

## Final Year Project Report

# MEDIQUICK

Find Medicines Nearby

**BESE (Batch: 2016-17)**

## Project Advisor

Engr. Dr. Raheela Asif  
Associate Professor, Department of Software Engineering  
NEDUET

## Submitted by

Muhammad Usman Sabir	SE-051
Yusra Wasi	SE-010
Muneeb Ahmed Khan	SE-066



**DEPARTMENT OF SOFTWARE ENGINEERING**

## **PREFACE**

---

This report comprises the summary of our work that we have achieved during our Final Year Project (FYP). We have chosen to develop an application that provides users to search medicines at their nearby retailers and reserve order. They can verify authenticity of any medicine they have purchased. This application also benefits the pharmaceutical companies by enabling them to manage their supply chain in a secure manner and eliminate counterfeit drugs in market of their company.

## **ACKNOWLEDGMENT**

---

In the name of Almighty Allah, the most merciful and the most beneficial, without whose blessings this would never have been possible.

First and above all we would like to thank Almighty Allah (SWT) for providing us this opportunity and blessings to finish this study successfully. We owe this project to many persons who helped and guided us during the development of this project. We would like to express our special appreciation and gratitude to our internal advisor **Engr. Dr. Raheela Asif** for her warm encouragement, thoughtful guidance and advices. Without their guidance this would not have been possible to achieve our goal. We highly admire the support, guidance, encourage, motivation and even scold of all of our teachers who taught and helped us during our academic time. We would also sincerely appreciate the whole-hearted co-operation and valuable help rendered by the non-teaching staff of university.

## GROUP MEMBERS

---



MUHAMMAD USMAN SABIR

SE-051

A certified blockchain and cloud engineer with industrial experience in backend development, client dealing and business analysis.

- ✓ Group Leader
- ✓ Blockchain Design & Development
- ✓ Frontend Development
- ✓ Architectural & Research Work
- ✓ API Integration
- ✓ Server-Side Development

Immediate Contact: 0303-0388456, [usmansabir98@gmail.com](mailto:usmansabir98@gmail.com)



YUSRA WASI

SE- 010

A certified Quality Assurance Engineer with industrial experience in Manual and Automated Testing.

- ✓ Server-Side Development
- ✓ API Testing
- ✓ Report Writing
- ✓ Database Design & Management
- ✓ System Design

Immediate Contact: 03112266294, [wasiyusra@gmail.com](mailto:wasiyusra@gmail.com)



MUNEEB AHMED KHAN

SE-066

A certified React Developer with expertise in developing scalable frontend architecture

- ✓ Frontend Development
- ✓ UI/UX Design and GUI
- ✓ API Integration
- ✓ Server-Side Development

Immediate Contact: 0336-1260355, [muneebahmed\\_127@hotmail.com](mailto:muneebahmed_127@hotmail.com)



**NED University of Engineering and Technology, Karachi**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**CERTIFICATE OF COMPLETION**

This is to certify that the following students

MUHAMMAD USMAN SABIR	SE-051
YUSRA WASI	SE-010
MUNEEB AHMED KHAN	SE-066

have successfully completed their final year project titled

**MEDIQUICK – FIND MEDICINES NEARBY**

in the partial fulfillment for the requirements of the Degree of Bachelor of Software Engineering during the academic session 2015-2016.

---

**Engr Dr. Raheela Asif**  
Associate Professor, Department of  
Software Engineering, NEDUET

---

**Professor Dr.Najmi Ghani Haider**  
Chairman, Department of Software  
Engineering, NEDUET

# **ABSTRACT**

---

**Opportunity:** Have you ever searched for a medicine but could not find it anywhere? Needed it quickly but had no idea which drugstore currently has it? Tired of inquiring at every other medical store with no success? Moreover, purchasing a medicine but unsure of its authenticity and facing problems due to counterfeit drugs? These are some of the questions that motivated us to propose a solution to the problem persistent in the pharmaceutical market.

**Mission:** We aim to eradicate the supply of counterfeit drugs with the game changing technology of blockchain that rightly serves the purpose of transparency, immutable tracking of assets and audit of drugs being manufactured and distributed. In addition, prompt supply of life saving and OTC drugs is our key motive.

**Our Solution:** We propose a cloud-based system that integrates multiple pharmaceutical companies, pharmacies and drug stores on a platform that provides the functionality to automate their end-to-end primary and tertiary sales. Multiple pharmacies and drug stores can choose to manage their inventories on our integrated module allowing them to optimize their stock and increase their sales. This will allow end-users (general public) to know the availability of their desired medicines and get easy access to nearby drug stores. Furthermore, we aim to incorporate a decentralized module on blockchain that will serve the purpose of keeping track of the medicines being manufactured and distributed, on its immutable ledger. This will ensure the traceability of each medicine at multiple points of supply chain.

**Market Focus:** For consumers, our target market would be focused around middle aged tech savvy people who are living a moderately privileged lifestyle. For lower classes and non-tech people, we aim to introduce our system to clinics with in-house pharmacies or attendants to direct their patients through our system. For businesses, we initially aim to target small and medium scale enterprises having little to no technological solutions deployed for their sales and inventory management.

# TABLE OF CONTENTS

---

<b>Contents</b>	
Final Year Project Report .....	1
PREFACE .....	2
ACKNOWLEDGMENT.....	3
GROUP MEMBERS .....	4
ABSTRACT.....	7
TABLE OF CONTENTS.....	8
LIST OF TABLES .....	11
LIST OF FIGURES .....	12
CHAPTER 1 – INTRODUCTION .....	17
1.1 Goals and objectives .....	17
1.2 System statement of scope.....	17
1.3 System context.....	18
1.4 Theoretical Background (of project) .....	21
1.5 Technology & Tools/hardware components (used in the Project) .....	22
CHAPTER 2 – USAGE SCENARIO / USER INTERACTION.....	26
2.1 User profiles .....	26
2.2 Use-cases .....	26
2.2.1 Unavailability of Medicines:.....	27
2.2.2 Manufacturer-Distributer-Retailer: .....	28
2.2.3 Admin Panel:.....	29
CHAPTER 3 – FUNCTIONAL AND DATA DESCRIPTION .....	31
3.1 System Architecture .....	31
3.1.1 Architecture model.....	31
3.1.2 Subsystem/modules overview .....	34
3.1 Data Description.....	44
Major data objects .....	44
3.2.2 System level data model.....	51
3.2 System Interface Description .....	52
CHAPTER 4 - SUBSYSTEM/MODULE DESCRIPTION .....	63
4.1 Description of Authentication Module .....	64
4.1.1. Authentication Module Scope .....	65
4.1.2. Authentication Module Flow diagram.....	65

---

4.1.3 Authentication Module Components .....	66
4.1.4 Restrictions or Limitations .....	67
4.1.5. Design Constraints.....	67
4.2 Description of Client Creation Module .....	68
4.2.1 Client Creation Module Scope .....	68
4.2.2 Client Creation Module flow diagram .....	68
4.2.3 Restrictions Limitations .....	69
4.3 Description of Medicine Availability Module.....	69
4.3.1 Medicine Availability Module Scope .....	70
4.3.2 Medicine Availability Module flow diagram.....	70
4.3.3 Medicine Availability Module Components .....	71
4.3.4 Restrictions/Limitations .....	73
4.4 Description for Medicine Authenticity Module .....	73
4.4.1 Medicine Authenticity Module Scope.....	73
4.4.2 Medicine Authenticity Module flow diagram .....	74
4.4.3 Restrictions/Limitations .....	74
4.5 Description for Inventory Management Module .....	74
4.5.1 Inventory Management Module Scope .....	75
4.5.2 Inventory Management Module Flow Diagram.....	75
4.5.3 Restrictions/ Limitations .....	76
4.6 Description for Supply Chain Module.....	76
4.6.1 Supply Chain Module Scope.....	76
4.6.2 Supply Chain Flow Diagram.....	77
4.6.3 Supply Chain Subcomponents .....	77
4.6.4 Restrictions/ Limitations .....	80
4.7 Description for Manufacturer Module.....	80
4.7.1 Manufacturer Module Scope .....	80
4.7.2 Manufacturer Module Flow Diagram .....	81
4.7.3 Manufacturer Module sub-components .....	81
4.7.4 Restrictions/ Limitations .....	83
4.8 Description for Admin Module .....	84
4.8.1 Admin Module Scope.....	84
4.8.2 Admin Module Flow Diagram .....	84
4.8.3 Admin Module Subcomponents.....	84

CHAPTER 5 - BEHAVIORAL MODEL AND DESCRIPTION .....	89
5.1 Description for system behavior.....	89
5.1.1 Events/interrupts.....	90
5.1.2 States .....	92
5.3 Control specifications .....	101
5.3.1 Control specifications for End user:.....	101
5.3.2 Control specifications for Retailer: .....	101
5.3.3 Control specifications for Manufacturer: .....	101
5.3.4 Control specifications for Distributer: .....	102
5.3.5 Control specifications for Admin:.....	102
CHAPTER 6 - SYSTEM PROTOTYPE MODELLING .....	103
6.1 Description of System Modelling Approach .....	103
6.2 Prototyping Requirements .....	103
CHAPTER 7 - SYSTEM ESTIMATES AND ACTUAL OUTCOMES .....	115
7.1 System Estimates and Actual Outcomes .....	115
Historical Data used for estimates.....	115
Estimation based on COCOMO II .....	115
7.2 Actual Results and Deviation from Estimates .....	118
7.3 System Resources (Required and Used).....	119
7.3.1 Man-force Required and Used .....	119
7.3.2 Hardware Required and Used .....	120
7.3.3 Software/Tools Required and Used.....	120
CHAPTER 8 - TEST PLAN .....	121
8.1 System Test and Procedure.....	121
8.2 Testing Strategy .....	122
8.2.1 Unit Testing.....	122
8.2.2 Integration Testing .....	125
8.2.3 Validation Testing .....	126
8.2.4 High-order Testing (System Testing).....	126
8.3 Testing Resources and Staffing .....	128
8.4 Test Metrics .....	128
8.5 Testing Tools and Environment .....	129
CHAPTER 9 - FUTURE ENHANCEMENTS AND RECOMMENDATIONS .....	130
CHAPTER 10 - CONCLUSION .....	131

APPENDICES.....	132
-----------------	-----

## **LIST OF TABLES**

---

Table 01 – Values of COCOMO Coefficients	117
Table 02 – Actual and Estimated Results	119
Table 03 – Man-force Required	119
Table 04 – Hardware Required and Used	120
Table 05 – Software/Tools Required and Used	120

---

## **LIST OF FIGURES**

---

Figure 1.1 –Scope	18
Figure 1.2 – Pakistan Pharmaceutical Overview	19
Figure 1.3 – Pakistan Pharmaceutical Overview	19
Figure 1.4 – Crimes in Pharmaceutical Industry	20
Figure 1.5 – Crimes in Pharmaceutical Industry	21
Figure 2.1 – Use Case End User	27
Figure 2.2 – Use Case Distributer, Manufacturer, Retailer	28
Figure 2.3 – Use Case Retailer	29
Figure 2.4 – Use Case Admin	30
Figure 3.1 – System Architecture	32
Figure 3.2 – Presentation Layer - React	37
Figure 3.3 – ASP .NET Core	39
Figure 3.4 – Blockchain Module	42
Figure 3.5 – Blockchain Module	43
Figure 3.6 – Blockchain Module	43
Figure 3.7 – Blockchain Module	44
Figure 3.8 – DFD	51
Figure 3.9 – Search Medicine	52
Figure 3.10 – Results	53
Figure 3.11 – Pharmacies	54
Figure 3.12 – Order History	54
Figure 3.13– Add to Inventory	55
Figure 3.14– Add to Inventory	56
Figure 3.15 – Add to Inventory	56
Figure 3.16 – Create Purchase Order	57

Figure 3.17 – Create Purchase Order	57
Figure 3.18 – Warehouse	58
Figure 3.19 – Login	59
Figure 3.20 – Admin Dashboard	59
Figure 3.21 – Company Profile	60
Figure 3.22 – Company Documents	60
Figure 3.23 – Client Management	61
Figure 3.24 – Location Management	61
Figure 3.25 – Medicine Management	62
Figure 3.26 – Medicine Management	62
Figure 4.1 – Authentication Flow diagram	65
Figure 4.2 – Registration Flow diagram	66
Figure 4.3 – Login Flow diagram	67
Figure 4.4 – Client Creation Flow diagram	69
Figure 4.5 – Medicine Availability Flow diagram	70
Figure 4.6 – Search Flow diagram	71
Figure 4.7 – Reserve Order Flow diagram	72
Figure 4.8 – Medicine Authenticity Flow diagram	74
Figure 4.9 – Inventory Management Flow diagram	75
Figure 4.10 – Supply Chain Flow diagram	77
Figure 4.11 – Transfer of to Stocks Flow diagram	78
Figure 4.12 – Manage Warehouse Flow diagram	79
Figure 4.13 – Manufacturer Module Flow diagram	81
Figure 4.14 – Manage Distributer Flow diagram	82
Figure 4.15 – Manage Medicines Flow diagram	83
Figure 4.16 – Admin Module Flow diagram	84
Figure 4.17 – Manage Medicines Flow diagram	85
Figure 4.18 – Manage Users Flow diagram	86
Figure 4.19 – Manage Company Profile Flow diagram	87

Figure 4.20 – Manage Location Flow diagram	88
Figure 4.21 – Search and Order Reserved State diagram	94
Figure 5.1 – Inventory Updated State diagram	95
Figure 5.2 – Registered State diagram	95
Figure 5.3 – Authenticated State diagram	96
Figure 5.4 – Purchase/Sales order created State diagram	96
Figure 5.5 – Connected to distributor State diagram	97
Figure 5.6 – Stocks Transferred State diagram	97
Figure 5.7 – Warehouse Records Updated State diagram	98
Figure 5.8 – User Records Updated State diagram	98
Figure 5.9 – Location Updated State diagram	99
Figure 5.10 – Company Profile Updated State diagram	99
Figure 5.11 – Medicine Records Updated State diagram	100
Figure 6.1 – Search Results	104
Figure 6.2 – Reserve Order	104
Figure 6.3 – Track Reservations	105
Figure 6.4 – Order History	105
Figure 6.5 – Available Medicines	108
Figure 6.6 – Alternate Medicines	109
Figure 6.7 – Order Reserve	110
Figure 6.8 – Reservation Details	110
Figure 6.9 – Register	111
Figure 6.10 – Create Purchase Order	111
Figure 6.11 – Create Purchase Order	112
Figure 6.12 – Create Purchase Order	112
Figure 6.13 – Add Medicine	113
Figure 6.14 – Add Medicine	113
Figure 6.15 – Add Inventory Item	114
Figure 7.1 – COCOMO Estimation	117



# **CHAPTER 1 – INTRODUCTION**

---

Have you ever searched for a medicine but could not find it anywhere? Needed it quickly but had no idea which drugstore currently has it? Tired of inquiring at every other medical store with no success? Moreover, purchasing a medicine but unsure of its authenticity and facing problems due to counterfeit drugs? These are some of the questions that motivated us to propose a solution to the problem persistent in the pharmaceutical market.

## ***1.1 Goals and objectives***

Provision of medicines quickly and transparency in the pharmaceutical supply chain defines the objectives of our research project. It is commonly observed that the drug prescribed by your medical practitioner is not available at your known drugstores. Secondly, even if you are able to get it at a pharmacy, its authenticity is still in question as our industry is facing the common issue of medicine counterfeit. The Drug Regulatory of Pakistan was established in 2012 under the DRAP Act, 2012 after the reported incidents of numerous deaths caused by the substandard counterfeit drugs throughout the country. It formulated regulations circled around maintaining the quality and transparency in the pharmaceutical industry and curb malpractices. The problem is still persistent as conventional methods prove to be insufficient in combating the illegal trade. Therefore, we aim to eradicate the supply of counterfeit drugs with the game changing technology of blockchain that rightly serves the purpose of transparency, immutable tracking of assets and audit of drugs being manufactured and distributed.

## ***1.2 System statement of scope***

- Ensuring availability of required medicines:**

As our project title suggests, the system allows user to search for the required medicine and the system will provide a list of nearby pharmacies and medical stores on google map. Users can then reserve a medicine through our system or place an order to get it delivered at its doorstep.

- Authenticating the medicines' supply chain via blockchain infrastructure.**

Our system aims to allow pharmaceutical companies, distributors, and retailers to manage identities of medicines and recording transaction on blockchain network. A patient or a healthcare professional at a pharmacy or at a hospital can easily scan the QR Code on the package before the product is sold or

administered. A mobile application decodes the QR Code, creates a hash of the QR code content, and verifies the existence of that exact hash on the blockchain.

- **Inventory Management:**

Owing to the problem of fragmentation, these stores struggle to buy inventory and spend hours every week dealing with dozens of different suppliers, besides dealing with late, incorrect and uncertain deliveries. As a result, stores overpay for inventory, fail to get stock when they need it, and suffer lower incomes, while suppliers miss out on sales from eager customers.

