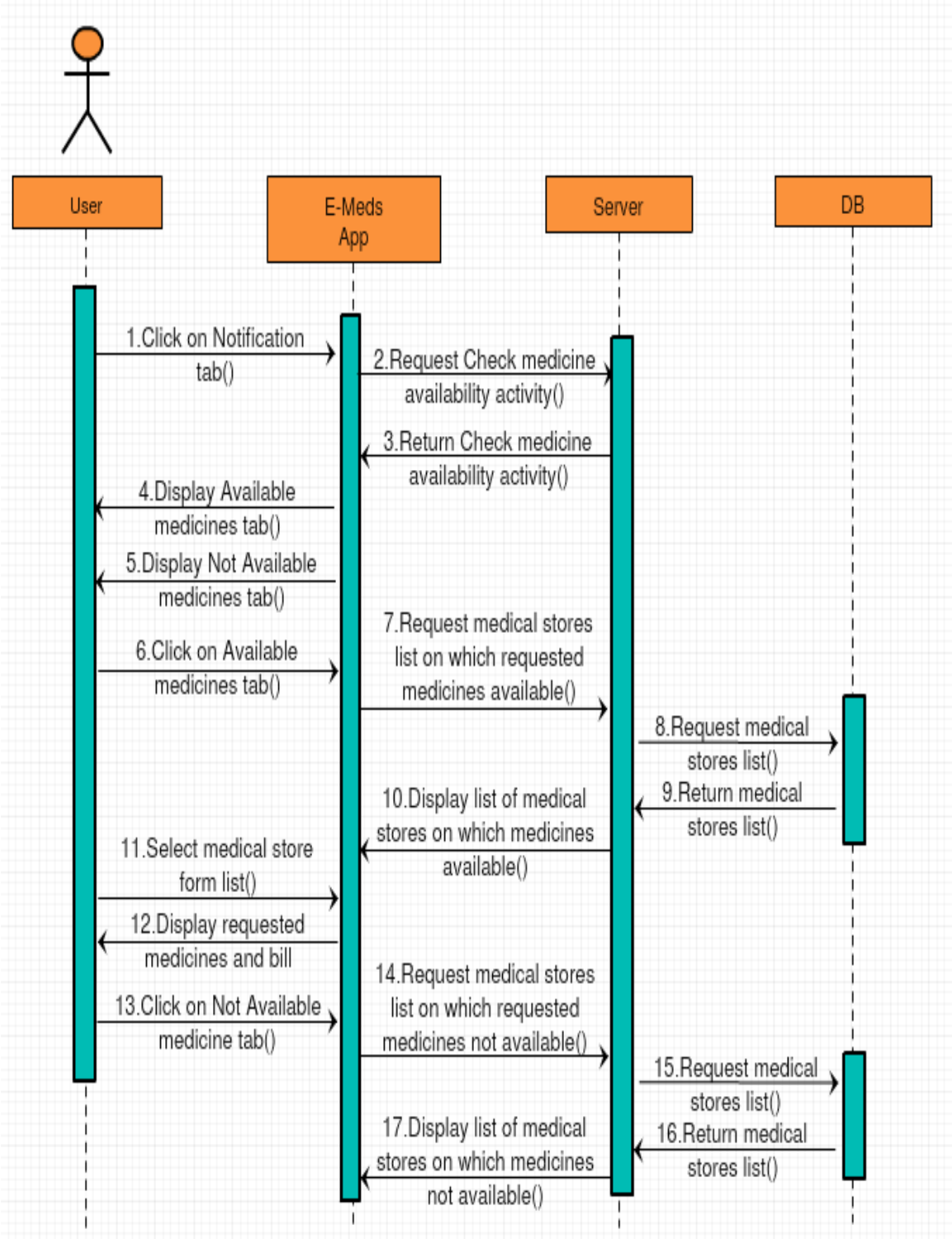

Figure 5.9: Sequence Diagram – Check Medicines Availability - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this check medicines availability sequence user, E-meds app, server and database are involved.

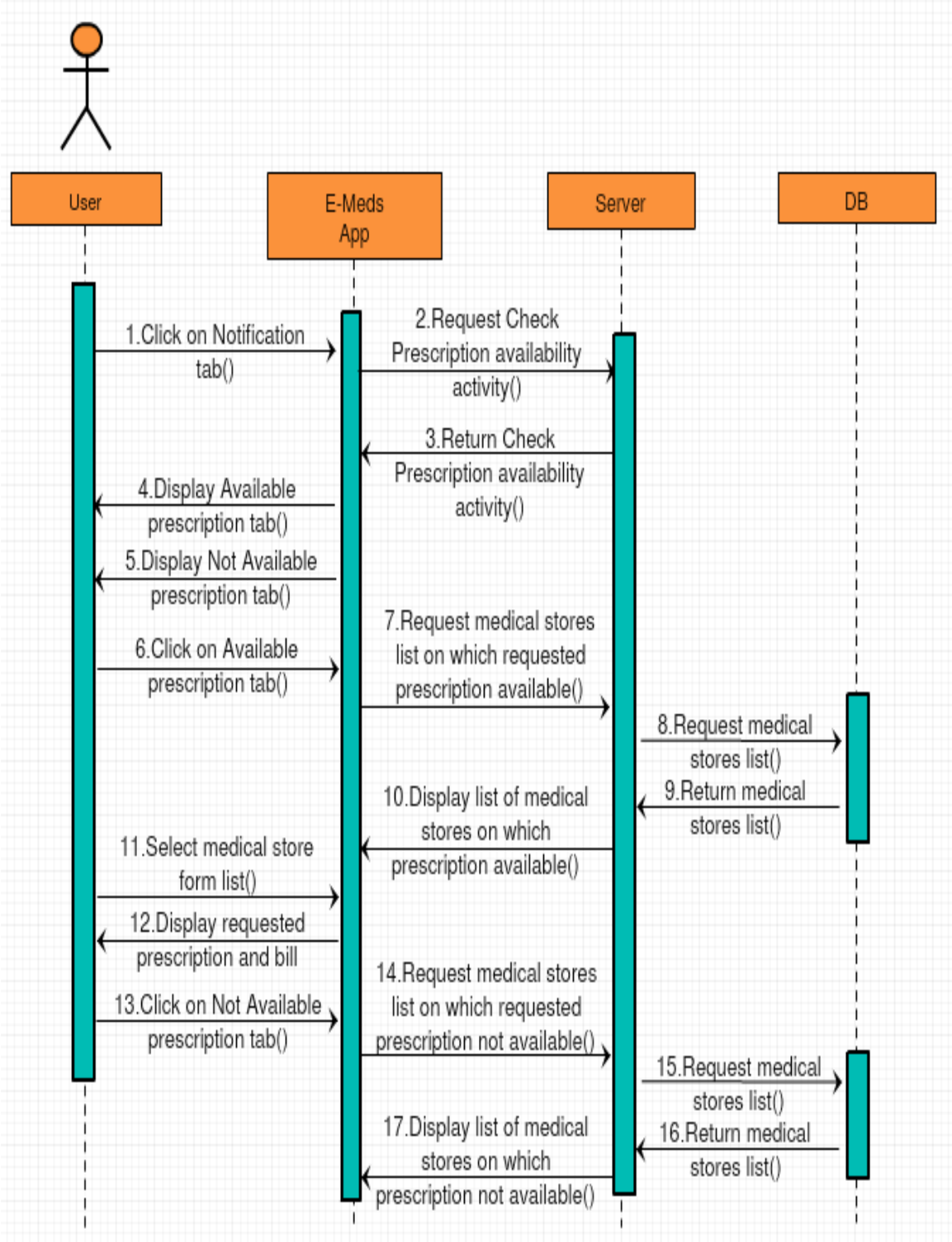

Figure 5.10: Sequence Diagram – Check Prescription Availability - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this check prescription availability sequence user, E-meds app, server and database are involved.

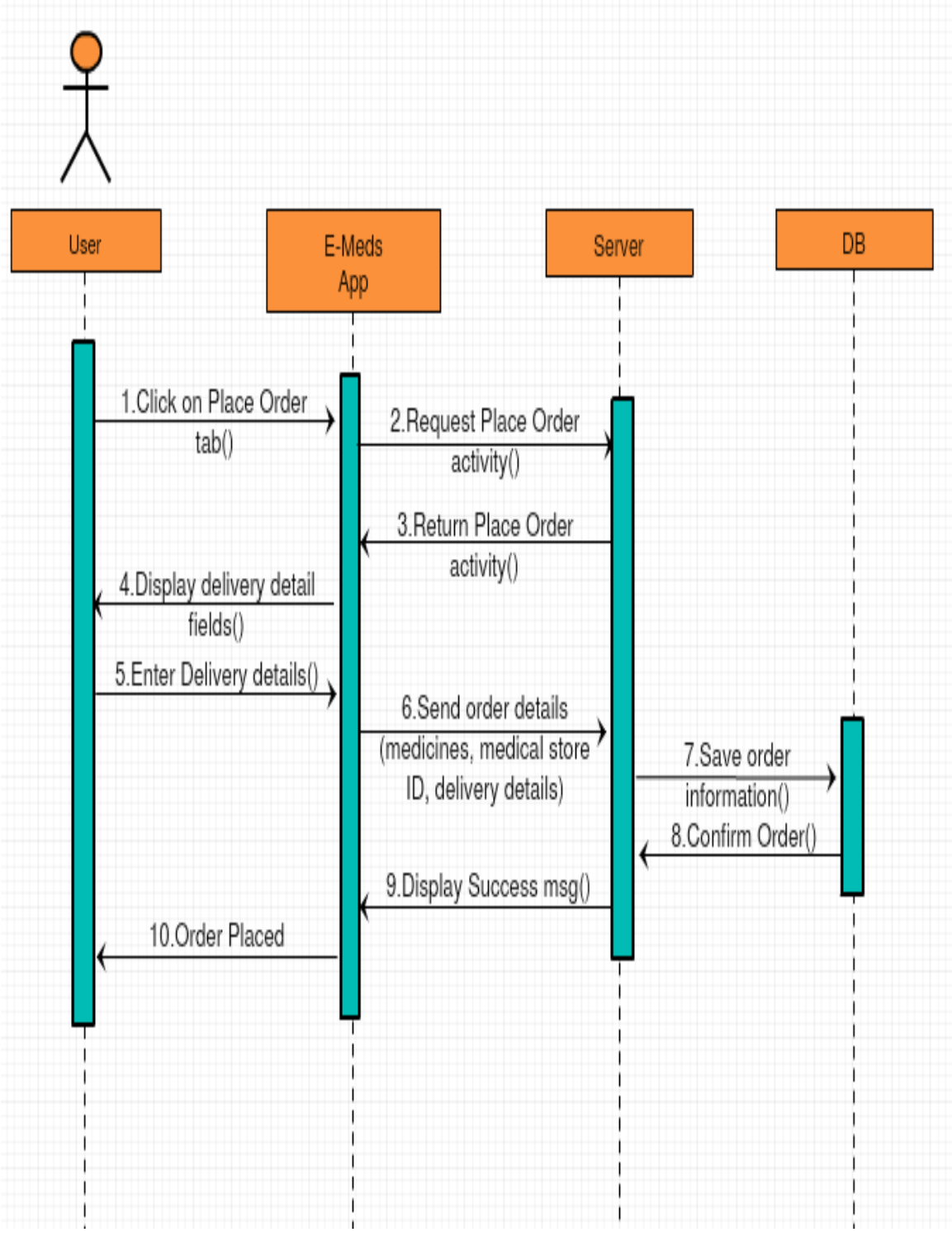

Figure 5.11: Sequence Diagram – Place Order - This figure describes the object interact with the functionality with the sequence of the scenario. User object functions performing a task and the other object play a part to fulfill the tasks. In this place order sequence user, E-meds app, server and database are involved.

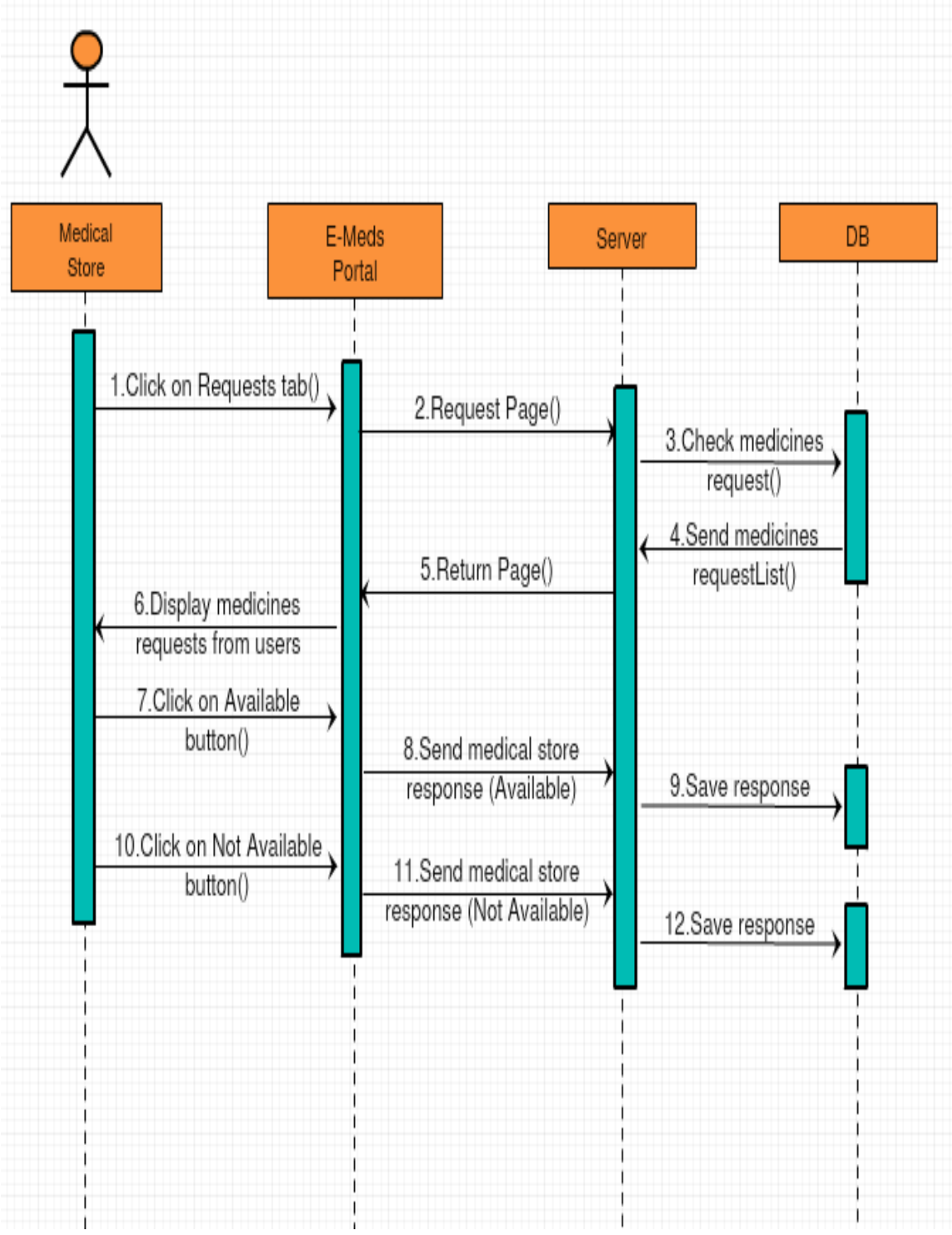

Figure 5.12: Sequence Diagram – Confirm Medicines Request - This figure describes the object interact with the functionality with the sequence of the scenario. Medical store user object functions performing a task and the other object play a part to fulfill the tasks. In this confirm medicines request sequence medical store user, E-meds portal, server and database are involved.