Through our application, storeowners can order inventory whenever they want, receive on-demand delivery, enjoy transparent, competitive prices, and choose from the largest selection of products available

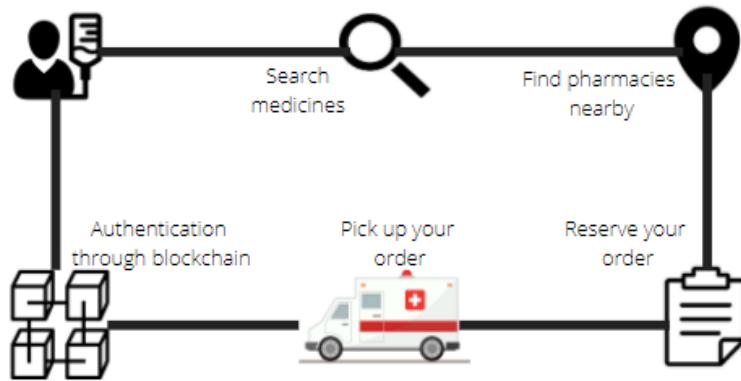


Figure 1.1- Scope

Lack of medicine availability and inefficient measures to counter counterfeit drugs provides us with an opportunity to tap this untouched market.

### **1.3 System context**

**Industry Type:** Pharmaceutical Industry

## Pakistan Pharmaceutical Overview

- Total Registered Pharmaceutical in Pakistan approximately **775**
- Multinational Pharmaceutical Company **26**
- National Pharmaceutical Company **749**
- Estimated size of the Pharmaceutical market is **Pkr 343 Billion** (IMS Q4, 2017)
- Projected Growth of Market is **11.42%** in 2018
- Molecules **2,200 plus**
- 1% of profit of All Registered Pharmaceutical to MOH for R&D Expenditure
- This again makes the country the region's 7th most attractive market in Asia.

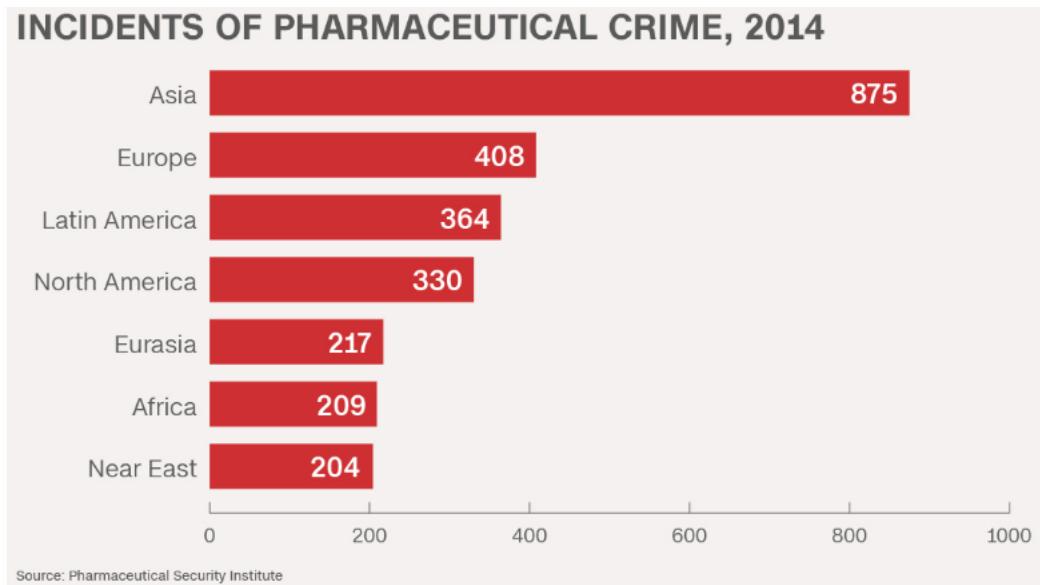
*Figure 1.2 - Pakistan Pharmaceutical Overview*

## Wholesale, Distributors & Pharmacies Universe

ENTIRE COUNTRY	
Total Wholesale	4,443
Total Distributors	1,045
Normal Pharmacies	47,139
Private Hospital Pharmacies & Vicinity Pharmacies	11,222
Total Pharmacies	58,541

*Figure 1.3 – Pakistan Pharmaceutical Overview*

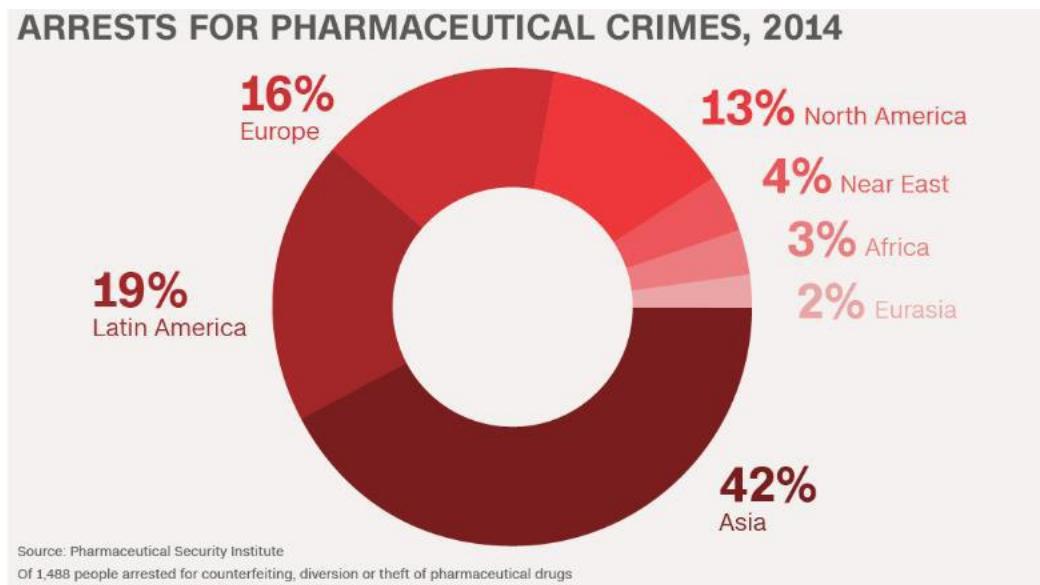
According to recent figures, Pakistan has a large pharmaceutical market and the above figures rightly reflects the potential in innovating the current obsolete methodologies adopted. By introducing a solution to provide an efficient access to the nearest drugstore will help the patients to better fulfill their immediate medical requirements.



*Figure 1.4 - Crimes in Pharmaceutical Industry*

According to statistics reported by Pharmaceutical Security Institute, the Asian region dominates the global reported crimes related to supply of counterfeit drugs. CNN published an article stating the current scenario of illegal drugs business in Pakistan. In an interview with a worker, he claimed to manufacturer all types of fake pills, syrups, injections etc. with similar ingredients and even using the same packaging. Therefore, it is difficult to tell whether the purchased medicine is a genuine drug belonging to the said company or is it a fake substandard drug.

Currently, conventional and centralized technological solutions are being used to provide prevention strategies for drug counterfeit. However, the methods can be easily tampered with and a malicious actor can manipulate the existing systems for personal gains. This is the reason why the problem still persists and no robust mechanism is currently employed to provide the customers with the ability to verify the drugs they are consuming.



*Figure 1.5 - Crimes in Pharmaceutical Industry*

**Blockchain: the solution for transparency in product supply chains:** We came across a white paper published by a UK based startup company by the name of Provenance that aims to solve the problem of transparency in the supply chain of different items through blockchain. This inspired us to adopt this strategy to implement transparency in medicine supply chain in order to ensure authenticity of medicines.

Our target market is categorized into two domains: the consumers and the businesses

**Consumers:** For our mobile app, we aim to target tech savvy and medium to high privileged classes aged between 18 to 45. For underprivileged and general mass, we plan to take on board clinics having small in-house pharmacies and attendants. They can assist the patients, having zero to little exposure to technology, by uploading their prescriptions on our web application and provide them with easy access to nearby drugstores.

**Businesses:** initially, we aim to target small and medium scale enterprises and retailers that have not yet integrated computerized systems in their pharmacies.

## ***1.4 Theoretical Background (of project)***

Being engineering students, we came up with the idea to bring revolution to the pharma industry with new emerging technologies. Our first ever pitch was at an event held at our university, DICE

VICE 2018. After getting meaningful feedback from this experience, we then further refined our idea into a business model. The initial efforts on our idea led us to pitch our proposed solution at multiple platforms including HULT Prize at NED, NIC Karachi and Gaditek, a well-known software house in Karachi. Upon receiving favorable reviews and support from the judges, we decided to take this idea further into the implementation stage.

We also developed a working prototype as part of our semester project and used it as a demo product for market feasibility study. We visited numerous drugstores, pharmacies and met pharmacists for their opinion. They appreciated our initiatives and efforts and suggested improvements on our product. This allowed us to incorporate the suggestions and improve our design and business logics.

Finally, with our tireless efforts, we finally got incubated at Technomites, a well-reputed software house in Karachi. We are currently working on the implementation under their supervision and plan to bring it in the industry very soon.

## ***1.5 Technology & Tools/hardware components (used in the Project)***

Throughout the design, implementation and testing phase of MediQuick application we have used variety of tools and technology that helped us produce a complete running reminder application.

### **1.5.1 ASP .NET Core (EF Core 3.0)**

.NET is open source, cross-platform framework that enabled us to achieve better performance, increase productivity and security and maintain docker support.

We used Entity Framework which is a modern object-database mapper for .NET for all database related activities like database connectivity and queries by the use of its support for LINQ and maintaining repositories of database tables instead of direct communication. It also helped us in bulk data uploading.

### **1.5.2 NEthereum**

An open source, cross-platform .NET integration library for blockchain. It enabled us to easily integrate Ethereum blockchain into our .NET application without having to change our chosen development environment to work with Ethereum.

### **1.5.3 Smart Contracts**

Smart Contracts are the agreement between two parties who are trading assets or performing transactions on the blockchain. It adds the business logic to the blockchain world. We used the solidity language to develop and implement our supply chain authentication logic using ERC721 Non Fungible Token standard.

### **1.5.4 React JS**

Is an open-source declarative, efficient, and flexible JavaScript (JS) library by Facebook which we used to develop the user interface (GUI front-end) of our application to give a professional look and great experience to our users. We integrated it with our .NET application to call APIs.

### **1.5.5 Visual Studio 2019**

It is a powerful IDE established by Microsoft that contains code editor (which enables accurate coding by providing assistance), packet manager, debugger, testing tools and much more. We used it to develop our ASP .Net Application and NEtherium blockchain modules.

### **1.5.6 Visual Studio Code**

Visual Studio Code is a powerful source code editor developed by Microsoft for different platforms that comes with helpful GUI, syntax highlighter, auto-indentation and much more. We used it to write an efficient and well-organized code for React JS.

### **1.5.7 SQL Server 2020**

SQL Server is a standard management tool for SQL Database by Microsoft. It enabled us to maintain database tables and retrieve required data easily and efficiently with reliability, better performance and increases scalability.

### **1.5.8 Postman**

Postman is a collaboration platform for API development. We used it to build REST API calls, constructing HTTP requests with data and reading responses in an organized way. API testing was also done using it.

### **1.5.9 Bitbucket**

Bitbucket is a version control and code collaboration repository hosting service(web-based) by Atlassian for code management of applications.

It made us use a Free unlimited private repository for our project so that we were able collaborate to share and merge code, test and deploy it at one place for .NET and React JS modules.

### **1.5.10 Infura Endpoint**

Infura is a public node that is connected to all the public test networks of Ethereum as well as the mainnet. We deployed our smart contract to Ropsten Test Network.

### **1.5.11 Geth Ethereum Client**

Geth is Go Ethereum's powerful and most popular CLI client used for running a node on the Ethereum network.

We used Geth Ethereum client to start local Blockchain as Geth creates a genesis block and mine our transactions

### **1.5.11 MS Project**

MS project as suggested by its name is a project management tool from Microsoft

We used it for distribution of work over timeline, to keep track of progress and achievement of milestones through Gantt Chart.

### **1.5.12 Google Sheet**

Google sheets is an online spreadsheet program that helped us to smartly keep track of our daily/weekly tasks and mark their statuses in a shared document/sheet.

### **1.5.13 StarUML**

StarUML is an open-source software by MK Labs for designing UML diagrams. We designed our use case, UML, state and data diagrams using it in an efficient, fast and flexible manner.

### **1.5.14 Draw.io**

Draw.io is a free online platform to create and edit diagrams of various categories with collaboration. We created organogram, work breakdown structure, architectural and other diagrams on it.

---

## **CHAPTER 2 – USAGE SCENARIO / USER INTERACTION**

---

Every product, software or a service is well-defined for a group of people or a community, the end users of MediQuick are defined below:

### ***2.1 User profiles***

There are five main users of our application; Admin, Pharmaceutical Company (Manufacturers), Distributors, Retailers and Consumers.

**2.1.1 End user/ Patients:** Patients or end users that are going to use our application to search medicines in nearby retail stores, place order or authenticate some medicine they have purchased.

**2.1.2 Retailers:** Medical store retailers who maintain their inventory on our system to get orders from the end users and place their orders to distributors.

**2.1.3 Distributors:** Medicine distributors who purchase medicines in bulk from manufacturing companies and sell them to retailers.

**2.1.4 Manufacturers:** Pharmaceutical companies who would keep a track of their supply chain on our system to ensure there is no breach in the chain through some counterfeit drug.

**2.1.5 Admin:** People of our company manage the users on our system.

### ***2.2 Use-cases***

All use-cases for the system are presented and descriptions of each use cases and diagram (if applicable).

## 2.2.1 Unavailability of Medicines:

Details:

**Actors:** End User / Patient and Retailer

Description:

End can search for medicines either by brand name or generic name. It can then check the availability of this medicine in nearby pharmacies or retail stores and get their information. User can further choose to view alternate/substitute medicines available.

User may also reserve an order for medicine and the respective Retailer will receive and fulfill the order.

User can also verify authenticity of medicine by scanning QR Code.

There is another feature for the user to view orders history as per the categories; fulfilled orders, orders that are still on pending or expired orders that have reached the reservation time limit and were never fulfilled.

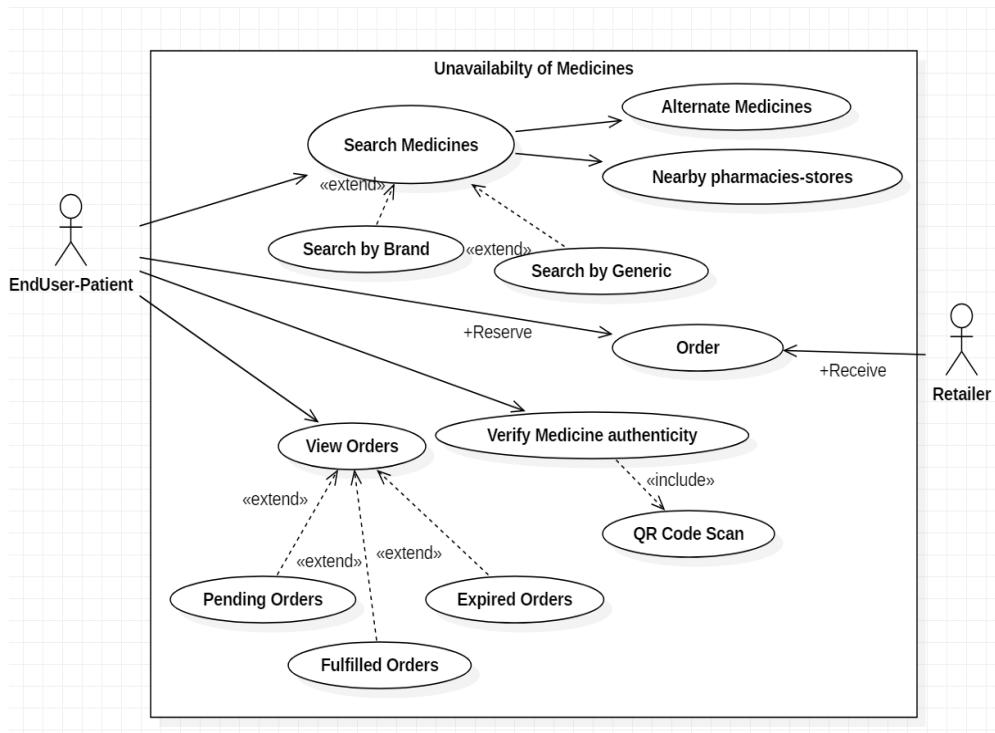


Figure 2.1 - Use Case End User

## 2.2.2 Manufacturer-Distributer-Retailer:

Details:

**Actors:** Manufacturer, Distributer and Retailer

Description:

Manufacturer can manage the medicines of their company (add, update or delete medicines).

Manufacturer can also manage the distributers to whom they supply medicines in bulk, if any of their distributer already exist on our system, they'd simply select them or else create new distributers.

Manufacturer's warehouse management can also be done on our system which enables creation of warehouses, creation of batches or uploading medicines in bulk. Any of these tasks performed would be punched to Blockchain.

Manufacturer can also transfer its stock to its warehouse or to any Distributer.

Distributers on the other hand, may place Purchase Orders to Manufacturer who would fulfill these and send Receive Orders to the Distributer and punch this action to Blockchain.

Similarly, a Retailer can also place Sales Orders to distributer who would send Receive Sales Order back to retailer with punch to Blockchain.