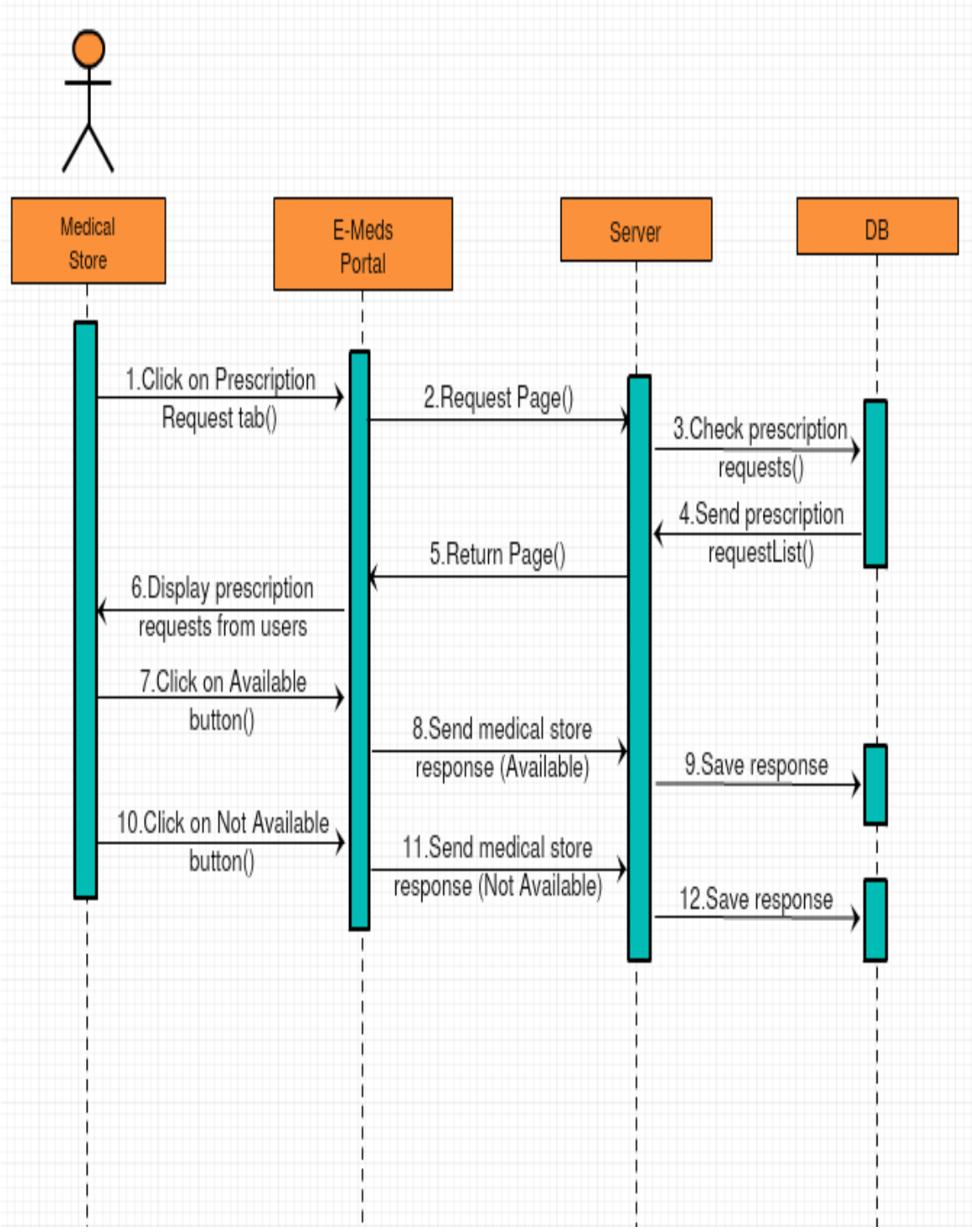

Figure 5.13: Sequence Diagram – Confirm Prescription Request - This figure describes the object interact with the functionality with the sequence of the scenario. Medical store user object functions performing a task and the other object play a part to fulfill the tasks. In this confirm prescription request sequence medical store user, E-meds portal, server and database are involved.

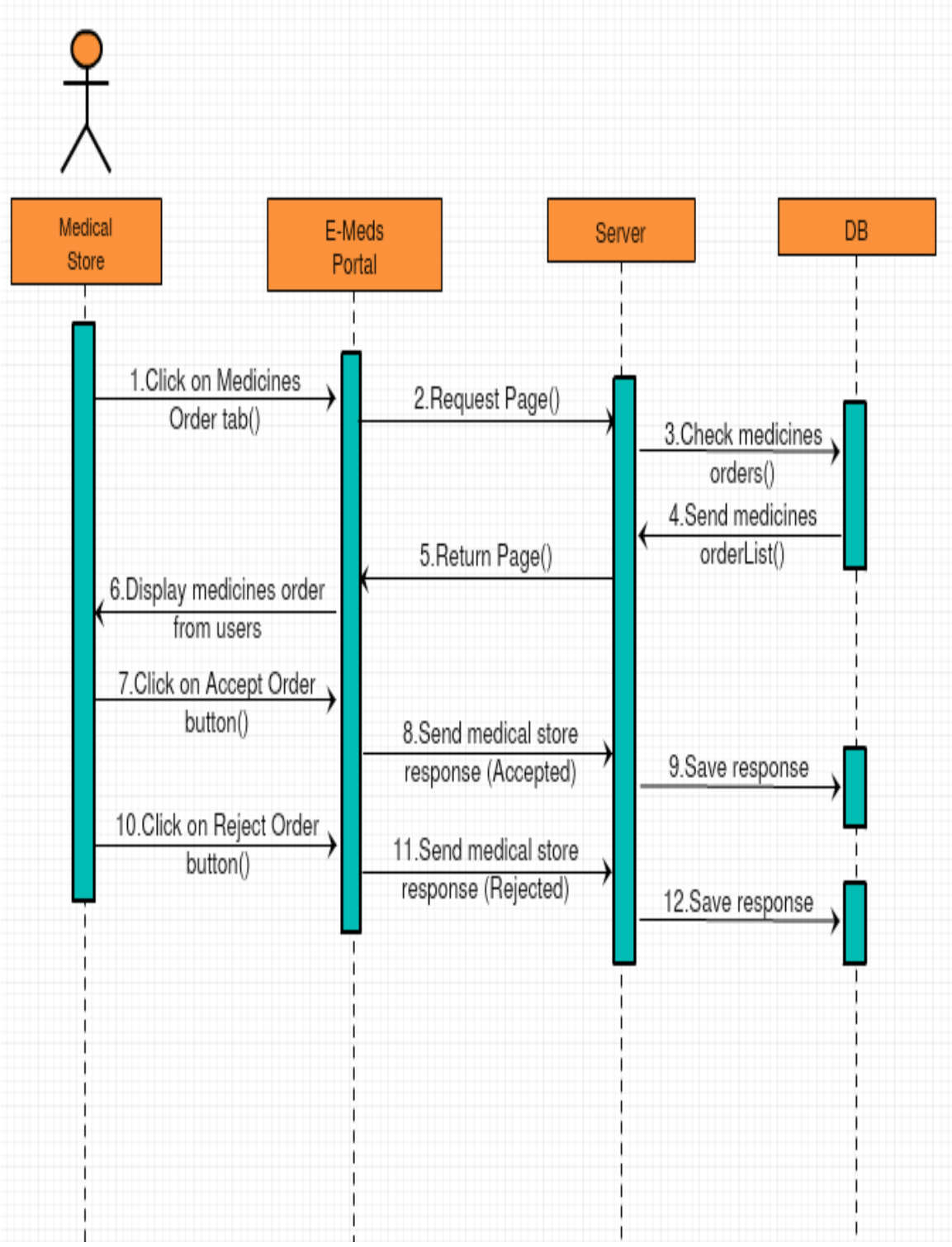

Figure 5.14: Sequence Diagram – Confirm Medicines Order - This figure describes the object interact with the functionality with the sequence of the scenario. Medical store user object functions performing a task and the other object play a part to fulfill the tasks. In this confirm medicines order sequence medical store user, E-meds portal, server and database are involved.

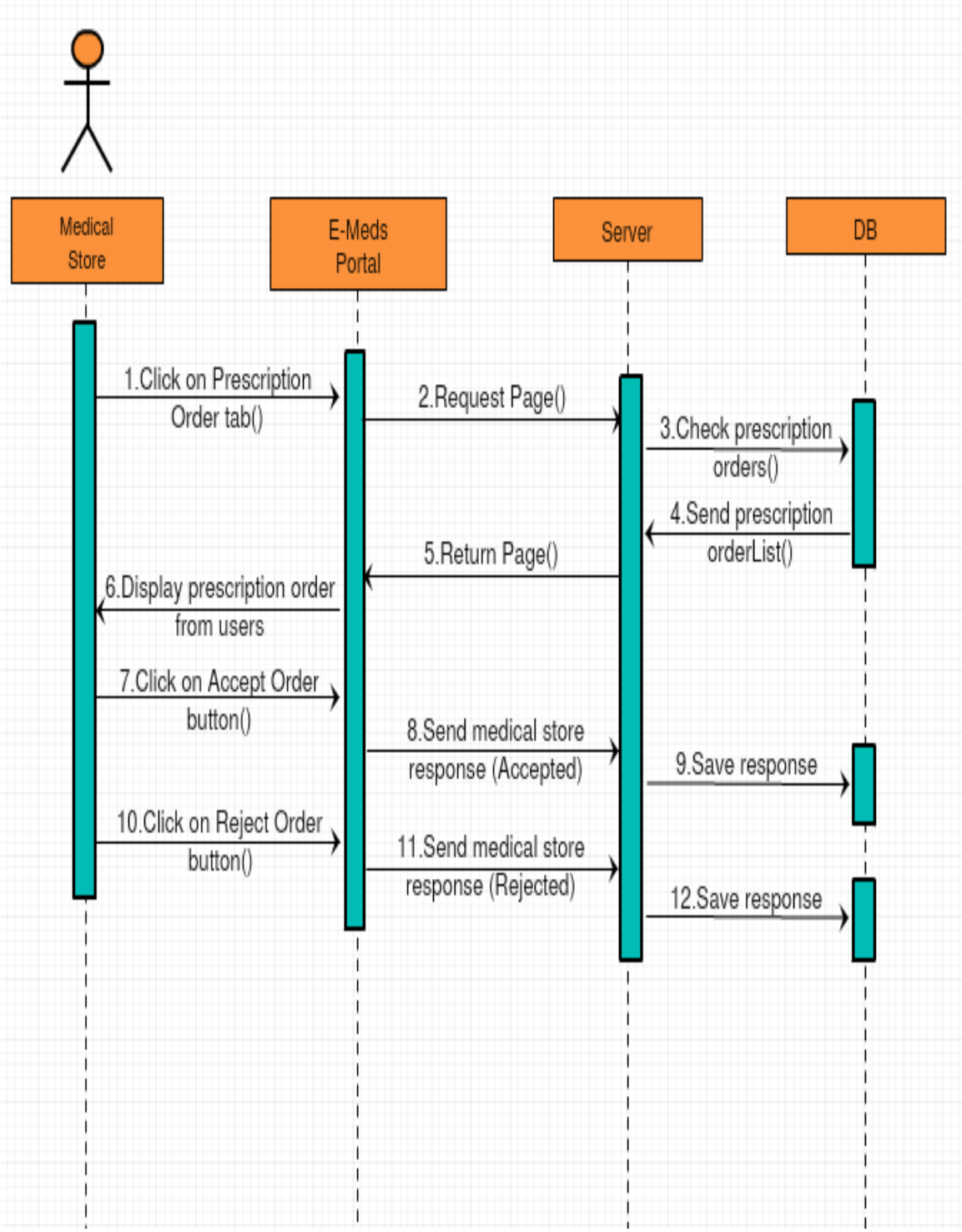

Figure 5.15: Sequence Diagram – Confirm Prescription Order - This figure describes the object interact with the functionality with the sequence of the scenario. Medical store user object functions performing a task and the other object play a part to fulfill the tasks. In this confirm prescription order medical store user, E-meds portal, server and database are involved.

5.4 Collaboration Diagrams

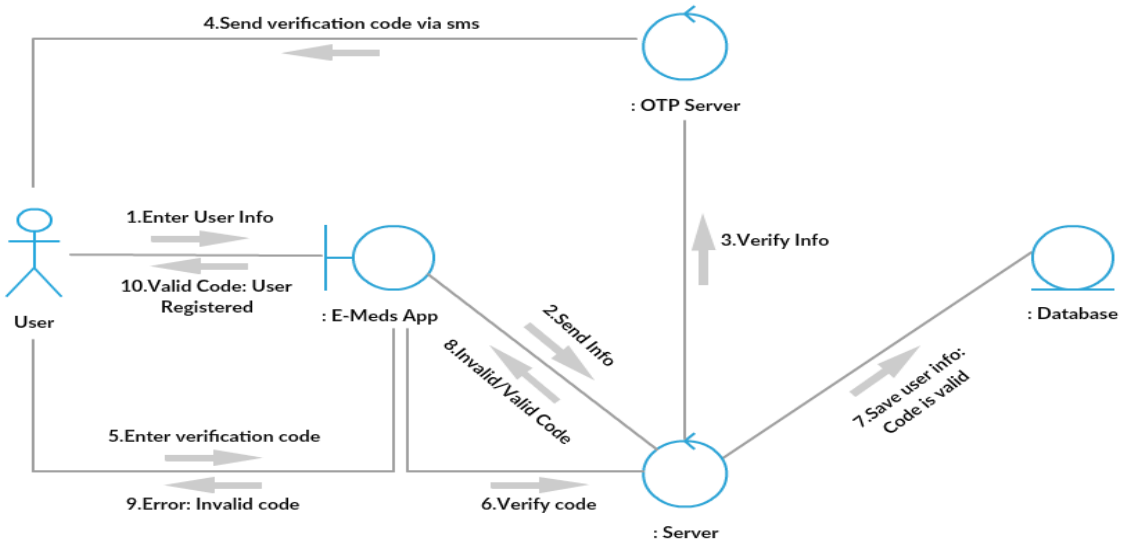

Figure 5.16: Collaboration Diagram – Register - This figure describes the object interact with the functionality with the sequence of the scenario. User object performing a functional task and the other object play a part to fulfill the tasks. In this register process user, E-meds app, server, OTP and database are involved.

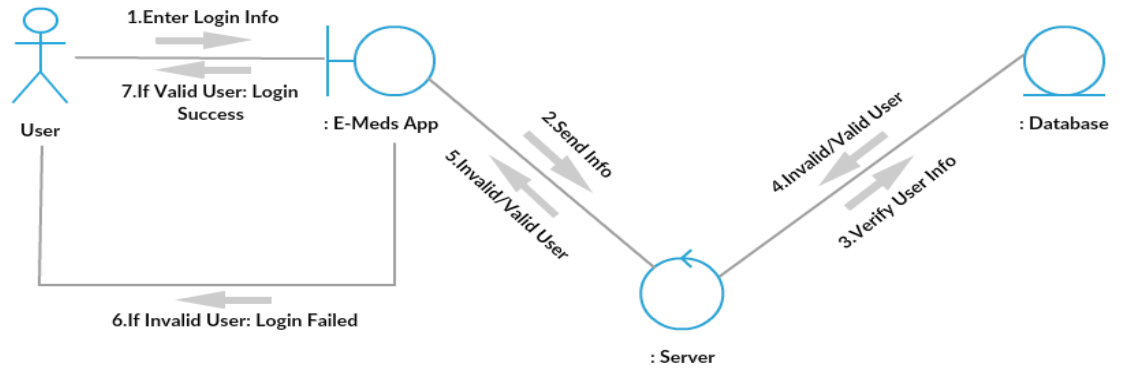

Figure 5.17: Collaboration Diagram – Login - This figure describes the object interact with the functionality with the sequence of the scenario. User object performing a functional task and the other object play a part to fulfill the tasks. In this login process user, E-meds app, server and database are involved.

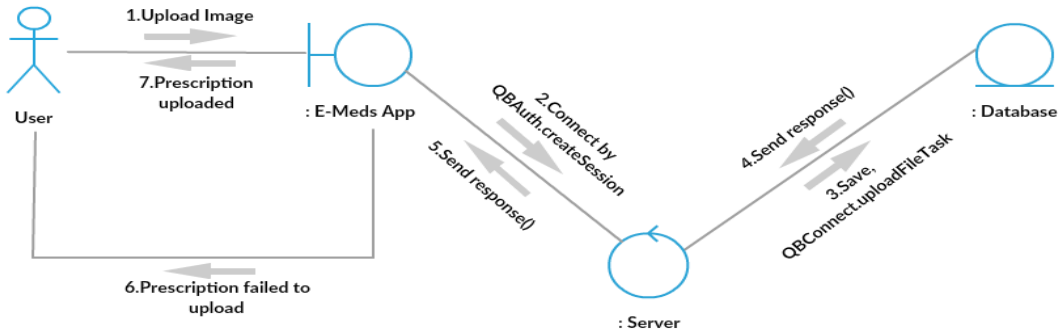

Figure 5.18: Collaboration Diagram – Upload Prescription - This figure describes the object interact with the functionality with the sequence of the scenario. User object performing a functional task and the other object play a part to fulfill the tasks. In this upload prescription process user, E-meds app, server and database are involved.

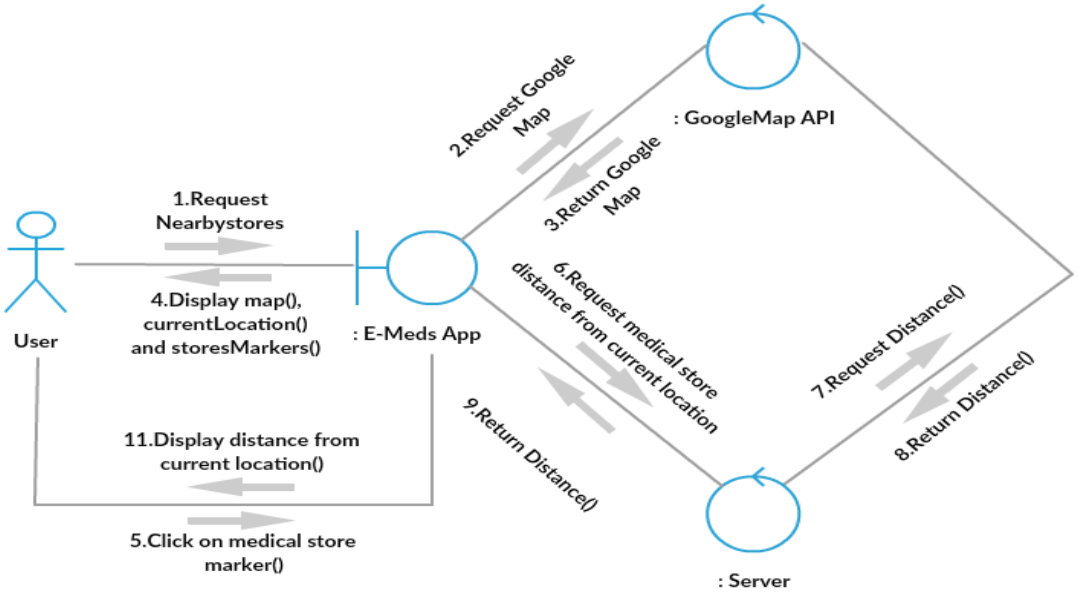

Figure 5.19: Collaboration Diagram – Check Nearby Stores - This figure describes the object interact with the functionality with the sequence of the scenario. User object performing a functional task and the other object play a part to fulfill the tasks. In this Check Nearby Stores process user, E-meds app, server and Google Map API are involved.