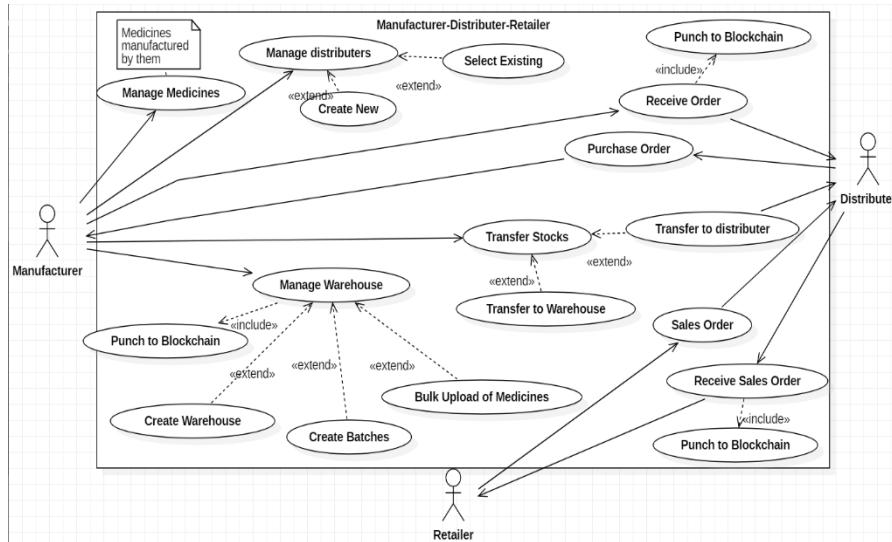


Figure 2.2 - Use Case Manufacturer, distributer, retailer

Retailer can manage their inventory on our systems and perform add, update and delete operations on the medicines.

Retailers can also view their orders history as per the categories; fulfilled orders, pending orders or expired orders that have reached the reservation time limit and were never fulfilled.

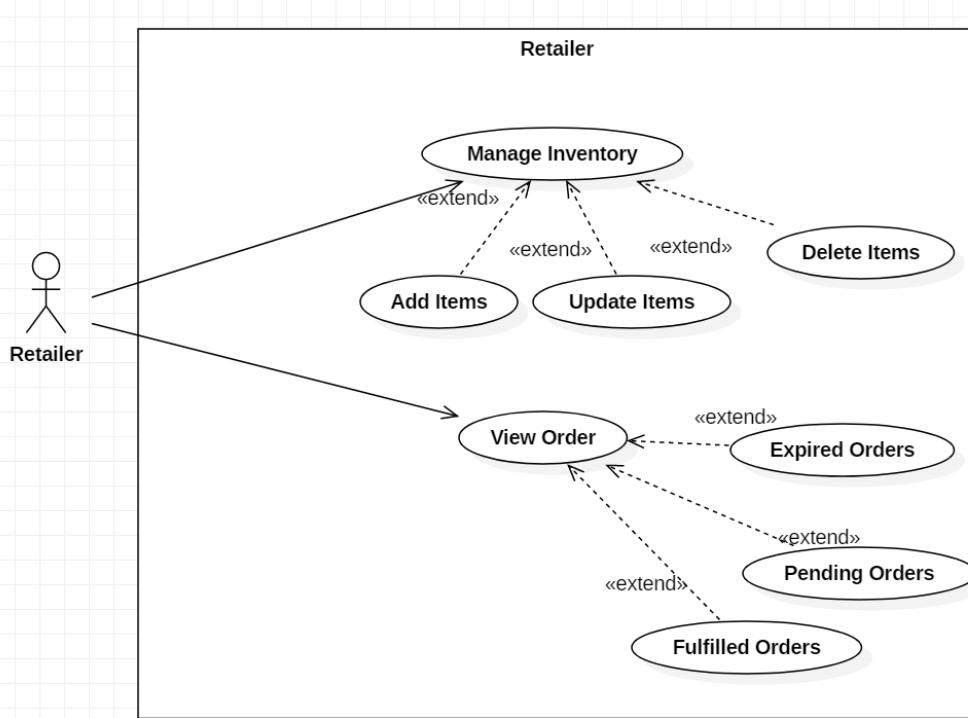


Figure 2.3 - Use Case Retailer

### 2.2.3 Admin Panel:

Details:

**Actor:** Admin

Description:

Admin can create accounts for the clients of three categories: distributors, retailers and manufacturers.

New locations can be created by Admin, he may add a new city, state or a country to our system.

Admin also holds right to manage medicines which means actions may be performed on brands, generics, dosage forms or variants of medicines

Admin user can manage our company Profile. It may further upload company's official documents.

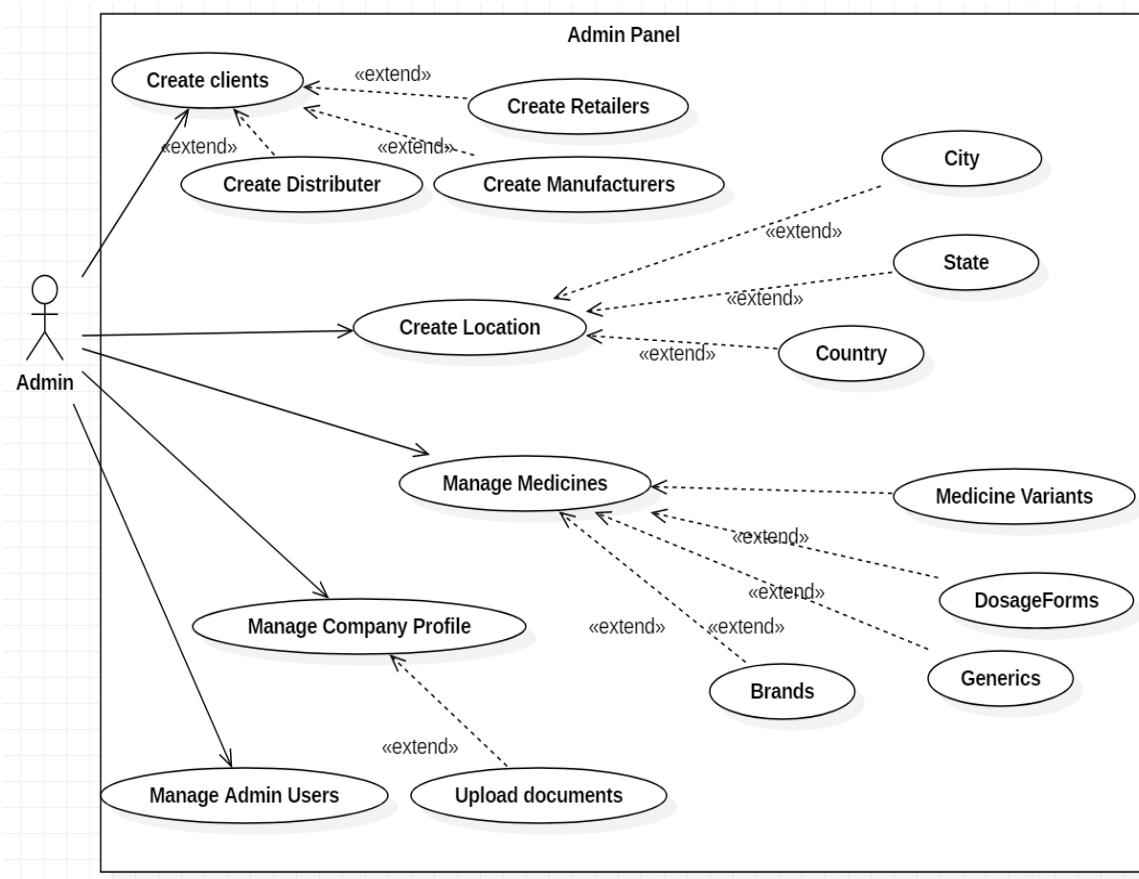


Figure 2.4 - Use Case Admin

# **CHAPTER 3 – FUNCTIONAL AND DATA DESCRIPTION**

---

This section provides an outline of the MEDIQUICK model, its integral parts and major data objects involved and their relationship that flows in the system. It gives us basic understanding about system architecture, how the data objects interact with the system and the overall workflow. The overall running process and the data and interface description is briefly described in this section below.

## ***3.1 System Architecture***

In order to build a scalable, efficient and maintainable software system, we have built a layered architecture for separation of concern, scalability and maintenance. Each layer is independent of each other as they are configured separately for each process. These independent layers can be replaced by multiple implementations if needed.

We incorporate **Single Responsibility** design principle in each of these layers to avoid tight coupling of modules and to support the concept of **Separation of Concern**.

### ***3.1.1 Architecture model***

Architecture of system is presented here. This includes detail description of system and system diagram

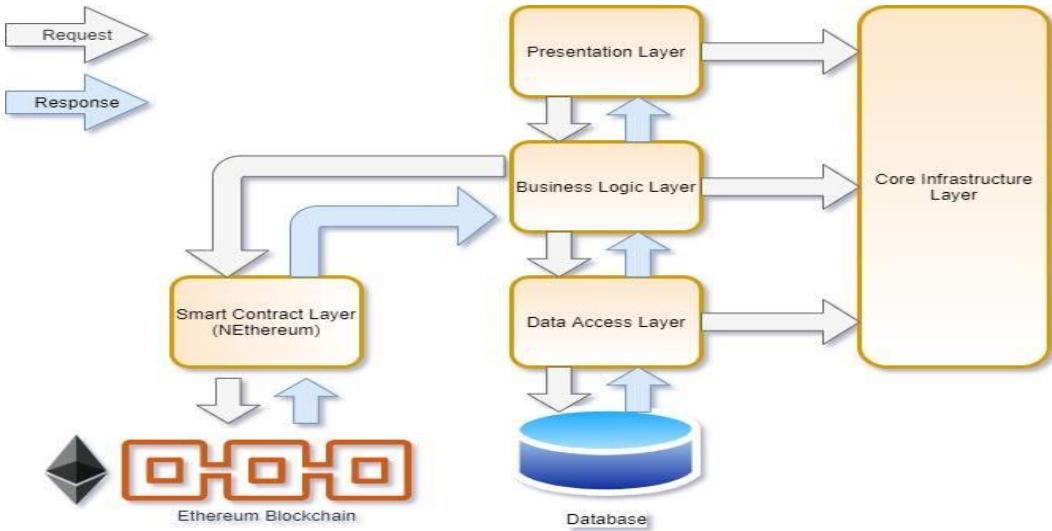


Figure 3.1 - System Architecture

### 3.1.1.1 Data Access Layer:

This layer handles **database interaction** of the application, and only location where database related queries are done. Here queries might vary according to underlying database, but these inner queries are not exposed. Functions related to CRUD are exposed publicly from this layer, where application can execute these methods. Then data access layer would connect to database, execute required query and return results to other layers, and thereby keeping other layers abstract from database integration. Typically data access layer is added as **repositories**.

### 3.1.1.2 Business Logic Layer:

This layer should handle **all domain specific logic** of the application, thereby complete logic is in a central location to be managed easily. Data access layer's atomic CRUD methods can be used to make **meaningful business scenarios**, and these business logic layer is typically added as **services**.

### 3.1.1.3 Presentation/UI Layer:

This layer is **entry point for external interaction** with the application. As an example it could either be REST endpoints if its ASP.NET Web API or HTML pages if its ASP.NET MVC application. Here it is expected to be without any business logic, but rather directly forward the request to business logic layer. Some simple operations such as **request validation** can be done here. Typically presentation layer is added as **controllers**.

### **3.1.1.4 Core Infrastructure Layer:**

Common entities for the application is stored in **models** and class library is created for these core changes. All **other layers will reference core layer** and import models for usage.

### **3.1.1.5 Smart Contract Layer:**

This is our blockchain layer where our .Net code interacts with the ethereum blockchain and calls smart contract functions deployed on a public network. Façade Pattern is used to wrap all the complex library code of Nethereum's Infrastructure. Two interfaces are defined here i.e. Common Smart Contract interface for deployment and receipt generation and ERC721 Non Fungible Token's Smart Contract interface.

### **3.1.2 Subsystem/modules overview**

#### **3.1.2.1 Data Access Layer:**

##### **3.1.2.1.1 Repository Pattern:**

Repositories are used to create **abstraction between database access and business logic**. Thereby usage of data, and access of data is separated where repository will manage interaction with database. This can be clearly seen in N-Tier architecture that is designed for separation of concern, and ASP.NET Core Web API can be also be modeled as N-Tier.

Repositories lie in the Data Access Layer of our n-tier architecture and separates Services/Business layer from database. Business logic cannot directly communicate with the database and execute queries.

In our application there are domain models such as Medicine, Brand and there will be methods associated to CRUD interaction related to each model. As per Medicine, methods in MedicineRepository would be as follows:

```
public Medicine getMedicineById(String authorId)  
public List< Medicine > getMedicine()  
public Medicine createMedicine(Medicine medicine)  
public Medicine editMedicine(Medicine medicine)  
public void DeleteMedicine(String medicineId)
```

##### **3.1.2.1.2 Why Generalize Repository Pattern?**

So as models increase, repository methods have to implemented in each repository. However most implementations will be similar and repetitive. As an example getMedicineById, getBrandById will have similiar database queries, and can be repetitive for all models and all methods

Thereby **generic approach** can be adapted to avoid these repetitions and avoid rework for any future changes. Thereby all models adapting this generic approach will inherit all these methods for usage. And new model will be able to inherit existing methods quickly. And if there is any database query change it can be done by few places.

### 3.1.2.1.2.1 Add Base Model

In order to make generic repository methods, models should have common properties such as **id**, **createdAt**, **updatedAt**.

As given by above repository methods, it can be seen that all models is required to have some generic properties such as id, createdAt, updatedAt. These common values can be added to **base model which will be extended by domain models**.

```
public class BaseEntity : IEntity
{
    public Guid Id { get; set; }
    public DateTime CreatedAt { get; set; } = DateTime.Now;
}

public interface IEntity
{
    public Guid Id { get; set; }
    public DateTime CreatedAt { get; set; }
}
```

### 3.1.2.1.2.2 Add Domain Models

After base model, separate classes should be created for domain models which will be used in application. Here Brand and Medicine models will be created in /Models which have to be **extended** from base model of BaseEntity.

### 3.1.2.1.2.3 Add Generic Repository

```
public interface IRepository<T> where T : BaseEntity
{
    IEnumerable<T> GetAll();
    T GetById(Guid id);
    void Insert(T entity);
    void Update(T entity);
    void Delete(Guid id);
}
```

### **3.1.2.1.3 Why Unit of Work?**

**Transaction support and connection management** is essential in an application architecture. Simply given business scenario should entirely succeed or entirely fail and not remain in partial condition.

Let's check mock scenario with domain models we have modeled before:

**BrandRepository.Insert(brand);**

**MedicineRepository.Insert(medicine);**

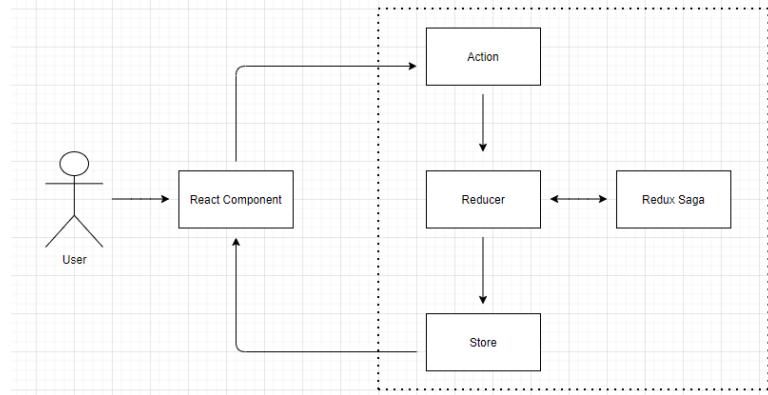
Here Brand is added and then **dependent** Medicine will be added. If the Brand was added so should its Medicine, and **shouldn't fail midway**. But from earlier, if we look into Repository implementation of write methods like Insert, Update or Delete it can be seen that context.SaveChanges() is executed at end. Here after each write database method, its committed to database. But what if application is somehow stuck after adding Book and failed to add Medicine, thereby a Brand will remain in database with a Medicine.

Thereby in implementation, any intermediate states should not be committed to database but rather all should be committed to database at end as an **Unit of Work**. This transaction support is achieved where commit or rollback are done at the very end and no immediate commit are done in-between. This pattern is called **Unit of Work** and should be **wrapped around repositories**.

**Unit of Work** responsibility can be summed as follows:

- Expose Repositories to outside. (Centralized database access)
- Commit or dispose changes to database. (Transaction management)
- Provide database context to Repositories. (Connection management)

### **3.1.2.2 Presentation Layer (React):**



*Figure 3.2 - Presentation Layer- React*

#### **3.1.2.2.1 React**

The main technology which we are using for the development of our application's front end is React. React is an open-source library based on java script and it is used for the development of interactive and responsive user interfaces. One of the reasons for using react is that it makes it very easy to design Progressive Web Applications (PWA's). PWA's combines the best of web and native applications. Using the latest web features, websites which are PWA's allows installation on a user's devices or computer, and provide a cross-platform application-like experience to the users. React provides the best options to build a responsive progressive web application which can be used cross-browser and is able to work offline using cache data from user's last interaction with the application.

#### **3.1.2.2.2 Redux**

Redux is a library or a pattern which is used for the management of the state of the applications. It provides a centralized data storing facility called a “store” which is used across the entire application, but this store or the data inside the store can only be modified or updated in a predictable and particular manner.