5.5 Other UMLs

No other UMLS

5.6 ERD

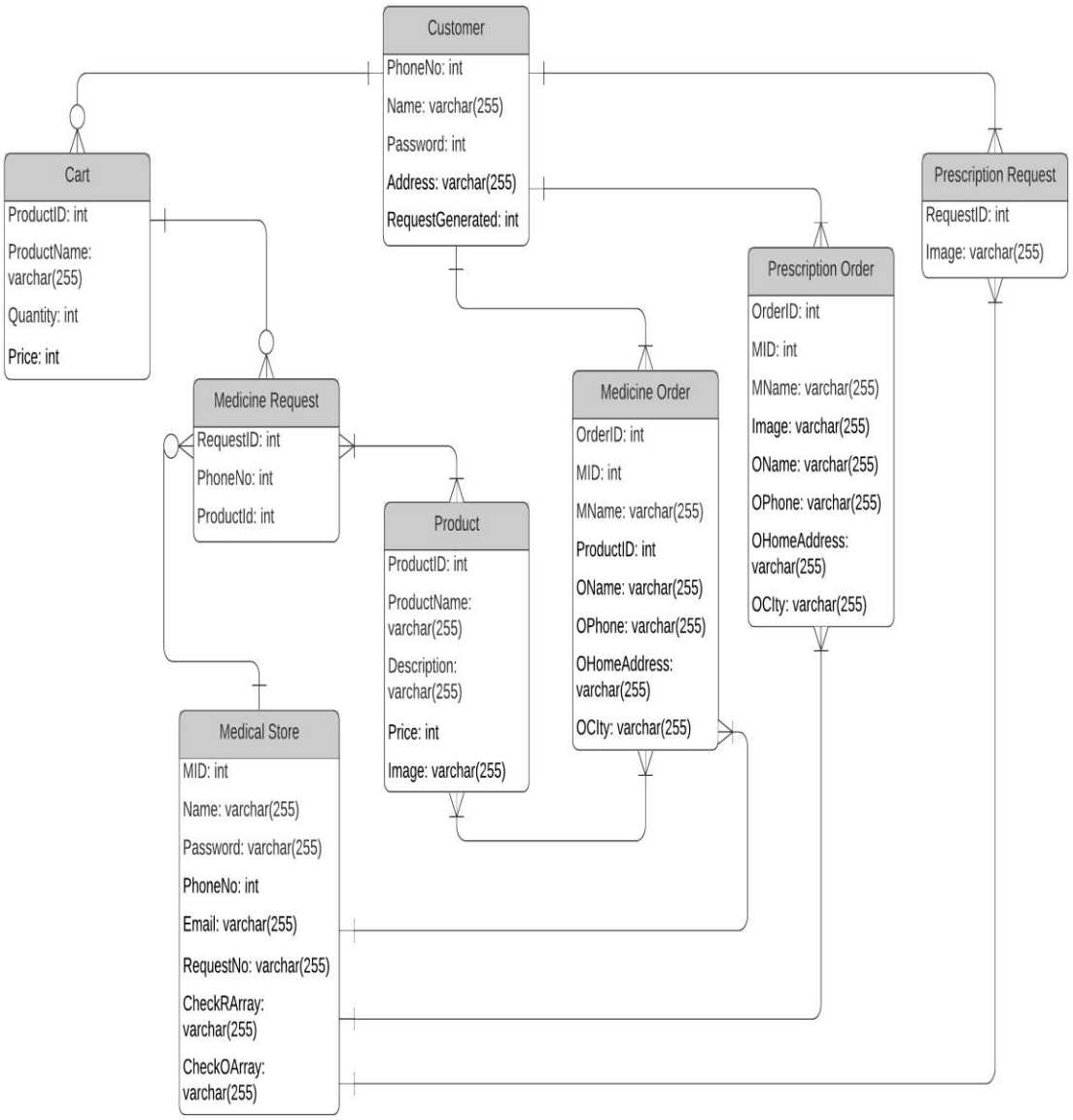

Figure 5.20: ERD Diagram - This figure describes that there are many objects, which are, connect with some type of relations. Every object has some attributes. Customer is an object and they have also some attributes. Customer object are connect with cart object. The customer user has one to many relations with cart object. Cart has also one to many relations with the medicine request whereas prescription request object has many to one relation with the customer object. The medical store user has one to many relations with prescription request, prescription order, and medicine request and medicine order.

CHAPTER # 6

6. IMPLEMENTATIoN DETAILS

6.1 Development Setup

The tools we use for the development of our project are as follow:

- Android Studio
- Firebase

Android Studio: We use android studio for the development of mobile application. Android Studio has intelligent code editor. ***Android Studio*** makes code implementation and code analysis faster and easier. Android Studio has faster programming and testing.

Firebase: We utilize Firebase for database. The firebase highlights permit you to assess client behavior, report on broken joins, and client intelligent and all other viewpoints of client encounter and user interface. In Firebase we are able illuminate execution issues of an app by settling mistakes right from backend arrangement. It is mindful of capturing mistakes for iOS apps and both lethal and non-fatal bugs for Android application. Crashes and errors reports are free with Firebase. You ought not to pay cash to induce this highlight.

Visual Studio Code: Visual Studio Code highlights as a perfect source code editor, simple for day-to-day utilize. With support for numerous programing languages, Visual Studio Code makes a difference us to right away deliver with language structure highlight highlights, bracket-matching highlights, programmed space and more.

Google Map API: We use Google Map API to check nearby medical stores for the ease of customers, so they can connect with the medical stores without any problem.

One Time Password (OTP): OTP Verification is use to verify the Email Address or Phone Number of users by sending **verification** code (**OTP**) to their phone numbers during registration. It removes the fear of a user registering with Fake Email Address or Phone Number.

The technologies we use for the development of our project are as follow:

- Java for android
- JavaScript for web

Java for android: application engineers who are focusing on Android showcase or the Android stage are for the most part utilizing Java. It make advancement simple for engineers so they can get the application once in Java and run it in on other stage that underpins Java improvement. Java can be introduced utilizing restricted authorization subsequently will not hurt your computer.

JavaScript for web: The JavaScript is simple to memorize and offers sentence structure that's near to English. When the source code is executed on the client side the anticipated comes about and forms are completed in less time depending on the errand.

6.2 Deployment setup

First we did work on react native and MERN stack, but we didn't feel comfortable with these technologies so we changed our technologies and for that reason we also change our documentation. We also face problem in Google API while implementing it.

6.3 Algorithms

```
@Override
   public boolean onOptionsItemSelected(MenuItem item) {


      int id = item.getItemId();


      if (id == R.id.top_action_cart) {
         // add your action here that you want
         Intent intent = new Intent(MainActivity.this, CartActivity.class);
         startActivity(intent);


      } else if (id == R.id.top_menu_settings) {
         Intent intent = new Intent(MainActivity.this, SettinsActivity.class);
         startActivity(intent);


      } else if (id == R.id.top_menu_about_us){
         Intent intent = new Intent(MainActivity.this, AboutUsActivity.class);
         startActivity(intent);


      } else if (id == R.id.top_menu_privacy_policy){
         Intent intent = new Intent(MainActivity.this, PrivacyPolicyActivity.class);
         startActivity(intent);


      } else if (id == R.id.top_menu_logout){
         // add your action here that you want



         new AlertDialog.Builder(MainActivity.this)
               .setTitle("Confirm")
               .setIcon(R.drawable.ic_exit_to_app_red_24dp)
               .setMessage("Are you sure you want to Logout?")
```

```
            .setPositiveButton("Logout", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    loadingBar.setTitle("Logout");
                    loadingBar.setMessage("Please wait, while we logging you out.");
                    loadingBar.setCanceledOnTouchOutside(false);
                    loadingBar.show();

                    Runnable progressRunnable = new Runnable() {

                        @Override
                        public void run() {
                            loadingBar.dismiss();
                            Paper.book().destroy();
                            FirebaseAuth auth = FirebaseAuth.getInstance();
                            auth.signOut();
                            Intent intent = new Intent(MainActivity.this, FirstActivity.class);
                            startActivity(intent);
                            finish();
                        }
                    };

                    Handler pdCanceller = new Handler();
                    pdCanceller.postDelayed(progressRunnable, 3000);

                }
            })
            .setNegativeButton("No", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            })
            .show();
    }

    return true; }
```

6.4 Constraints

6.4.1 Assumptions

Before the main integration we assume the following things:

- Every user has the basic knowledge of running an app.
- The main area that are frequently customers of medical stores.
- Every person has an android mobile.

6.4.2 Internet Connectivity is optimized.System constraints

Constraints in the project are:

- Multiple payment methods
- No Chat Option
- Single Language Option

6.4.3 Runtime Feedback Restrictions

Some Of the Main Restrictions are given below:

- We don’t provide delivery option, medical stores have to deliver the medicine to the customers.
- The user can’t check the medicine availability from a single store.

6.4.4 Limitations

The system has some of the following limitations:

- Initially it is not build for large databases.
- It is not on IOS Systems.
- Multi Language Integration.