We decided to use redux along side react for the development of our front-end architecture is because react manages a component-level state. It means that whenever data is required to be passed from one component to another, it is done using props which makes it quite complex to manage the data in the application because our application has a large amount of data which is needed at different places in the application at the same time and this data is being constantly

updated and modified by different parts of the applications. So, in order to avoid data inconsistency, we decided to use redux along-side react for data management.

The main three components of Redux are explained as follows:

#### **3.1.2.2.1 Action:**

Actions are plain objects which can change the state in the store. Actions are the only way to send data into the store. Action describes what happened, but not how it happened or how the state of the application changed. Action calls are made from react components.

#### **3.1.2.2.2 Reducer:**

Reducers in redux are used to set out that how the state of the application is changed when an action is dispatched to the store. Actions in turn call the reducers using action types which update the state of the applications.

#### **3.1.2.2.3 Store:**

A redux store is where the whole state tree of our application is held and actions are the only method through which the state inside the store can be changed. Whenever the state of application is updated, the store updates itself.

### **3.1.2.2.3 Redux-Saga**

In order to carry out the side-effects in our application, we have used a library called redux-saga. Side effects includes asynchronous features like fetching of data from the database and accessing of the browser's cache. Redux-saga makes it easier to manage data and is better at handling failures. The CRUD operations are carried out using redux-saga in our application.

### **3.1.2.2.4 React-Redux**

As we are using react and redux together, so react-redux is a library which binds redux with our UI library, react. It is necessary to bind react with redux in order to interact with the store from our react components.

The react-redux library helped us a lot in enhancing our architecture. It enabled us to split our components into types. The container components are responsible for collecting and managing data where as the presentational components are used to display the data they receive. React-redux provides a connect function which generates a wrapper component call "container" and this container a responsible for all the interactions with the store.

React-Redux plays a great role in optimizing the performance of our application through multiple means. It controls the unnecessary re-rendering of the components and we can also ensure that

each component gets only specific data, which is actually required by the component instead of fetching the whole store.

### 3.1.2.2.5 Role-based Access Control

Role based access controls refers to the restricting the access to the system to some authorized users. It is an access-control mechanism which is built around privileges and roles.

As our application has multiple users including the super admin, manufacturers, retailers and the end-users so implementation of role-based access control was a must. The users in our application can only access features according to their roles and privileges assigned by us. Each user only sees or access features, which have been designed for them and has no control or access over features which do not concern them. We have implemented role-based access control using ReactJs.

### 3.1.2.3 ASP .NET Core – N-tier Architecture

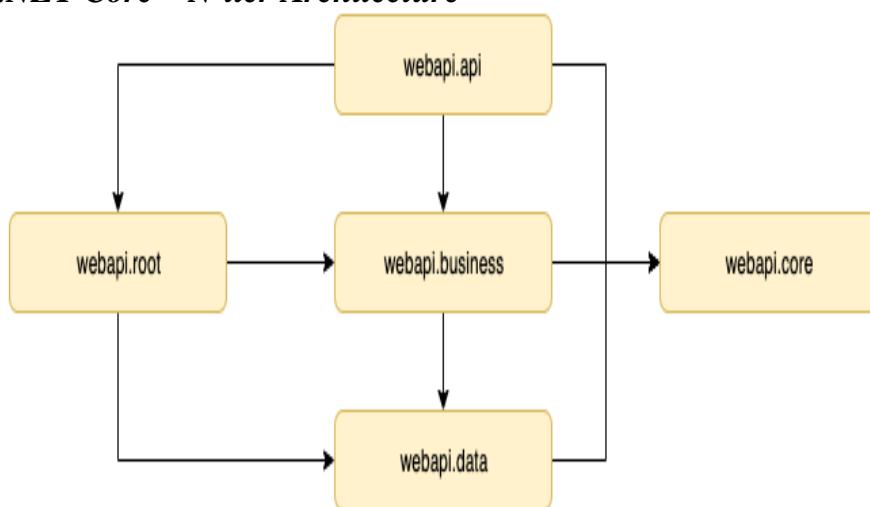


Figure 3.3 - Asp .NET Core

Our n-tier architecture consists of layers with single responsibility to provide separation of concerns logically and physically both.

Following is the description of each layer:

#### 3.1.2.3.1 Webapi.data – Data Access Layer:

It acts as a data access layer to communicate with the underlying SQL database, SQL queries. No other layer can directly connect to the database and unaware of its integration, only this layer would connect to database to execute query required by other layers and return the results. This layer consists of Repositories with functions related to CRUD , Unit of Work and DatabaseContext to handle the repositories against each database table communicate to these repositories. Database migrations are also created here.

### **3.1.2.3.2 Webapi.business - Business Logic Layer:**

This layer implements our domain specific business logic by implementing CRUD methods to create business scenarios as Services that make use of Repositories and unit of work placed in webapi.data.

### **3.1.2.3.3 Webapi.api- Presentation Layer:**

This acts a point of external interaction for REST APIs. It does not implement any business logic but forward requests to our business logic layer. IT receives api requests and validate them before forwarding to Services in webapi.business layer. It is implemented through Controllers.

### **3.1.2.3.4 Webapi.core- Infrastructure Layer:**

In ASP .NET Core entities are created as models, they might be placed in data access layer in a typical 3 tier architecture but here it is not appropriate to because models are used in Controllers of presentation layer for validations or as return type of controller methods. This would allow the presentation layer to directly communicate to the database layer which is against our design and not a good practice. Therefore, another infrastructure layer called webapi.core is added where the Models and ViewModels are kept, all other layers would communicate to this layer to reference and use entities (Models).

### **3.1.2.3.5 Webapi.root – Dependency Injection Layer:**

Dependency occurs between th layers like business logic layer depends upon on the data access layer's Unit of work to perform CRUD operations using its repositories. In ASP .NET Core ,

Dependency injections are injected to avoid object creation everytime (which leads to extreme memory consumption) and create objects one time to pass as parameters in constructors of dependent ones.

This is done in Startup file in presentation layer in a typical architecture. However, in n-tier where we would not want to violate data flow (webapi.api referencing webapi.data), it might sound possible to place these in webapi.core but this is not suitable either.

All other layers are already referencing Webapi.core, if injections are placed in webapi.core it would need it to reference back those layers to resolve dependencies creating a circular dependency. All the layers would be stuck in a loop, unable to compile. Hence, a new Dependency Injection layer called webapi.root is added to resolve all the dependencies (of Services, Repositories and DatabaseContext) at one place.

As stated above, dependency injections are placed in Startup file of presentation layer, therefore it keeps a reference to webapi.root to initiate the process of resolving dependencies.

### **3.1.2.4 Smart Contract Layer**

#### **3.1.2.4.1 Blockchain Sub Module:**

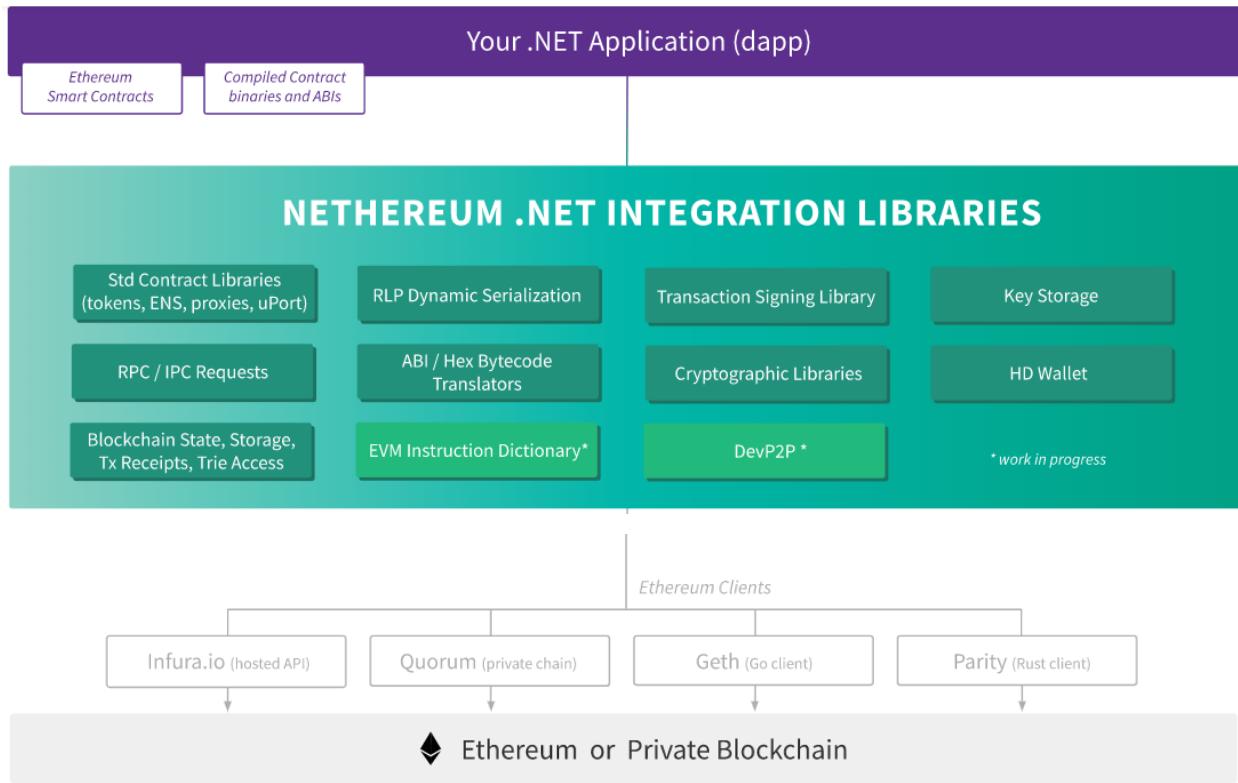
Our blockchain module is divided into two interfaces described below. Both define the function calls required to perform the transactions on ethereum blockchain. Nethereum provides a set of integrated libraries like Web3, Accounts, ManagedAccounts, ABI Translators, RPC Requests, Key Stores and HD Wallets.

##### **3.1.2.4.1.1 Common Interface:**

Contains the contract operations for common deployment function and contract information for receiving the ABI, Bytecode and Transaction Receipts

##### **3.1.2.4.1.2 ERC721 Interface:**

This interface defines our contract function calls like Mint, Transfer, GetToken and GenericTransactions etc,



*Figure 3.4 - Blockchain Module*

When a request is received by our contract code, the application fetches the deployed contract and its ABI (Application Bytecode Interface) from the given ethereum address deployed on a public blockchain. It estimates the gas used and signs the transaction offline using the **private key** of the sender.

The code then asynchronously calls the Nethreum Transaction function and the transaction request is forwarded to public ethereum network. Once it is mined successfully, the transaction result is returned and the transaction hash and block hash is received.

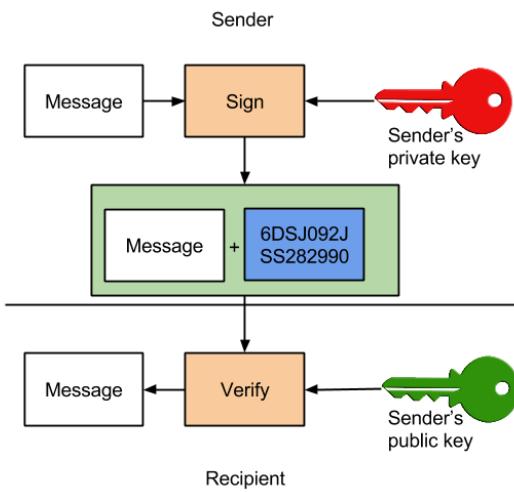


Figure 3.5- Blockchain Module

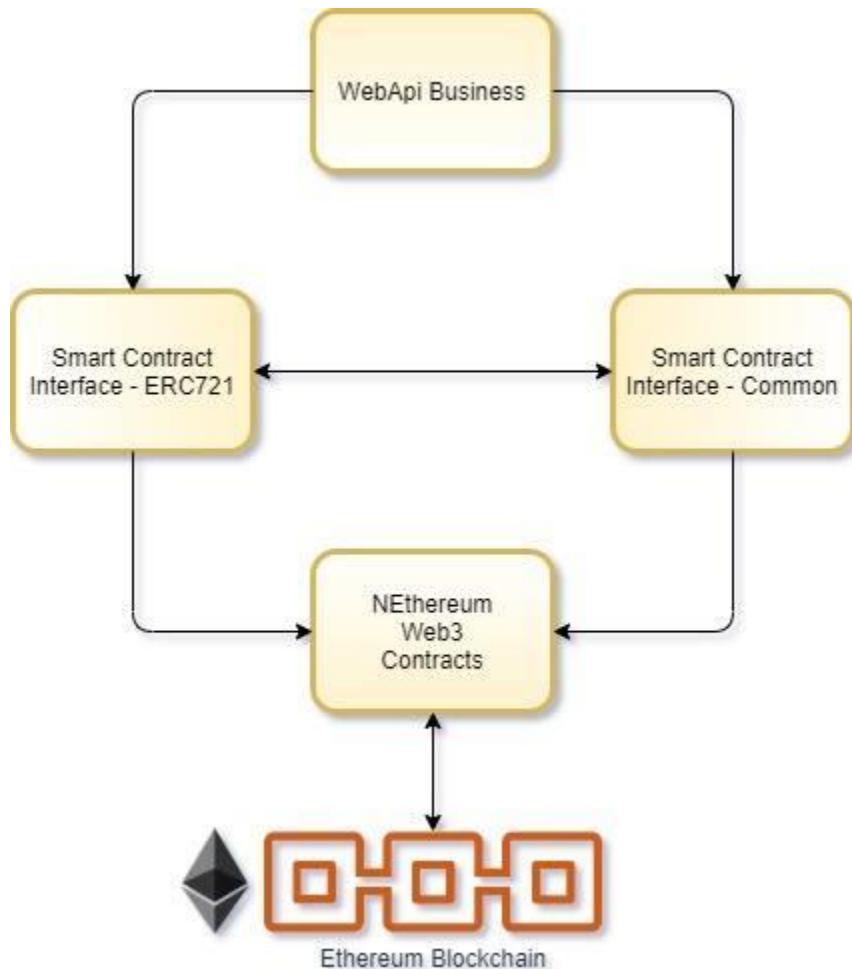
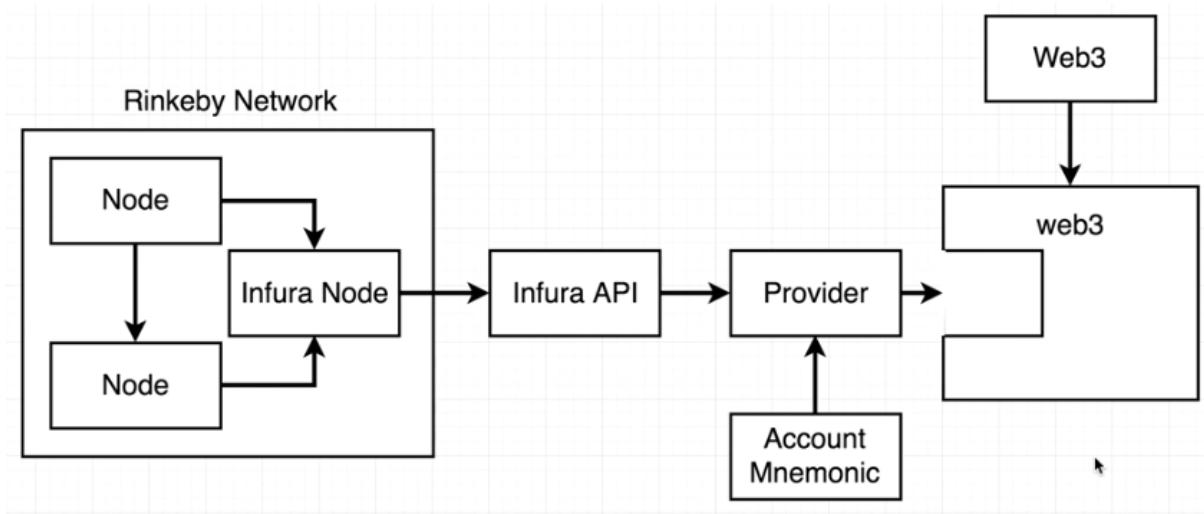


Figure 3.6- Blockchain Module



*Figure 3.7- Blockchain Module*

The above diagram illustrates the process of deploying our smart contract on the Rinkeby or Ropsten Test Network. Infura is a freemium tool to manage our blockchain deployments on their node instead of building the blockchain architecture from scratch on our computers which would take days and are highly complex. Infura allows us to create a project on their website and use their generated API endpoint to connect to ethereum network. In order to communicate with the node, we need a Web3 provider with our Ethereum account injected in it. This is how we can communicate using our authenticated blockchain mechanism.