CHAPTER # 7

7. TESTING

7.1 Extended Test Cases

Table 7.1: Test Case (Register) This test case is for checking the user register on the account. For this, the users enter a valid mobile number and verify the phone number with the help of verification code. After that users enter their name, address and other general information.

Test Case ID		01		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Register user account
Test Description				This test check that the user can register their account	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
Check if the user can register his account or not.				Name: Ahmad Mobile number:0300-1234567 Password: abcd1234	
Pre-Condition				User wants to create account.	
Post-Condition				The user account created successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on join now button.	New page will open and user will enter their name, address and password.	A new page open and user enters his information.	
3	Click on next button.	New page will open and user enter their mobile no and get a verification code	Users enter the verification code and register themselves.	PASS

Table 7.2: Test Case (Login) This test checks that the user can login into their account. For that the user enters the verified mobile number and password. If the mobile number is valid and password is correct then user successfully enters into the account otherwise not.

Test Case ID		02		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Login User Account
Test Description				This test checks that the user can login into their account	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
Checks if the user can login into his account.					
Pre-Condition				User wants to login.	
Post-Condition				User login the account successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on already have an account button.	New page will open and user will enter his mobile number and password.	A new page open and user enters his information and click on login button.	
3	Click on login button.	New page will open and user into the application.	After entering login button user successfully enter into their account.	PASS

Table 7.3: Test Case (Check Nearby Stores) This test case is used to navigate the nearby medical stores. Firstly the user will give location access to the application, and click on search nearby stores. After that all the nearby stores will be shown to the user. So the user can easily purchase medicine from their desired medical store.

Test Case ID		03		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Check Nearby Medical Stores.
Test Description				This test checks the nearby medical stores.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
Checks the nearby medical stores.					
Pre-Condition				User will give location access to the application.	
Post-Condition				Nearby medical stores will be shown to the user.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the nearby stores button.	Map will open and user will give location access to the application.	User pin his current location and nearby stores are now shown successfully.	
3		All the nearby stores will show to the user with distance.	Nearby medical stores show to the user with distance from their pin location.	PASS

Table 7.4: Test Case (Search Medicines) This test checks the searching of the medicines. User wants to search the medicines by clicking on the search bar. User type on the search bar and the medicines, which are added on the application, will show to the user. If the medicine will not show then it means that the test case is failed.

Test Case ID		04		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Search Medicines.
Test Description				This test checks the searching of medicines.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
Checking the searching of medicines.					
Pre-Condition				User wants to search medicines.	
Post-Condition				Search result will be shown to the user.	
Step #	**Test Step**	**Expected Results**		**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.				
2	Click on the search bar.	New page will open and user will search the medicines.		User enters the medicine name in the search bar.	
3		The search medicines will be shown to the user with price.		The result will show to the user.	PASS

Table 7.5: Test Case (Check Medicines Availability) This test case is used to check the medicine availability. User sends request to the medical store about the medicines, so user will know that their desired medicines are available at which medical store.

Test Case ID		05		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Check medicine availability.
Test Description				This test checks that the medicines are available on the stores or not.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
This test checks the availability of medicines.					
Pre-Condition				User wants to check the availability of medicines.	
Post-Condition				User checks the availability of medicines successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the search bar and search medicines.	After searching medicines the result will show to the user.	Result show successfully.	
3		User will send request to all the medical stores, and the medical stores will respond accordingly that the medicines are available or not.	User sends medicine availability request and the store respond that the medicines are available or not.	PASS

Table 7.6: Test Case (Upload Prescription) This test case is for uploading prescription. There are some medicines, which cannot be delivered without prescription. First user have to upload prescription and then purchase medicines.

Test Case ID		06		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Upload Prescription
Test Description				This test checks that the user will upload the prescription or not.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test user will upload prescription.					
Pre-Condition				User wants to upload prescription.	
Post-Condition				Prescription is uploaded successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on upload prescription button.	Now user will have the option to upload prescription by taking picture or choosing from gallery.	User have two options take picture or from gallery	
3		After choosing the prescription picture, user clicks the request button, and the prescription will uploaded successfully.	The requested picture uploaded successfully.	PASS

Table 7.7: Test Case (Check Prescription Availability) This In this test case user wants to test the prescription availability by sending request to the medical store by uploading the prescription. So the medical store who have the medicine will respond.

Test Case ID		07		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Check prescription availability.
Test Description				This test is used to check the prescription availability.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test user wants to check the prescription Availability					
Pre-Condition				User wants to check prescription availability.	
Post-Condition				User checks the availability of medicine successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the check prescription availability button.	New page will open and user will check on which medical stores prescription medicines are available or not.	After the process user successfully checks the prescription medicines are available on which medical stores.	PASS

Table 7.8: Test Case (Update Profile) This test case is for updating the profile. When the new page will open user can update the name, address, mobile number and password. If these information of the user will update then the test case will pass.

Test Case ID		08		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Update profile.
Test Description				This test is used to update profile	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test user update their profile.					
Pre-Condition				User will update their profile	
Post-Condition				Profile updated successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Clicks on the update profile button.	New page will open with a form of new name, new address and new password.	After the process new page open with a form.	
3		User will update their profile with new name, password and adderss and click on update profile button.	After updating the information profile is updated successfully.	PASS

Table 7.9: Test Case (Confirm Medicines Request) This test case is used for confirm medicine request. This test case is for medical store user, when the user send request for their desired medicines and the medical store will respond to the request, that the desired medicines are available or not.

Test Case ID		09		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Confirm medicines request.
Test Description				This test is used to confirm medicines request.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test medical store want to confirm medicines request.					
Pre-Condition				Medical store will confirm medicines request.	
Post-Condition				Medical store confirm the request successfully.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the confirm request button.	The list of medicine request will show to the medical store.	All the requested medicines have been shown to the medical store.	
3		The response will be sending to the user by the medical store that the requested medicines are available or not.	After the process user successfully checks the response by the medical store.	PASS

Table 7.10: Test Case (Confirm Medicines Order) This test case is for confirm medicine order. When the user send request for medicine availability to the entire medical store and the medical store response, after that the user send order now the medical store have to confirm the order, for that they again have to response to the user that their medicine order is confirm.

Test Case ID		10		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Confirm medicine order.
Test Description				This test is used to confirm the medicine order.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test medical store will confirm the order.					
Pre-Condition				Medical store want to confirm the order.	
Post-Condition				Order will confirm.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the confirm order button.	The list of medicine orders will be shown to the medical store.	All the medicines orders from the users have been shown to the medical store.	
3		The response will be sending to the user by the medical store that your order is confirmed.	After the process user successfully checks the response by the medical store.	PASS

Table 7.11: Test Case (Confirm Prescription Request) This test case is for confirm prescription request. User can also send request of medicine with prescription photo, so it is easy for the medical store to deliver prescription medicine. Sometimes it is difficult for the patient to read the name of the medicine, now with the help of prescription option user can upload prescription and the medical store have to confirm the prescription request.

Test Case ID		11		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Confirm prescription request.
Test Description				This test is used to confirm the prescription request.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test medical store will confirms the prescription request.					
Pre-Condition				Medical store wants to confirm the prescription request.	
Post-Condition				Prescription request will confirm.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the confirm prescription request button.	The list of prescription medicines will be shown to the medical store.	The list of prescription medicines has been shown to the medical store.	
3		New page will open and medical store will confirm the prescription request send by the user to the medical store.	After the process user will successfully checks the prescription medicines are available or not.	PASS

Table 7.12: Test Case (Confirm Prescription Order) After sending the prescription request to the medical store and after the response from the medical store side, now the user have to send the order to the medical store and the medical store have to confirm the order and send the respond back to the user.

Test Case ID		12		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Confirm prescription order.
Test Description				This test is used to Confirm prescription order.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test medical store wants to Confirm prescription order.					
Pre-Condition				Medical store want to Confirm prescription order.	
Post-Condition				Medical store Confirm prescription order.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on Confirm prescription order button.	The list of prescription medicines will be shown to the medical store.	The list of prescription medicines has been shown to the medical store.	
3		New page will open and user will Confirm the prescription order send by the user.	After the process user successfully checks the confirmation of the prescription order.	PASS

Table 7.13: Test Case (Manage Medicines) The admin have to manage the medicine. The admin responsibility is to add or delete the medicine from the database. If the stock of particular medicine is ended, so it's the duty of the admin to delete the medicine and add new medicine.

Test Case ID		13		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Manage medicines.
Test Description				This test is used to manage medicines.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test admin wants to manage medicines.					
Pre-Condition				Admin want to manage medicines.	
Post-Condition				Admin manage all the medicines.	

Step #	**Test Step**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.			
2	Click on the manage medicines button.	New page will open and admin will start managing medicines.	After the process admin will start processing all the medicines.	PASS

Table 7.14: Test Case (Manage Users) is the responsibility of the admin to manage users. For example if user wants to change their mobile number, so all the data has to be managed by the admin.