## 3.1 Data Description

Top-level data objects that will be managed/manipulated by the MEDIQUICK application are defined below:

### 3.1.1 Major data objects

Data objects and their major attributes in MEDIQUICK application are described in this section:

MEDICINE:

{

```
"BrandId": 1,  
"DosageFormId": 1,  
"ManufacturerId": 1,  
"Packing": "500s",  
"GTIN":"put30" ,  
"CreatedAt": "2020-08-25 01:46:12",  
“Code” : “EJFK678”  
Generics: [{"GenericId": 1, "Strength": "500mg"}, {"GenericId": 5, "Strength": "500mg"}  
]  
}
```

#### RETAILER/ DISTRIBUTER

```
{  
"Name": "M Ali",  
"Email": " m.ali1224@gmail.com",  
"Password": "123456&ab"  
"Contact": "03113458594",  
“Address”: “ABC Road”,  
“CityId” : 1,  
“LicenseExpiryDate” : “2020-08-25”,  
“LicenseNo” : 533747384567456,  
“TRN” : 533747384567456,  
"CreatedAt": "2020-08-25 01:46:12"  
}
```

#### MANUFACTURER

```
{  
"Name": "M Ali",  
"Email": " m.ali1224@gmail.com",
```

```
"Password": "123456&ab"  
"Contact": "03113458594",  
"Address": "ABC Road",  
"LicenseExpiryDate": "2020-08-25",  
"LicenseNo": 533747384567456,  
"TRN": 533747384567456,  
"Code": 12345678  
"CreatedAt": "2020-08-25 01:46:12"  
}
```

## ROLE

```
{  
    "Name": "Admin",  
    "CreatedAt": "2020-08-25 01:46:12"  
}
```

## ADMIN USERS

```
{  
    "Name": "M Ali",  
    "Password": "123456&ab",  
    "RoleId": 1,  
    "CreatedAt": "2020-08-25 01:46:12"  
}
```

## CITY

```
{  
    "Name": "Karachi",  
    "StateId": 1,  
    "CreatedAt": "2020-08-25 01:46:12"  
}
```

```
STATE
{
  "Name" : "Sindh",
  "CountryId" : 1,
  "CreatedAt": "2020-08-25 01:46:12"
}
```

```
COUNTRY
{
  "Name" : "Karachi",
  "CreatedAt": "2020-08-25 01:46:12"
}
```

```
INVENTORY
{
  "MedicineId" : 1,
  "RetailerId" : 2,
  "Quantity" : 50,
  "IsDependable" : TRUE
  "ReservedQuantity": 20,
  "OnCallQuantity" : 10,
  "CreatedAt": "2020-08-25 01:46:12"
}
```

```
PURCHASE ORDER
{
  "ToId" : 2,
  "FromId" : 3,
  "CreatedBy" : 4,
  "Date" : "2020-08-25",
  "ShipDate": ""2020-08-25",
```

```
“StatusId” : 3,  
"CreatedAt": "2020-08-25 01:46:12",  
PurchaseOrderDetails: [{"MedicineId" : 70,  
“OrderedQuantity” : 1,} ]  
}
```

## PURCHASE ORDER RECEIVING

```
{  
“PurchaseOrderId” : 1,  
“ExpiryDate”: “2020-08-25”,  
“BatchCode” : “203638”,  
“ProductCode” : “AJFEK475”,  
"CreatedAt": "2020-08-25 01:46:12"  
}
```

## RESERVE ORDER

```
{  
InventoryId" : 70,  
“OrderDate” : “2020-08-25”,  
“DeliveryDate”: “2020-08-25”,  
“StatusId” : 3,  
“CreatedAt” : “2020-08-25 01:46:12”,  
ReserveOrderDetails: [ {  
“Quantity” : 1,  
"CreatedAt": "2020-08-25 01:46:12" } ]  
}
```

## STATUS

```
{
```

```
    "Name" : "Pending",
    "CreatedAt" : "2020-08-25 01:46:12"
}
```

BATCH:

```
{
    "BatchNo": "83418t",
    "MedicineId": 1,
    "ExpiryDate" : "07-12-20",
    "ManufacturingDate" : "07-08-20",
    "ManufacturerId" : 2,
    "BatchDetails": [{"SerialNumber" : 7000}, {"SerialNumber" : 6030} ]
}
```

WAREHOUSE:

```
{
    "ManufacturerId": 2,
    "DistributerId": 6 ,
    "Address": "xyzroad",
    "CityId": 1,
    "StateId" : 1,
    "CountryId" : 2,
    "Latitude" : "23",
    "Longitude" : "34"
}
```

WAREHOUSEMEDICINES

```
{
    "BatchId" : 1,
```

```
“WarehouseId” : 2,  
“Quantity” : 500  
}
```

### 3.2.2 System level data model

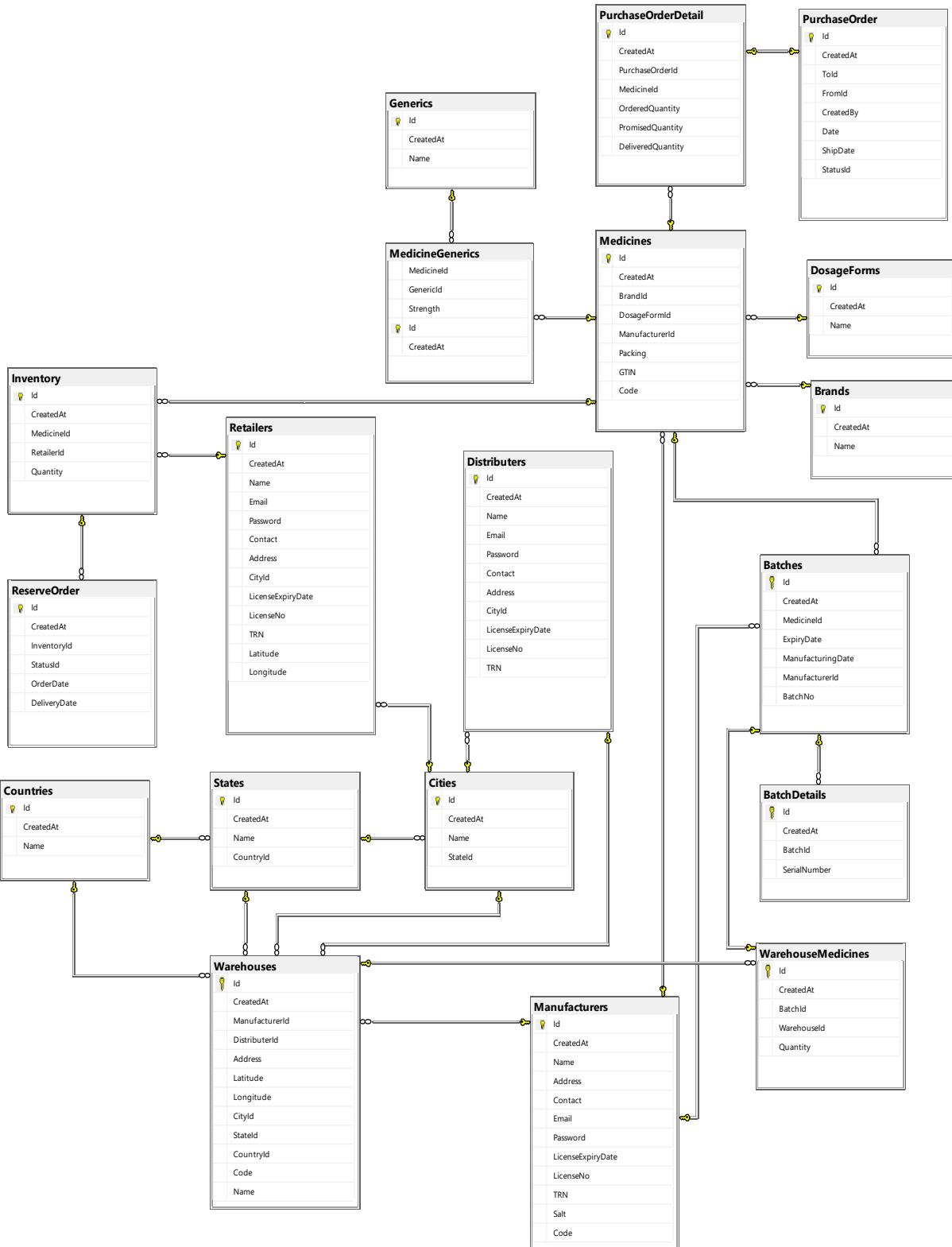


Figure 3.8 - DFD

## 3.2 System Interface Description

The System's interface(s) to the outside world are described.

The System's interface is how the end users and other profiles of our system perceive MediQuick application. It is a user-friendly web application that can also be downloaded as PWA (Progressive Web App) in mobile phones.

Interface would be different for different user profiles or categories which is as follows:

### 3.2.1 For End-User/Patient:

User are redirected to home page, they may or may not SIGNIN/SIGNUP for searching the medicines. They may either search by brand or generic name. The search bar will autocomplete the medicine name as only system recognized medicines can be searched

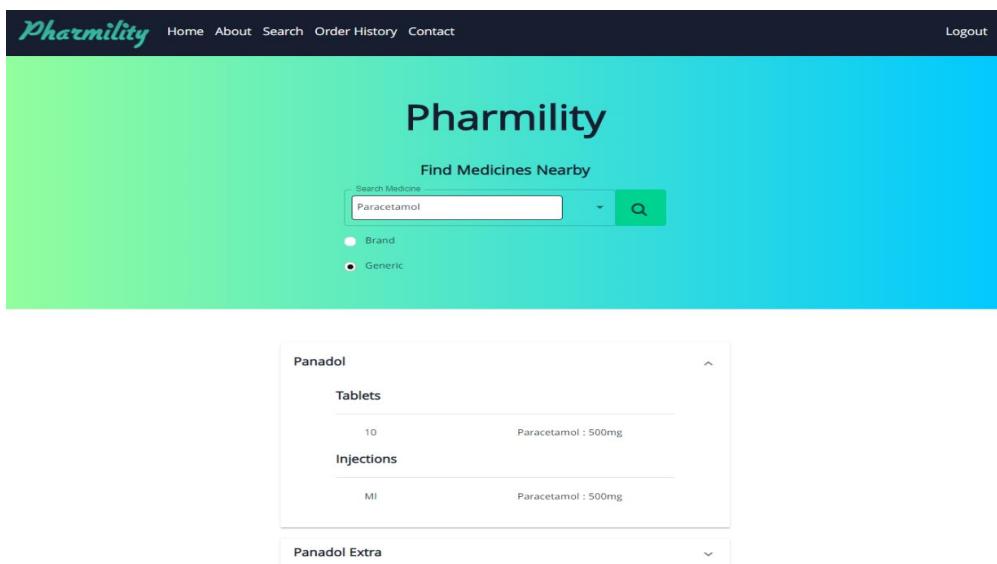


Figure 3.9 – Search Medicine

Search results are shown with medicine availability in nearby stores and an option to view alternate medicines which on click shows substitute medicines with their drug composition. Any of the displayed nearby stores information can be expanded by the user to select a particular store and view the location/route on map.

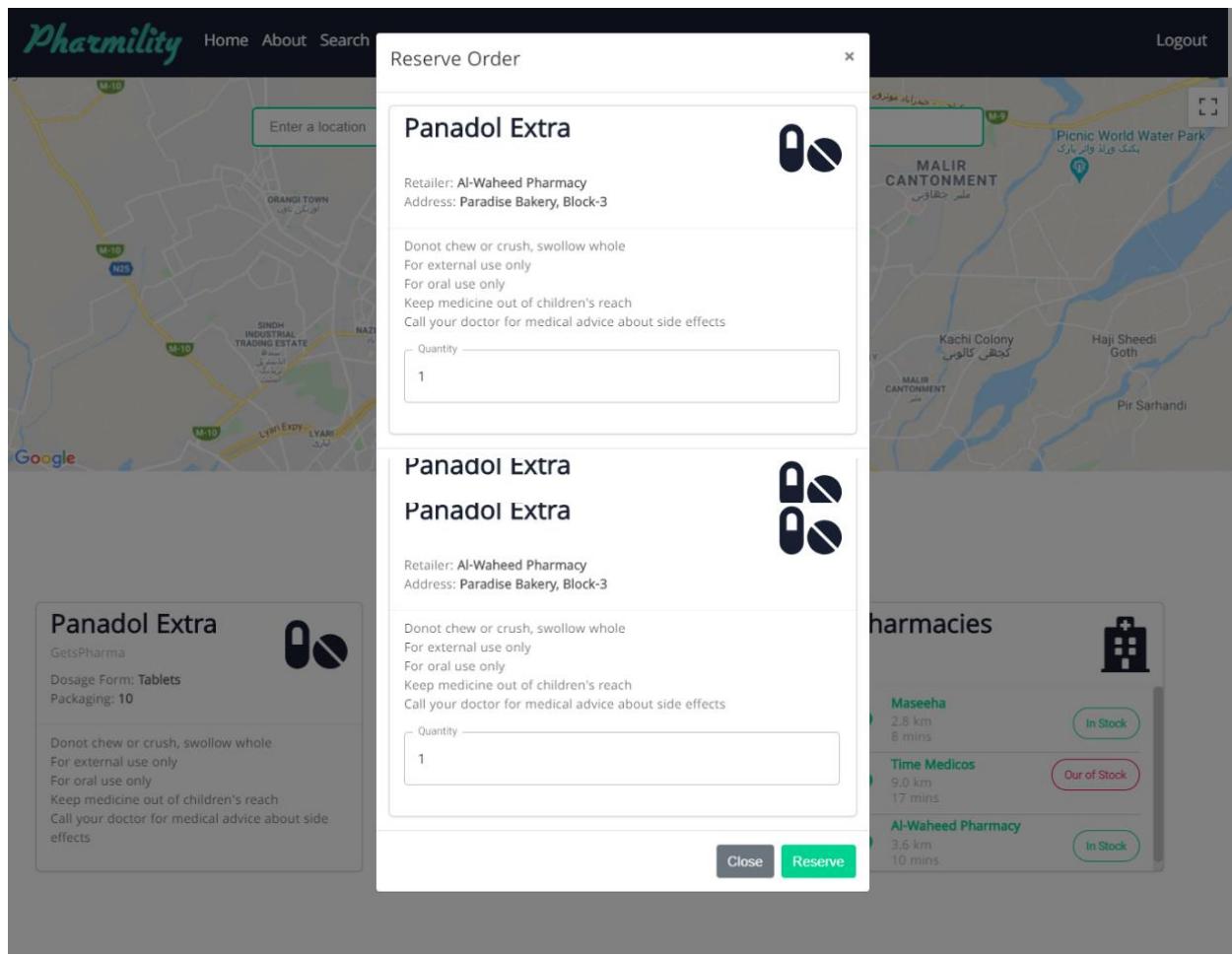


Figure 3.10- Results

If user now chooses to reserve the order at the selected store, SIGNUP/LOGIN modal is opened before proceeding further. Signup form contains fields for email, password, contact and other essential information whereas email and password will be required to Login.

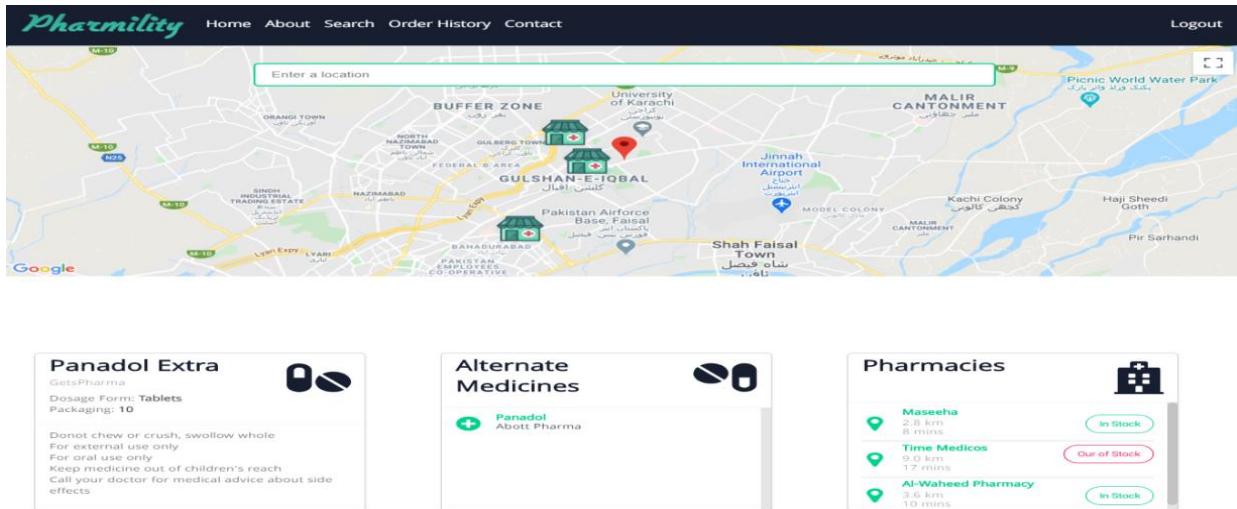


Figure 3.11- Pharmacies

After signing in successfully, order is reserved with confirmation toast message display.