Test Case ID		14		**Created By**	Umair
Reviewed By		Zaidi		**Test Module Name**	Manage Users.
Test Description				This test is used to Manage users.	
Tester's Name	Fahad	**Test Date**	22/03/20	**Test Priority**	High
Test Title				**Test Data**	
In this test admin wants to manage the application users.					
Pre-Condition				Admin wants to manage users.	
Post-Condition				User will be managed by the admin.	
Step #	**Test Step**	**Expected Results**		**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	Click on the E-Meds application.				
2	Click on manage user button.	New page will open and admin will manage the users by providing the information.		After the process user will be managed by the admin.	PASS

7.2 Decision Table

7.2.1 Code snippet

```
public class MapsActivity extends FragmentActivity implements GoogleMap.OnMarkerClickListener, OnMapReadyCallback {

    //Marker Code (decleration) below

    private static final LatLng RoyalPharmacy = new LatLng(31.460007,74.282841);
    private static final LatLng Servaid = new LatLng(31.420507,74.265415);
    private static final LatLng ParagonPharmacy = new LatLng(31.435330,74.269094);
    private static final LatLng SehatPharmacy = new LatLng(31.436932,74.268332);
    private static final LatLng GreenPharmacy = new LatLng(31.432145,74.265623);
    private static final LatLng PunjabPharmacy = new LatLng(31.472187,74.309076);
    private static final LatLng BashirPharmacy = new LatLng(31.459174,74.308411);
    private static final LatLng SamiPharmacy = new LatLng(31.469369,74.299469);
    private static final LatLng FirstCarePharmacy = new LatLng(31.451825,74.286927);
    private static final LatLng NovartisPharmacy = new LatLng(31.434589,74.268279);
    private static final LatLng MehmoodPharmacy = new LatLng(31.441262,74.274963);
    private static final LatLng ShakooriPharmacy = new LatLng(31.441628,74.276250);
    private static final LatLng PunjabPharmacy2 = new LatLng(31.442214,74.276459);
    private static final LatLng LiakatHospitalPharmacy = new LatLng(31.444205,74.277682);
    private static final LatLng EverCarePharmacy = new LatLng(31.437331,74.281321);
    private static final LatLng FazalDinPharmacy = new LatLng(31.439541,74.283810);
    private static final LatLng Servaid2 = new LatLng(31.438443,74.286787);
    private static final LatLng MadinaPharmacy = new LatLng(31.441276,74.293241);
    private static final LatLng ZiaPharmacy = new LatLng(31.451852,74.298739);
    private static final LatLng MananPharmacy = new LatLng(31.456740,74.297999);
    private static final LatLng Servaid3 = new LatLng(31.458707,74.303277);
    private static final LatLng IqraPharmacy = new LatLng(31.467872,74.305729);
    private static final LatLng AhsanPharmacy = new LatLng(31.474822,74.301706);
    private static final LatLng MehmoodSonPharmacy = new LatLng(31.434598,74.300907);
    private static final LatLng LahorePharmaPharmacy = new LatLng(31.433793,74.300929);
    private static final LatLng AdilPharmacy = new LatLng(31.435495,74.309694);
```

```
    private static final LatLng HealthCarePharmacy = new LatLng(31.438855,74.311422);
    private static final LatLng BHUPharmacy = new LatLng(31.445049,74.300953);
    private static final LatLng MedixPharmacy = new LatLng(31.445937,74.306350);
    private static final LatLng PharmacyHall = new LatLng(31.443233,74.305537);
    private static final LatLng Servaid4 = new LatLng(31.441622,74.323015);
    private static final LatLng AlrehmanPharmacy = new LatLng(31.440047,74.307694);
    private static final LatLng ClinixPharmacy = new LatLng(31.463805,74.319796);
    private static final LatLng RahatPharmacy = new LatLng(31.466441,74.308295);
    private static final LatLng FazalDinPharmacy2 = new LatLng(31.468843,74.306991);
    private static final LatLng Servaid5 = new LatLng(31.489075,74.326176);
    private static final LatLng BioCarePharmacy = new LatLng(31.462759,74.318129);
    private static final LatLng ClinixPharmacy2 = new LatLng(31.456451,74.278251);
    private static final LatLng HijabClinicPharmacy = new LatLng(31.455711,74.279553);
    private static final LatLng ZohaibPharmacy = new LatLng(31.443895,74.267387);
    private static final LatLng Servaid6 = new LatLng(31.469539,74.273546);
    private static final LatLng PunjabPharmacy3 = new LatLng(31.479421,74.280492);
    private static final LatLng ClinixPharmacy3 = new LatLng(31.483033,74.298100);
    private static final LatLng GreenPharmacy2 = new LatLng(31.483143,74.298561);
    private static final LatLng FazalDinPharmacy3 = new LatLng(31.483408,74.298969);
    private static final LatLng NeuroClinicPharmacy = new
LatLng(31.479090,74.300074);
    private static final LatLng AlAfzalPharmacy = new LatLng(31.473426,74.298636);
    private static final LatLng AbbotPharmacy = new LatLng(31.466053,74.252865);
    private static final LatLng NawabPharmacy = new LatLng(31.464863,74.250151);
    private static final LatLng LahoreCarePharmacy2 = new
LatLng(31.449195,74.212031);
    private static final LatLng DWatsonPharmcy = new LatLng(31.500624,74.320149);
    private static final LatLng AABPARAPharmcy = new LatLng(31.508823,74.339897);
    private static final LatLng DecentPharmacy = new LatLng(31.475577,74.338343);
    private static final LatLng FazalDinPharmacy4 = new LatLng(31.467626,74.316354);



    private Marker mRoyalPharmacy;
    private Marker mServaid;
    private Marker mParagonPharmacy;
```

```
private Marker mSehatPharmacy;
private Marker mGreenPharmacy;
private Marker mPunjabPharmacy;
private Marker mBashirPharmacy;
private Marker mSamiPharmacy;
private Marker mFirstCarePharmacy;
private Marker mNovartisPharmacy;
private Marker mMehmoodPharmacy;
private Marker mShakooriPharmacy;
private Marker mPunjabPharmacy2;
private Marker mLiakatHospitalPharmacy;
private Marker mEverCarePharmacy;
private Marker mFazalDinPharmacy;
private Marker mServaid2;
private Marker mMadinaPharmacy;
private Marker mZiaPharmacy;
private Marker mMananPharmacy;
private Marker mServaid3;
private Marker mIqraPharmacy;
private Marker mAhsanPharmacy;
private Marker mMehmoodSonsPharmacy;
private Marker mLahorePharmaPharmacy;
private Marker mAdilPharmacy;
private Marker mHealthCarePharmacy;
private Marker mBhuPharmacy;
private Marker mMedixPharmacy;
private Marker mPharmacyHall;
private Marker mServaid4;
private Marker mAlRehmanPharmacy;
private Marker mClinixPharmacy;
private Marker mRahatPharmacy;
private Marker mFazalDinPharmacy2;
private Marker mServaid5;
private Marker mBioCarePharmacy;
private Marker mClinixPharmacy2;
```

```
private Marker mHijabClinicPharmacy;
private Marker mZohaibPharmacy;
private Marker mServaid6;
private Marker mPunjabPharmacy3;
private Marker mClinixPharmacy3;
private Marker mGreenPharmacy2;
private Marker mFazalDinPharmacy3;
private Marker mNeuroClinicPharmacy;
private Marker mAlAfzalPharmacy;
private Marker mAbbotPharmacy;
private Marker mNawabPharmacy;
private Marker mLahoreCarePharmacy2;
private Marker mDWatsonPharmacy;
private Marker mAABPARAPharmacy;
private Marker mDecentPharmacy;
private Marker mFazalDinPharmacy4;
    }
```

7.2.2 Decision coverage table

Table 7.15: Decision coverage table for login This decision table explained different expected results like if the user not entering the e-mail or password what will be the results.

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
E-mail	F	T	F	T
Password	F	F	T	T
Expected Results	Error: Please enter e-mail	Error: Please enter password	Error: Please enter e-mail	Login proceed

Table 7.16: Decision coverage table for Placing Order This decision table explained different expected results like that the user is a registered user or not, and if the user is registered he/she can place order and if not a registered user then he/she can't place order.

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
Registered User?	F	F	F	F	T	T	T	T
Place Order	F	F	T	T	F	F	T	T
Expected Results	No	No	No	No	No	No	No	Yes

Table 7.17: Decision coverage table for Uploading Prescription This decision table explained different expected results like the format of the prescript image, the resolution of the image and provided the expected results by giving different types of format and resolution.

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
Format	.jpg	.jpg	.jpg	.jpg	Not .jpg	Not .jpg	Not .jpg	Not .jpg
Size	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb
Resolution	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177
Expected Results	Photo uploaded	Error: Message resolution mismatch	Error: Message size mismatch	Error: Message size and resolution mismatch	Error: Message for format mismatch	Error: Message format and resolution mismatch	Error: Message for format and size mismatch	Error: Message for format, size, and resolution mismatc

7.3 Traceability Matrix

7.3.1 RID vs UCID (requirements vs use cases)

Table 7.18: RID vs UCID (requirements vs use case) this traceability matrix is in between use-cases and the requirements that the use-case fulfill the given requirements. The usecases and the requirements are fully explained on the previous tables along with their description.