In normal scenario of LOGIN/SIGNUP, user lands on the homepage where it can view the order history (pending/fulfilled/expired)

Orders	
	Time Medicos Stadium Road 2020-09-20T00:00:00 Panadol, Tablets <span style="float: right;">Qty: 10 Pending</span>
	Maseeha Block-7, Gulshan-e-Iqbal 0001-01-01T00:00:00 Panadol, tablets <span style="float: right;">Qty: 5 Pending</span>
	Medic Block-7, Gulshan-e-Iqbal 2020-09-20T00:00:00 Panadol, Tablets <span style="float: right;">Qty: 2 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-09-20T00:00:00 Buscopan, Tablets <span style="float: right;">Qty: 2 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-01T00:00:00 Myteka, Drops <span style="float: right;">Qty: 1 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 0 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 10 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 56 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 5 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 1 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 2 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR <span style="float: right;">Qty: 1 Pending</span>
	Maseeha Block-7, Gulshan-e-Iqbal 2020-10-04T00:00:00 Rigix, Tabs DS <span style="float: right;">Qty: 2 Pending</span>
	Al-Waheed Pharmacy Paradise bakery, Block-3 2020-10-05T00:00:00 Buscopan, Tablets <span style="float: right;">Qty: 2 Pending</span>

Figure 3.12- Order history

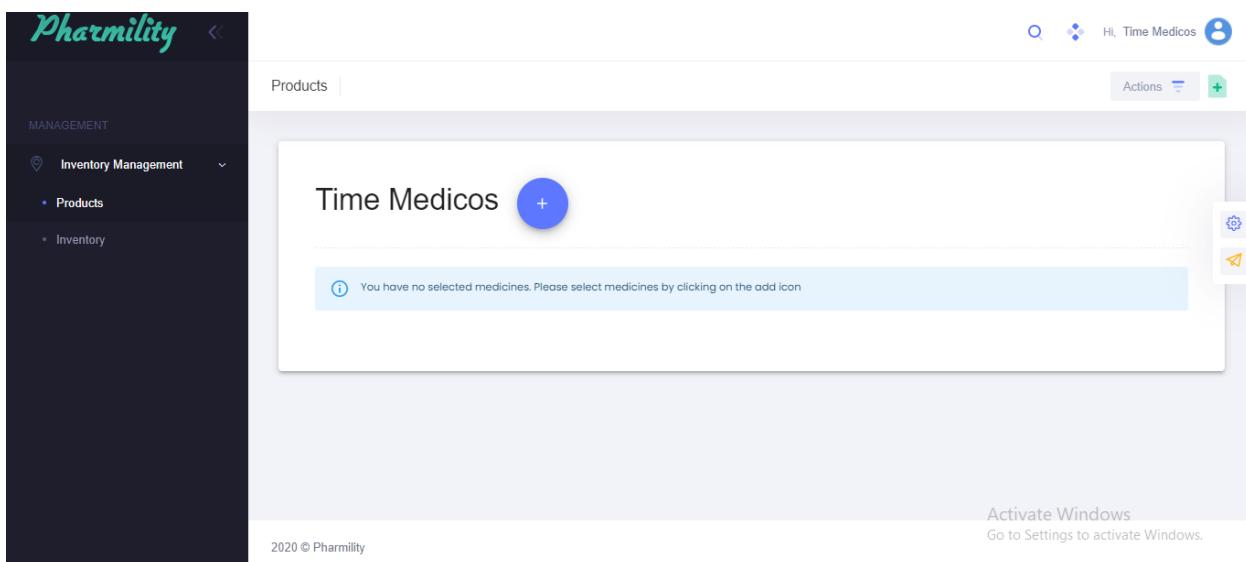
To verify authenticity of any medicine, user will be required to scan QR Code on the packaging using their mobile phone and the result will be displayed after processing it. Here, it is not required for the user to log in.

### **3.2.2 For Retailer/Distributer/Manufacturer:**

Retailer, distributer and manufacturer will be required to LOGIN with email and password provided by our organization at the time of contract.

#### **3.2.2.1 Retailer:**

Once logged-in, retailer will land on home page where its order history is displayed along with the notifications for any new orders. Retailer will be able to update the status by clicking on the particular orders except for expired ones.



*Figure 3.13- Add to Inventory*

Inventory Management module has the options to add, edit or delete medicines in the inventory. When adding any new medicine, retailer will only be allowed to select the medicine recognized by our system using auto-complete and enter the quantity manually. For already present medicines, he will be able to edit the quantity or delete it.

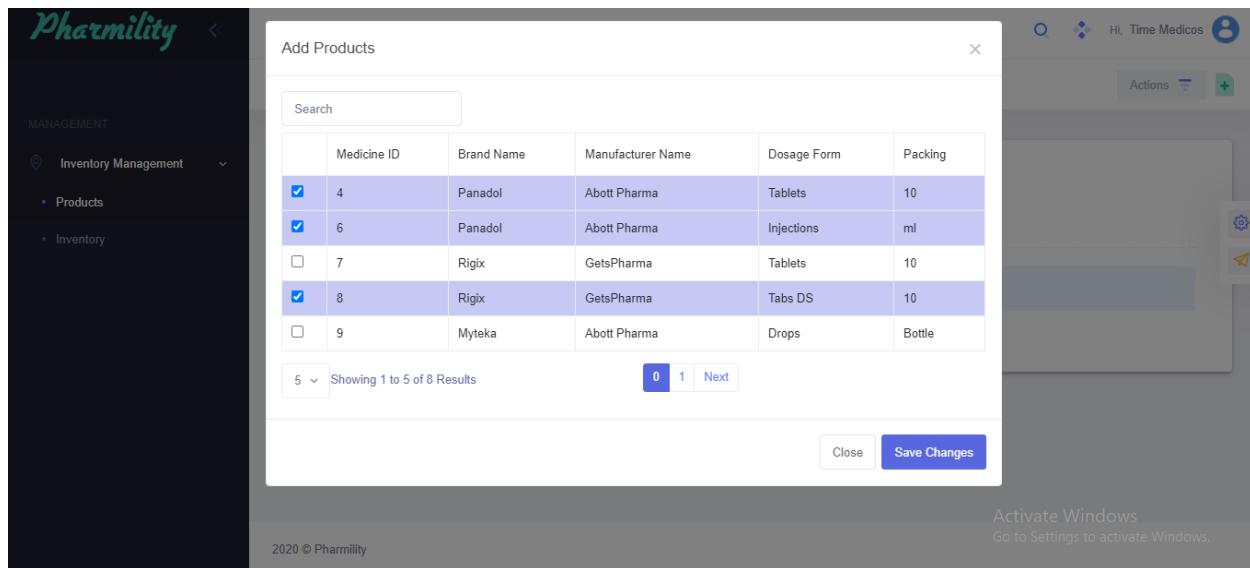


Figure 3.14- Add to Inventory

Retailer has option to place Sales orders to distributers. For placing the order, retailer will click on create new order. Form will open up to enter the details like required medicines and quantity and select the distributer name from autocomplete. Order placement confirmation will be shown.

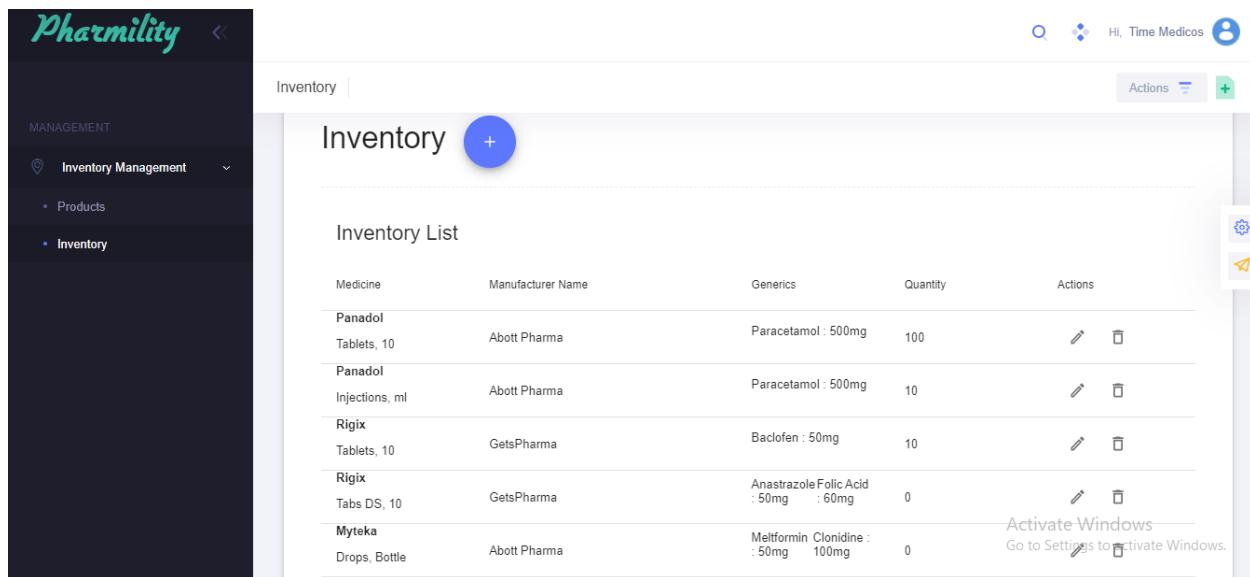


Figure 3.15- Inventory

### 3.2.2.2 Distributer:

Distributers will be notified on home page for the Sales orders placed by retailer for which he can update the status. Distributer can also view order history.

For placing the Purchase Order to manufacturers, distributor will click on create new order. Form will open up to enter the details like required medicines and quantity and select the manufacturer name from autocomplete. Order placement confirmation will be shown.

The screenshot shows the 'Create Purchase Order' screen. At the top, there are input fields for 'ABC' (likely medicine code) and 'Select Manufacturer'. Below these are two date pickers labeled 'Order Date' and 'Shipping Date', both set to '18/Sep/2020'. To the right, a status field shows 'Status Created'. A message at the bottom left says 'You have no selected medicines. Please select medicines by clicking on the add icon'. A blue '+' button is located above this message. At the bottom right is a 'Submit' button.

Figure 3.16 - Create Purchase Order

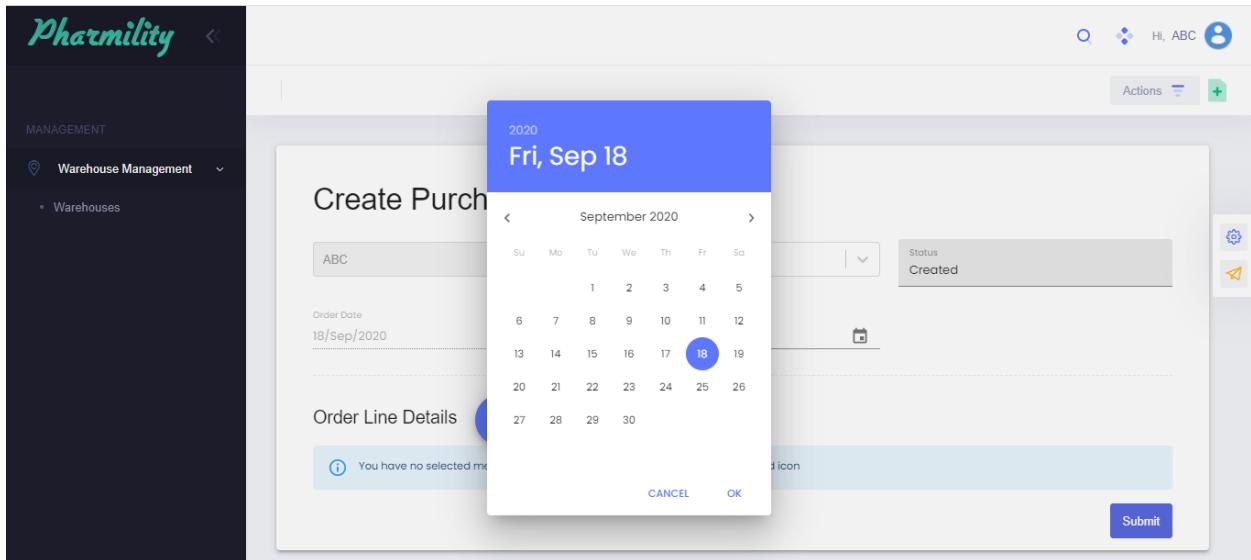


Figure 3.17- Create Purchase Order

### 3.2.2.3 Manufacturer:

Manufacturer will be notified on home page for the Purchase orders placed by distributor for which he can update the status and send back Purchase Order Receiving to fulfill the order. Orders history is also displayed.

To transfer stocks to warehouse or distributor, manufacturer will click transfer stocks button which asks to make a choice between warehouse and distributor and fill required details in the form.

Manufacturer can manage medicine but only medicines of his own company to perform edit, delete medicines or create new.

On clicking Manage Warehouse, user will be provided to select among Create warehouse, create batch or bulk upload of medicines. On making the choice, respective form will open up. For bulk upload, simply an excel file will be uploaded from browsing on device.

The screenshot shows the Pharmility software interface. On the left, there is a dark sidebar with the Pharmility logo at the top, followed by 'MANAGEMENT' and a 'Warehouse Management' dropdown menu with 'Warehouses' selected. The main area has a light gray header with 'Warehouses' and a search bar. Below the header is a large white box titled 'Warehouses' with a blue circular '+' button. Inside this box, there is a sub-header 'Warehouses' and a table with two rows of data. The table columns are: Warehouse, Code, Lat, Long, Address, City, State, Country, and Actions. The first row shows 'ABC Wareh...' with code 'ABCWH', coordinates '22.4, 67.5', address 'Gulshan', city 'Karachi', state 'Sindh', country 'Pakistan', and actions 'Edit' and 'Delete'. The second row shows 'ABC Wareh...' with code 'ABCWH2', coordinates '44.2, 65.4', address 'Highway', city 'Hyderabad', state 'Sindh', country 'Pakistan', and actions 'Edit' and 'Delete'. At the bottom of the main area, there is a footer with the text '2020 © Pharmility'.

Figure 3.3 – Warehouse

#### 3.2.2.4 Admin:

Admin user lands on Admin Dashboard after logging-in with admin id/password.

Admin manages the brands, generic names, dosage forms and medicine variants by clicking on buttons to perform actions of Add or Edit by filling out/updating form fields. On selecting the delete option, user is asked for confirmation of his choice. Confirmation message is displayed on successful completion of actions.

Similarly, admin can create new locations (cities, states and countries) in the system by clicking on the option.

When creating new client users, category is selected (distributer, manufacturer or retailer) and required information fields are filled in the form.

Admin will manage the people of the organization by adding or updating the granted access using clickable icons or even click on Add new admin users to fill the form and create new admin users.

Admin may also update information related to the company by editing the form fields. Company documents are also uploaded with description and file extension (document type) with clickable icons to edit the name, description, replace file or download it.

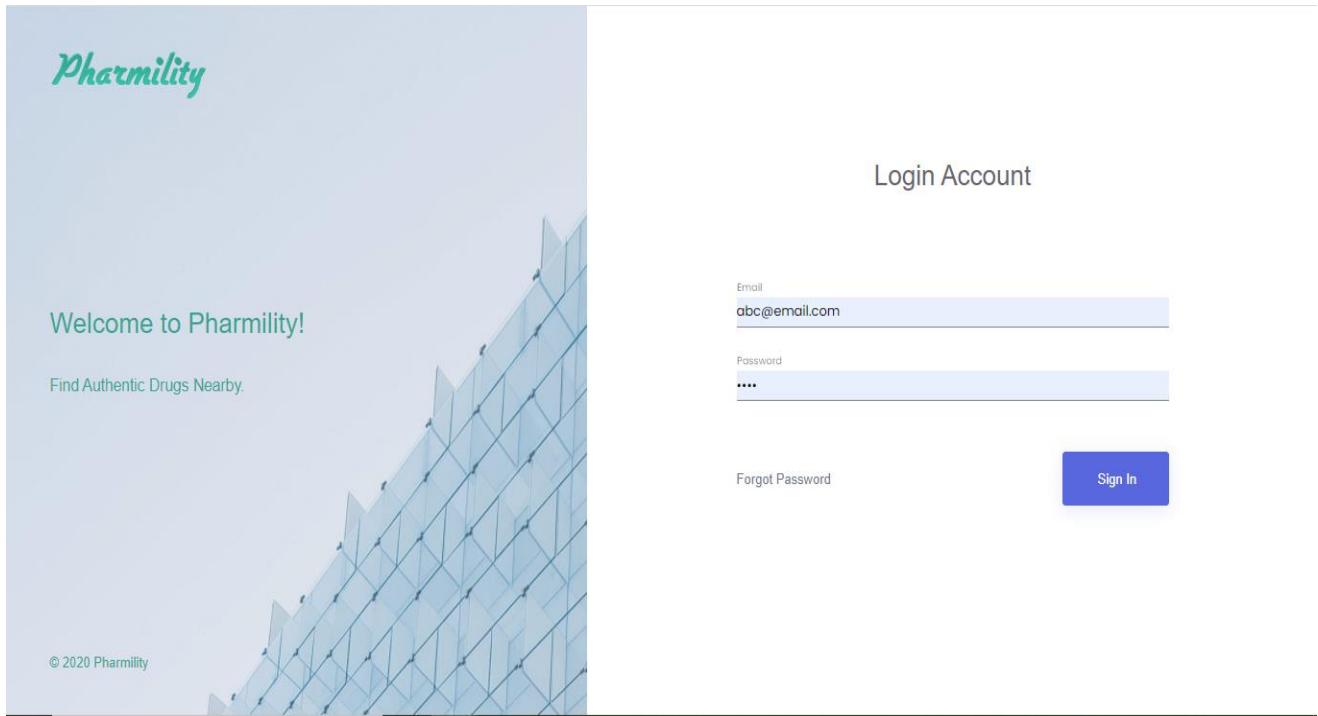


Figure 3.4 – Login

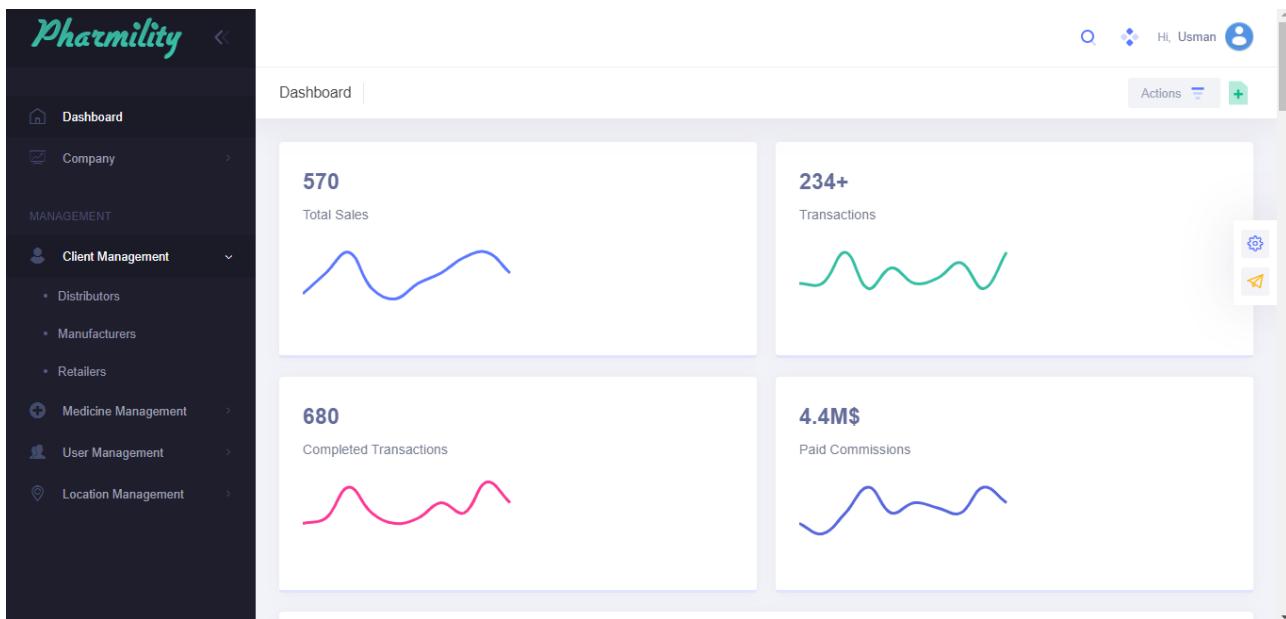


Figure 3.5 – Admin Dashboard

The screenshot shows the 'Company Profile' section of the Pharmility application. The left sidebar has a dark theme with categories like 'Dashboard', 'Company' (selected), 'Documents', 'Management' (Client Management, Medicine Management, User Management, Location Management), and 'Actions' (Search, Refresh, Hi Usman, +). The main content area has a light background with a header 'Company Profile' and the 'Pharmility' logo. It displays company details in four sections: 'Company Name' (Pharmility), 'Company TRN No' (123456464778975), 'Company License No' (I23456789012346), 'License Expiry' (25/Nov/2020), 'Company Address' (Karachi Pakistan), 'Company Contact 1' (03030388456), and 'Company Contact 2' (03348289867). On the right, there are icons for settings and sharing.

Figure 3.6- Company Profile

The screenshot shows the 'Company Documents' section of the Pharmility application. The left sidebar is identical to Figure 3.6. The main content area has a light background with a header 'Company Documents' and a blue '+' button. It displays a table of documents:

Name	File Description	Actions
BS EE Course	Dubai Uni	[Edit] [Delete] [PDF]
New Doc	sample	[Edit] [Delete] [PDF]
Sample Word File	hello world	[Edit] [Delete] [DOC]
sadas	asdasd	[Edit] [Delete] [PDF]

Figure 3.7 - Company Documents

The screenshot shows the Pharmility application's Client Management section. The left sidebar includes options like Dashboard, Company, MANAGEMENT (Client Management, Medicine Management, User Management, Location Management), and a search bar. The main area is titled 'Retailers' and displays a table with columns: Name, Email, Contact No., City Name, and Actions. One row is shown: Time Medicos, admin@timemedicos.com, 03333148282, Karachi, with edit and delete icons. A blue '+' button is at the top right of the list.

Name	Email	Contact No.	City Name	Actions
Time Medicos	admin@timemedicos.com	03333148282	Karachi	

Figure 3.8 - Client Management

The screenshot shows the Pharmility application's Location Management section. The left sidebar includes options like Dashboard, Company, MANAGEMENT (Client Management, Medicine Management, User Management, Location Management), and a search bar. The main area is titled 'Countries' and displays a table with columns: Country Name and Actions. Two entries are listed: India and Pakistan, each with edit and delete icons. A blue '+' button is at the top right of the list.

Country Name	Actions
India	
Pakistan	

Figure 3.9 - Location Management

The screenshot shows the Pharmility platform's Medicine Management section. On the left, a dark sidebar menu includes 'Dashboard', 'MANAGEMENT' (with 'Medicine Management' expanded), 'Warehouse Management', and other options like 'Brand', 'Manufacturer', 'Generics', and 'Dosage Form'. The main area is titled 'Medicines' and contains a table of two entries:

Created At	Brand Name	Dosage Name	Manufacturer	Drugs	Packing	Code	Actions	
2020-07-29...	Buscopan	Drops	Pfizer	Montelukast: 50mg	100ml	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	
2020-08-24...	Exlant	Tablets	Pfizer	Montelukast: 20mg	10s	EXL10	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Below the table are pagination controls: 'Rows per page: 10', '1-2 of 2', and navigation arrows.

Figure 3.10 - Medicine Management

The screenshot shows the 'Add Medicine' dialog box overlaid on the main interface. The dialog has fields for 'Brand' (Pfizer), 'Dosage Form' (Select Dosage Form), 'GTIN' (Text input), 'Code' (Text input), and a 'Drugs' section. The 'Drugs' section contains a 'Select Drug' dropdown, a 'Strength' input field, and a 'Remove' button. At the bottom of the dialog are 'Close' and 'Submit' buttons. In the background, the main 'Medicines' table is partially visible.

Figure 3.11 - Medicine Management

## **CHAPTER 4 - SUBSYSTEM/MODULE DESCRIPTION**

---

This section describes the main modules of our application, sub-modules and their components that are implemented in order to achieve required results. Our application MediQuick is comprised of the following subsystem and their components.

- Authentication Module
  - Login component
  - Signup component
- Client Creation Module
  - Manufacturer/Distributer/Retailer
- Medicine Availability Module
  - Search component
  - Reserve Order component
- Medicine Authenticity Module
- Inventory Management Module
- Supply Chain Module
  - Create Sales Order
  - Purchase Order
  - Transfer Stocks
  - Manage Warehouse
- Manufacturer Module
  - Manage Distributer
  - Manage Medicines
- Admin Module
  - Manage Medicines
  - Manage Users
  - Manage Company profile
  - Create Location

## **4.1 Description of Authentication Module**

### **JWT Authentication**

In our application, we have used JSON Web Tokens (JWT's) for authorization of users. JWT is an open standard (RFC 7519) that provides a secure method of transmitting data and information between parties as a JSON object. They are commonly used for authorization in applications.

In our application, when a user login's, each request made by the user will include a JWT in it's authorization attribute in the header of the request which allows the user to access routes and services that can be accessed with that token.

JSON Web Tokens consist of three parts:

- Header
- Payload
- Signature

#### **Header:**

Header consist of two parts: the type of the token i.e. JWT and the algorithm which is used to sign the token. In our application, we are using SHA256 encryption algorithm for signing the token.

#### **Payload:**

The second part of the token is the payload, which contains the claims. Claims are attributes of an entity (mostly, the user) and some additional data. In our application, we are forming the payload using multiple claims and most of them are unique for each user. We have used the user-id, username and the type of the user which is trying to login to the system, as claims. If the password and username entered by the user are correct and the user exists in our database, then a JWT token is returned to our front-end application for authorization.

#### **Signature:**

The last part of the token consists of a signed combination of the encrypted header, encrypted payload, a secret key and an algorithm specified in the header.

### **Hash Passwords in ASP.NET Core:**

In-order to protect the user credentials at all cost, we store user passwords in the passwords after using a one-way cryptographic hash function. If, in case, a hacker is able to hack our database, the person can never get hold of the user credentials because it is impossible to reverse a one-way

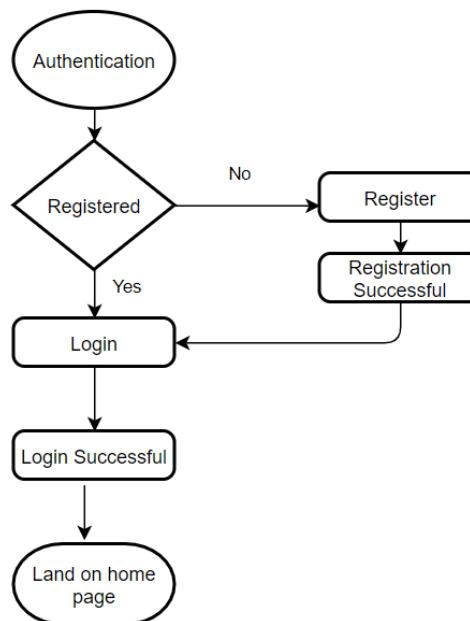
hash. In order to hash the user credentials, we are using a package provided by ASP.NET Core called as Microsoft.AspNetCore.Cryptography.KeyDerivation which hashes the passwords using the PBKDF2 algorithm

#### **4.1.1. Authentication Module Scope**

Role based access controls refers to the restricting the access to the system to some authorized users. It is an access-control mechanism which is built around privileges and roles.

As our application has multiple users including the super admin, manufacturers, retailers and the end-users so implementation of role-based access control was a must. The users in our application can only access features according to their roles and privileges assigned by us. Each user only sees or access features, which have been designed for them and has no control or access over features which do not concern them. We have implemented role-based access control using ReactJs.

#### **4.1.2. Authentication Module Flow diagram**



*Figure 4.1 - Authentication Flow Diagram*

### **4.1.3 Authentication Module Components**

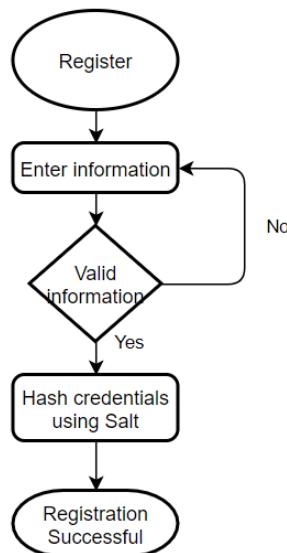
The authentication module of MediQuick application consists of three components that may be used according to the requirements.

These three components are:

- Login Component
- Reset password Component
- Signup Component

#### **4.1.3.1 Authentication Module- Registration Component Description**

Registration component consists of a simple form with validation. However, we have implemented hashing for credentials so that the account can not be hacked easily. If user clicks on Register, A sign up form is opened with fields related to basic contact information, name, email and password. Each field is validated, on clicking register user account is created. User can now login to this newly created account.

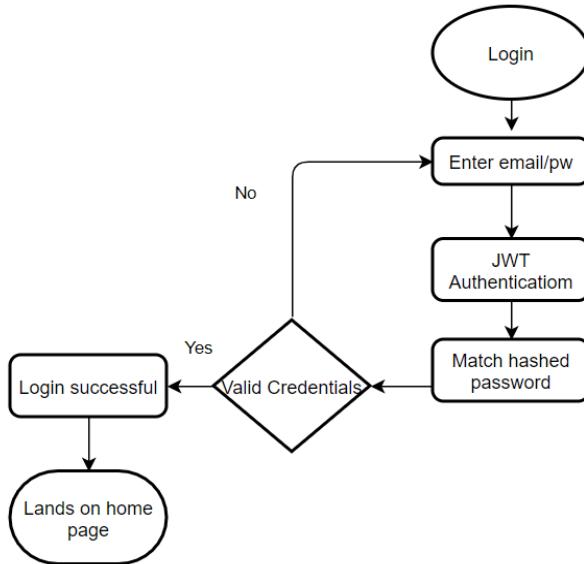


*Figure 4.2 - Registration Flow Diagram*

#### **4.1.3.2 Authentication Module- Login Component Description**

Login requires user to enter credentials (email and password). The user credentials are authenticated using JWT so that the user can only access features according to their roles and privileges assigned by us. Password is hashed to match the hashed password stored in our database. On successful authentication, user is logged into the account.

If user clicks on Login, A small form is opened with fields to enter credentials (email and password). Each field is validated and authenticated to let the user login to their account. On successful login, user is redirected to their homepage.



*Figure 4.3 - Login Flow Diagram*

#### **4.1.4 Restrictions or Limitations**

The limitation of authentication module is that it can affect performance and since MediQuick is a business-oriented application, the performance of our application has serious implications for the success of the business.

#### **4.1.5. Design Constraints**

MediQuick allows the user to signin using email address. You need to enter the valid email address in order to get authenticated. It doesn't allow the user to signin using facebook or phone number.

## **4.2 Description of Client Creation Module**

This module covers the details regarding the creation of client accounts on the application. To initiate the procedure, user has to login with Admin user credentials.

User will then click on option to create the required type of Client. There are three options available as per the categories of clients which are distributor, manufacturer and retailer. Any of the options selected opens up a Create client modal (form) with the required information fields for the respective client type.

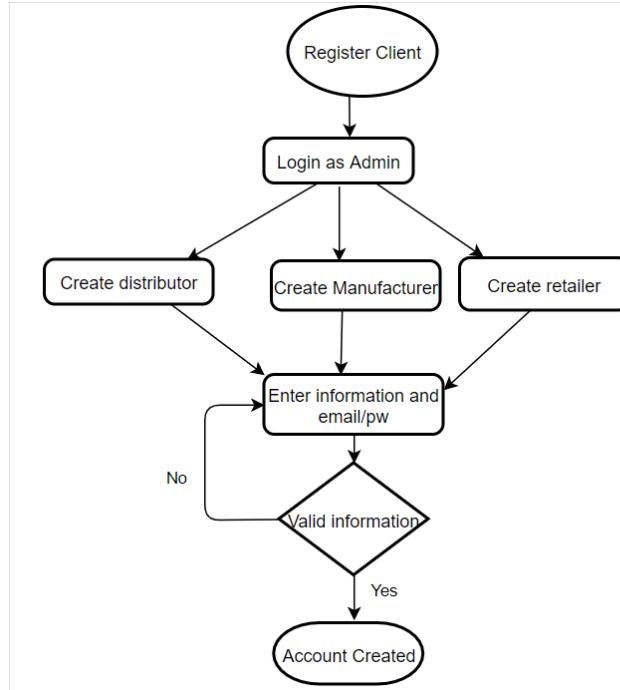
The information entered including email, License details, contact information and other information is entered. The entered information is validated.

If information is successfully validated, user can save it and then client account is registered with a confirmation message of successful account creation or registration. Else, on providing invalid information user is required to reenter the detail which were invalid to continue registration process.

### ***4.2.1 Client Creation Module Scope***

As per our business model, contracts with the clients (distributor, manufacturer and retailer) are signed one-to-one in person to get registered on our system. After which, Admin user creates accounts for the client on our system as per the provided details. This module covers this registration process on the application. The account credentials are then provided to the client for usage who then logins the application and access the granted features.

### ***4.2.2 Client Creation Module flow diagram***



*Figure 4.4 - Client Creation Flow Diagram*

### **4.2.3 Restrictions Limitations**

#### **4.2.3.1 Performance Issues**

Process may get slow due to many validations on the form fields in the creation modal.

#### **4.2.3.2 Design Constraints**

Only Admin user has access to create the client user accounts. Client cannot register themselves unlike end-users. They can only login to the application using the credentials provided by admin.

## **4.3 Description of Medicine Availability Module**

To search medicines in our application, user just needs to access our application. User just go to Search and enter medicine to view results that include availability of medicine in near by stores and suggestion of alternate medicines which may be expanded by user to view details. User then proceeds to view medical store location on map.