UCID/RID	**R1-01**	**R1-02**	**R1-03**	**R1-04**	**R1-05**	**R1-06**	**R1-07**	**R1-08**	**R2-01**	**R2-02**	**R2-03**	**R2-04**	**R3-01**	**R3-02**	**R3-03**
UC 01	✓														
UC 02		✓													
UC 03			✓												
UC 04				✓											
UC 05					✓										
UC 06							✓								
UC 07						✓									
UC 08								✓							
UC 09									✓						
UC 10										✓					
UC 11											✓				
UC 12												✓			
UC 13														✓	
UC 14													✓		
UC 15															✓

7.3.2 Prototypes (RID vs PID)

Table 7.19: Prototypes (RID vs PID) this traceability matrix is in between prototypes and the requirements that the prototypes fulfill the given requirements. The prototypes and the requirements are fully explained on the previous tables along with their description.

PID/RID	**R1-01**	**R1-02**	**R1-03**	**R1-04**	**R1-05**	**R1-06**	**R1-07**	**R1-08**	**R2-01**	**R2-02**	**R2-03**	**R2-04**	**R3-01**	**R3-02**	**R3-03**	**R3-03**
PID 1	✓	✓														
PID 2	✓															
PID 3	✓															
PID 4		✓														
PID 5	✓	✓														
PID 6			✓													
PID 7			✓													
PID 8			✓													
PID 9			✓													
PID 10				✓												
PID 11	✓	✓	✓	✓												
PID 12	✓	✓	✓	✓												
PID 13	✓	✓	✓	✓												
PID 14							✓									
PID 15							✓									
PID 16							✓									
PID 17										✓	✓	✓	✓			
PID 18										✓	✓	✓	✓			
PID 19											✓					
PID 20								✓								
PID 21								✓								

7.3.3 Test Cases (RID vs TID)

Table 7.20: Test Cases (RID vs TID) this traceability matrix is in between test cases and the requirements that the test case fulfill the given requirements. The test cases and the requirements are fully explained on the previous tables along with their description.

TID/RID	**R1-01**	**R1-02**	**R1-03**	**R1-04**	**R1-05**	**R1-06**	**R1-07**	**R1-08**	**R2-01**	**R2-02**	**R2-03**	**R2-04**	**R3-01**	**R3-02**	**R3-03**
TID 01	✓														
TID 02		✓													
TID 03			✓												
TID 04				✓											
TID 05					✓										
TID 06						✓									
TID 07							✓								
TID 08								✓							
TID 09									✓						
TID 10										✓					
TID 11											✓				
TID 12												✓			
TID 13														✓	
TID 14													✓		
TID 15															✓

7.3.4 Coverage (UCID vs TID)

Table 7.21: Test Cases (UCID vs TID) this traceability matrix is in between use-cases and the test cases that the use-case fulfill the requirements of test-cases. The use-cases and the test cases are fully explained on the previous tables along with their description.

UCID/TID	**TID 01**	**TID 02**	**TID 03**	**TID 04**	**TID 05**	**TID 06**	**TID 07**	**TID 08**	**TID 09**	**TID 10**	**TID 11**	**TID 12**	**TID 13**	**TID 14**	**TID 15**
UCID 01	✓														
UCID 02		✓													
UCID 03			✓												
UCID 04				✓											
UCID 05					✓										
UCID 06							✓								
UCID 07						✓									
UCID 08								✓							
UCID 09									✓						
UCID 10										✓					
UCID 11											✓				
UCID 12												✓			
UCID 13													✓		
UCID 14														✓	
UCID 15															✓

CHAPTER # 8

8. RESULTS/OUTPUT/STATISTICS

8.1 %completion

All the requirements are fulfilled according to the table 7.3.1

Table 8.1: Matrix (8.1) % Completion we have completed our project 80%. We met all the functional requirements that we discussed.

UCID/RID	**R1-01**	**R1-02**	**R1-03**	**R1-04**	**R1-05**	**R1-06**	**R1-07**	**R1-08**	**R2-01**	**R2-02**	**R2-03**	**R2-04**	**R3-01**	**R3-02**	**R3-03**
UC 01	✓														
UC 02		✓													
UC 03			✓												
UC 04				✓											
UC 05					✓										
UC 06							✓								
UC 07						✓									
UC 08								✓							
UC 09									✓						
UC 10										✓					
UC 11											✓				
UC 12												✓			
UC 13														✓	
UC 14													✓		
UC 15															✓

8.2 %accuracy

All requirements have been implemented correctly according to the matrix 7.7.7

Table 8.2: Matrix (8.2) % Accuracy our project is working 85% accurate. It fulfills all the functional and non-functional requirements.

TID/RID	**R1-01**	**R1-02**	**R1-03**	**R1-04**	**R1-05**	**R1-06**	**R1-07**	**R1-08**	**R2-01**	**R2-02**	**R2-03**	**R2-04**	**R3-01**	**R3-02**	**R3-03**
TID 01	✓														
TID 02		✓													
TID 03			✓												
TID 04				✓											
TID 05					✓										
TID 06						✓									
TID 07							✓								
TID 08								✓							
TID 09									✓						
TID 10										✓					
TID 11											✓				
TID 12												✓			
TID 13														✓	
TID 14													✓		
TID 15															✓

8.3 %correctness

All the requirements have been tested to be conforming to requirements according to the matrix 7.3.4

Table 8.3: Matrix (8.3) % Correctness As we have tested all the requirements and made the test cases clear and we removed all the mistakes so now our project is 90% correct.

UCID/TID	**TID 01**	**TID 02**	**TID 03**	**TID 04**	**TID 05**	**TID 06**	**TID 07**	**TID 08**	**TID 09**	**TID 10**	**TID 11**	**TID 12**	**TID 13**	**TID 14**	**TID 15**
UCID 01	✓														
UCID 02		✓													
UCID 03			✓												
UCID 04				✓											
UCID 05					✓										
UCID 06							✓								
UCID 07						✓									
UCID 08								✓							
UCID 09									✓						
UCID 10										✓					
UCID 11											✓				
UCID 12												✓			
UCID 13													✓		
UCID 14														✓	
UCID 15															✓

9. CONCLUSION

We provide ease to the customers, so they can get all types of medicine at their doorstep with the help of mobile phones. E-Meds, an online medicine delivery marketplace, connects the customers to place orders from a huge selection of medical stores. This way medical store will get business and customers will get full comfort and ease. People now a day suffers a lot in finding the medicine from different medical stores and it is possible that the medicine is not available on the stores (means he get the medicine after visiting 2 to 3 stores), so it is a process of time wasting. We gather all the medical stores on one platform so the customers can check that at which medical store all his desired medicines are available so they can take medicines directly or get a delivery. It is easy for the customers that they can find the medicine easily with the help of our application.

10. FUTURE WORK

We expand our idea in future by deploying the application in market, because there is no such application in the market who facilitate the customers, who wants to buy medicines through various medical stores with the help of an application. We are not providing delivery facility at that time, but we will provide the delivery services to all the medical stores in future. As we all know that Food Panda has their own riders. We will provide the same services as Food panda for example those medical stores that don't have delivery boys, so we provide them the delivery boys. We will also build an inventory for other medical equipments like wheel chairs, bood pressure machine, sugar checker machine etc. If user want to purchase these equipments they should order us. We will also add discount alert. A user will be notified if there will be discount at nearby medical stores. We also add AI Bot, in which user should tell their symtoms and the system will give recommended medicines according to their disease. This system is not authentic, this is only for informative purpose.

11. BIBLIOGRAPHY

11.1 Books

The lists of books that we read are:

- Eloquent JavaScript: A Modern Introduction to Programming
- Android Application Development for Dummies

11.2 Articles

The list is given below:

- https://www.techspot.com/tag/android/
- https://developer.android.com/
- https://proandroiddev.com/

11.3 Other References

The website links to previous service Giants are given below:

- https://medicalstore.com.pk/
- https://sehat.com.pk/
- https://dawaai.pk/

12. APPENDIX

12.1 Glossary of terms

A

Assumption: To expect or made guess about something.

Accuracy: How precise is something.

Android: Android is an operating system used in various handsets available now a day to run apps and kernel as well.

Android Studio:

Android studio is a tool that is used to develop android apps

C

Constraint: To restrict from something.

D

DFD: Data Flow Diagram

Domain: A specified area of knowledge.

Database: Database is a storage scheme for data that is entered in the app

E

Event Listener: A prebuilt function in java to run on an event occurrence

F

Feedback: Response of something, to comment.

Firebase: Firebase is the database integration setup and database as well used worldwide for android app development.

I

Interface: Interaction with the other person, system etc.

P

Prototype: A model of something

S

Stakeholder: A person directly or indirectly involved with the system.