Now, user can choose to exit the application after getting the required information or proceed further to reserve the order at medical store. At this stage, Logging in is a must to get valid orders

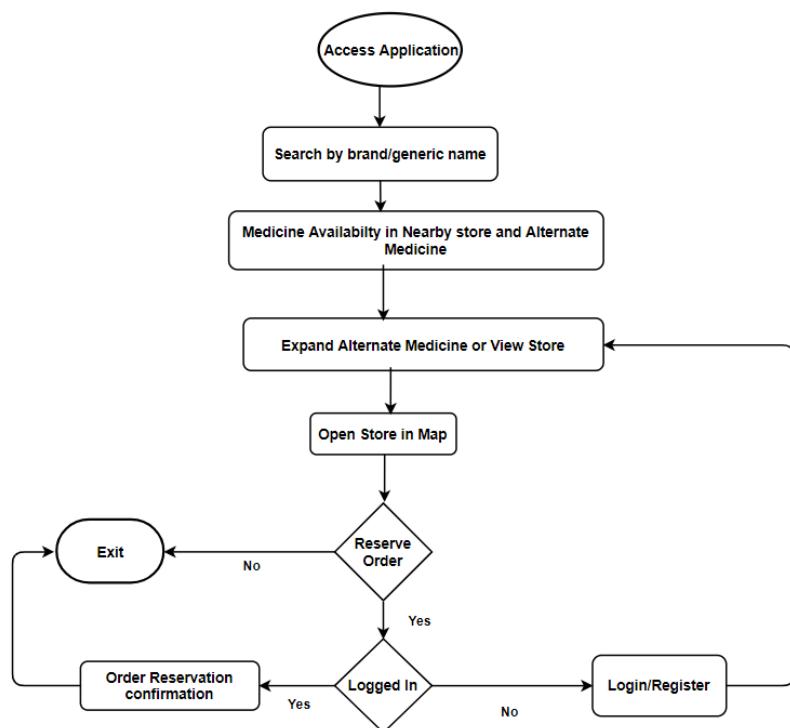
from authentic users. It is checked that user has logged in or not, if it has not and it is redirected to login/register page and after logging in it is redirected back to store.

For a logged in user, reserve order is proceeded by providing details and on user's confirmation order is placed.

#### **4.3.1 Medicine Availability Module Scope**

Medicine availability module covers the mechanism to enable end users to search the required medicines and get information of their availability at nearby medical stores or pharmacies quickly, this supports the case of urgency as logging in is optional. Alternate medicines that contain the exact drug composition are also suggested to the users. User are able to connect to the retailers by reserving order at their store.

#### **4.3.2 Medicine Availability Module flow diagram**



*Figure 4.5 - Medicine Availability Flow Diagram*

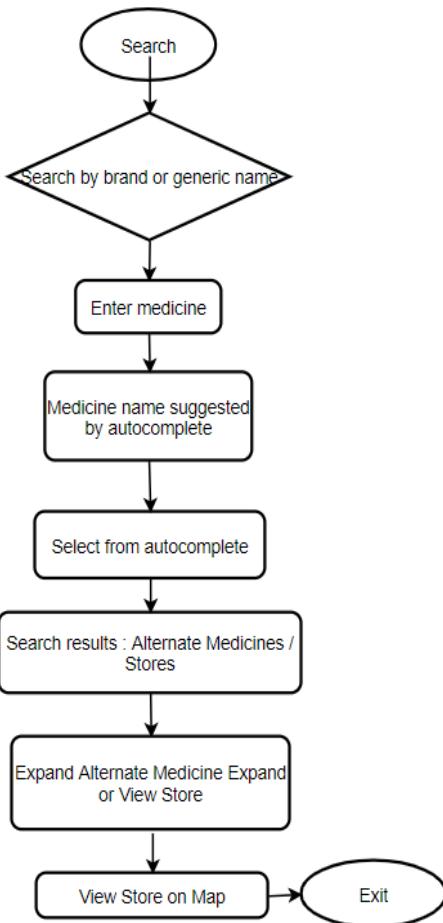
### **4.3.3 Medicine Availability Module Components**

Medicine availability module consists of two major components:

- Search Module
- Reserve Order Module

#### **4.3.3.1 Description of Search Module**

Search module enables user to search medicines available at nearby stores. It even suggest alternate medicines with same drug composition. This benefits the user if the required medicine is not available or it is more costly than the alternate ones. User can view the store location with route to reach there with estimated time of travel, so that he can make a correct choice in urgency.



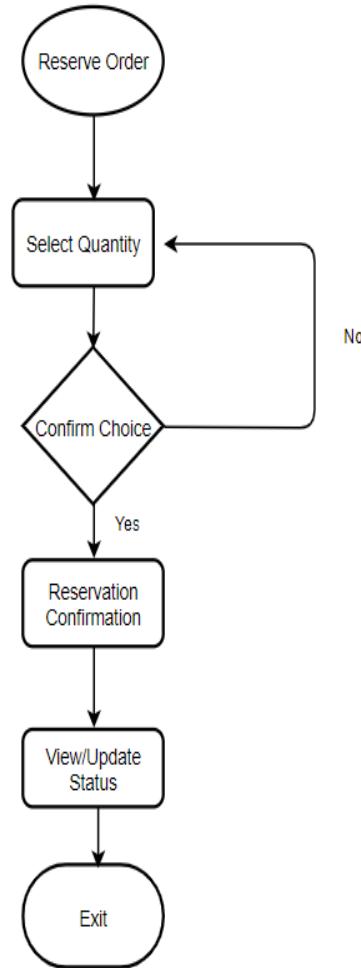
*Figure 4.6 - Search Flow Diagram*

#### **4.3.3.1.1 Search Module Interface Description**

Search module interface consists of a Search bar which has two radio buttons to either select search by brandname or generic name. User then enters the medicine name and system autocompletes it with suggestion as user is only able to search medicines that are

recognized by our system. Results are shown once the user hits enter which include Alternate medicine and medical stores in which the medicine is available. User can further select to expand these results to view details or select a particular store to open and view its location and route on map.

#### **4.3.3.2 Description for Reserve Order Module**



*Figure 4.7 - Reserve Order Flow Diagram*

##### **4.3.3.2.1 Reserve Order Module Interface Description**

After searching the medicine, user may choose to reserve the order at a particular medical store. For this, information will be required to fill by the user in a form. After entering the details and quantity, user will be asked to confirm the choice to reserve the order.

#### **4.3.4 Restrictions/Limitations**

##### **4.3.4.1 Performance Issues**

Search may get slow if user enters the medicine completely wrong, in that case autocomplete will not be able to propose the required medicine name.

##### **4.3.4.2 Design constraints**

Random medicine names cannot be entered in search bar due to autocomplete feature.

Also, to reserve order user has to login or sign up on our application to only allow orders from authentic customers.

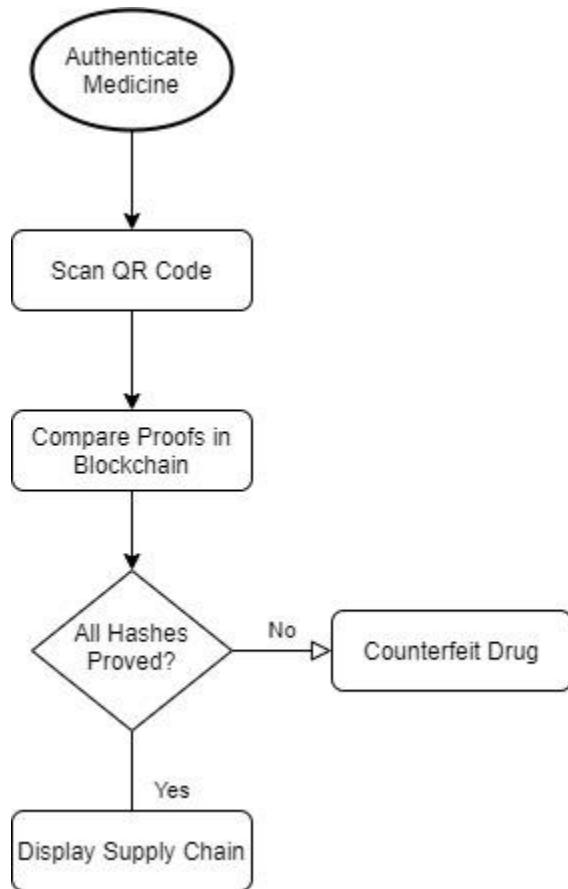
### **4.4 Description for Medicine Authenticity Module**

In this module, user can scan QR codes printed on medicines to verify the authenticity of the medicine by validating the supply chain of the drug.

#### **4.4.1 Medicine Authenticity Module Scope**

We track the supply chain and stock transfers of the medicine from manufacturer warehouse to distributor and then to various retailers and wholesale stores or pharmacies. Each point of supply chain is recorded on the immutable ledger of blockchain which helps the user to know the journey of their purchased medicine.

#### **4.4.2 Medicine Authenticity Module flow diagram**



*Figure 4.8 - Medicine Authenticity Flow Diagram*

#### **4.4.3 Restrictions/Limitations**

##### **4.4.3.1 Performance Issues**

As the public blockchain is slower at times, the validation and authentication process might take a few seconds.

##### **4.4.3.2 Design constraints**

The user does not need to log in to scan QR codes.

## **4.5 Description for Inventory Management Module**

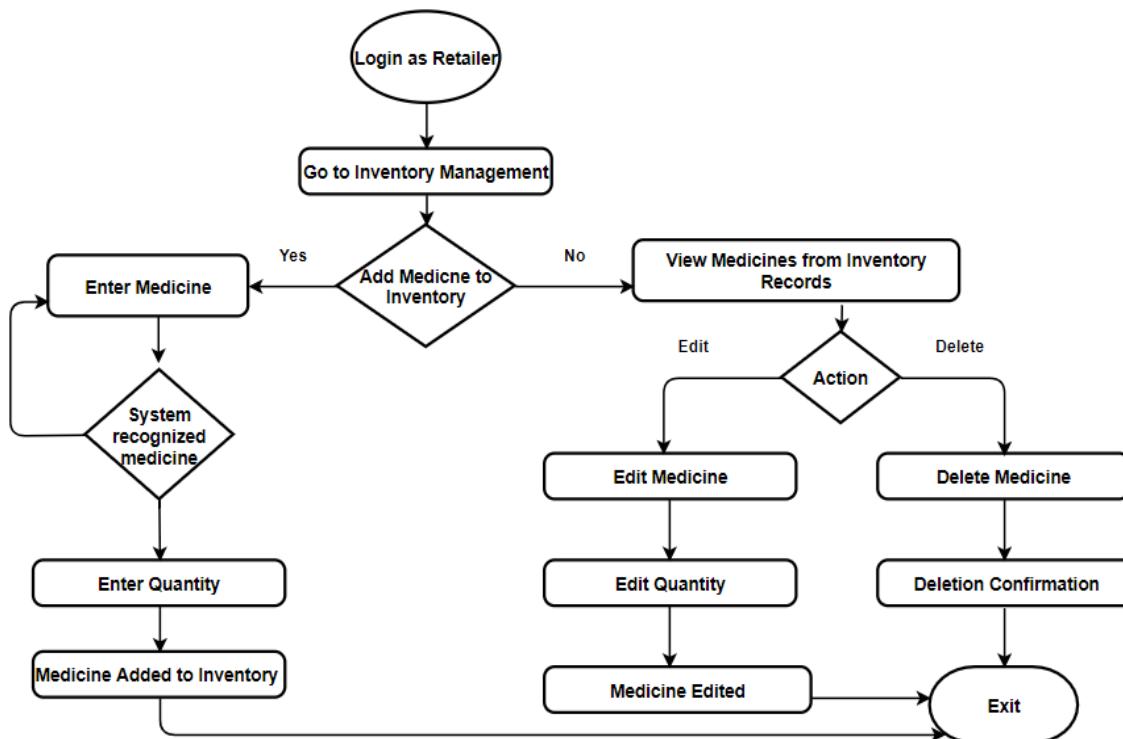
To manage inventory, retailer logins with the credentials provided by admin. Retailer then goes to inventory management.

Option to add new items to inventory is available, if user selects to add a new medicine, a create medicine modal (form) is opened. User enters the medicine name using autocomplete and fill other details such as quantity. In another case, user might not want to create a new item and instead view existing records to perform actions such as edit or delete. If user selects to edit, he may edit details or update the quantity. After confirming the edit operation, inventory item is updated. Similarly, on clicking on delete option, user is asked to confirm its choice. Confirmation message is displayed after each successful operation.

#### **4.5.1 Inventory Management Module Scope**

Inventory Management module is only for our retailer clients. Retailers can manage their inventories on our system. Only our system's recognized medicines can be added as items to their inventory with price and quantity to avoid the possibility of existence of any PDA unrecognized drugs on our system.

#### **4.5.2 Inventory Management Module Flow Diagram**



*Figure 4.9 - Inventory Management Flow Diagram*

### **4.5.3 Restrictions/ Limitations**

#### **4.5.3.1 Performance Issues**

API is optimized to process the query faster so no performance bottlenecks experienced.

#### **4.5.3.2 Design Constraints**

Only retailer client has access to inventory management. Only PDA recognized medicines on our system can be added to the inventory.

## **4.6 Description for Supply Chain Module**

Supply chain punches the user performed actions to blockchain. The actions that are punched to blockchain are Sales and Purchase Order Creation by retailer and distributor respectively, Transfer of stocks and management of warehouse by manufacturers. Each user has to login in order to perform the concerned actions. For each of these actions, transactions on blockchain are created.

### **4.6.1 Supply Chain Module Scope**

Supply Chain module ensures the supply chain management of medicines on our system. It ensures each related action is punched to blockchain so that it can be tracked later to ensure supply of authentic medicines in the chain to each receiver, be it from manufacturer to distributor or from distributor to medical store retailer. Or even, transfer of stocks by manufacturer to its own warehouse are being tracked.

#### 4.6.2 Supply Chain Flow Diagram

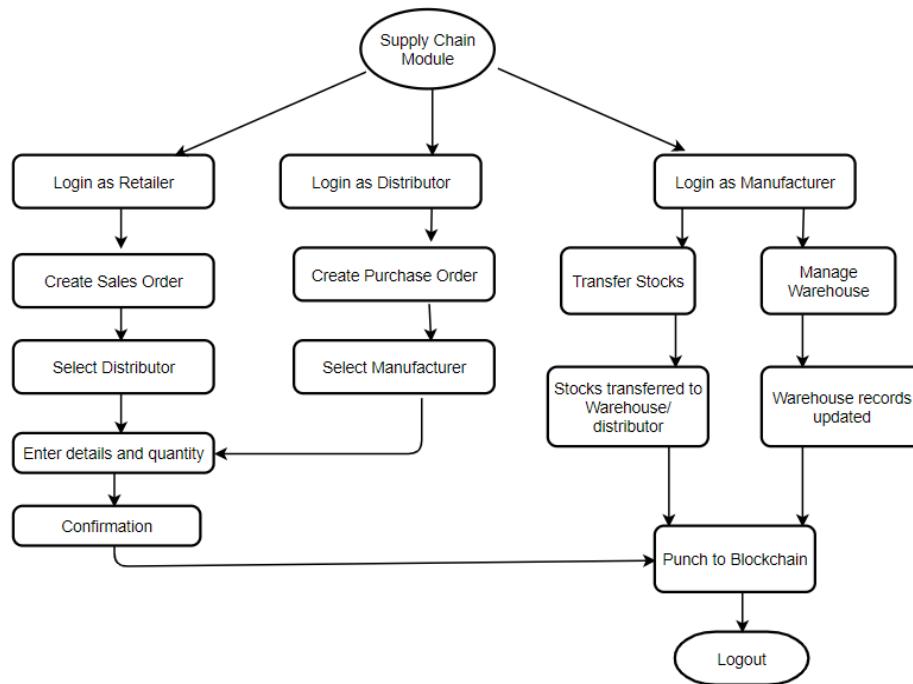


Figure 4.10 - Supply Chain Flow Diagram

#### 4.6.3 Supply Chain Subcomponents

Supply chain module consists of the following subcomponents:

- Create Sales Order
- Purchase Order
- Transfer Stocks
- Manage Warehouse
- Supplychain issues, massive data fetching time

##### 4.6.3.1 Description for Create Sales Order

Medical store retailer creates Sales order to get stock of medicine delivered from any distributor that is registered on our system. The distributor gets notified for the order and fulfills the order. Both of them can view the status of the sales order placed.

To create sales order, user is required to login as retailer with credentials provided by admin. User then goes to Create Sales Order which opens up a modal(form), where user enters the medicine he

wants to order with quantity and selects the distributor who will fulfill this order. On user's confirmation, successful order placement message is shown to the user. This order's status can be viewed by the retailer and distributor (who is also notified for the order).

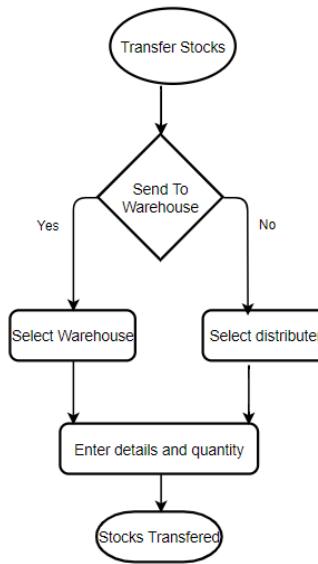
#### **4.6.3.2 Description for Create Purchase Order**

Distributor creates Purchase order to get stock of medicine delivered from any manufacturer that is registered on our system. The manufacturer gets notified for the order and fulfills the order. Both of them can view the status of the sales order placed.

To create purchase order, user is required to login as distributor with credentials provided by admin. User then goes to Create Purchase Order which opens up a modal(form), where user enters the medicine he wants to order with quantity and selects the manufacturer who will fulfill this order. On user's confirmation, successful order placement message is shown to the user. This order's status can be viewed by the distributor and manufacturer (who is also notified for the order).

#### **4.6.3.3 Description for Transfer Stocks**

Transfer of stocks enables the manufacturers to transfer their stocks to warehouse or to the distributors. Transaction on blockchain is created for each transfer by movement of token.



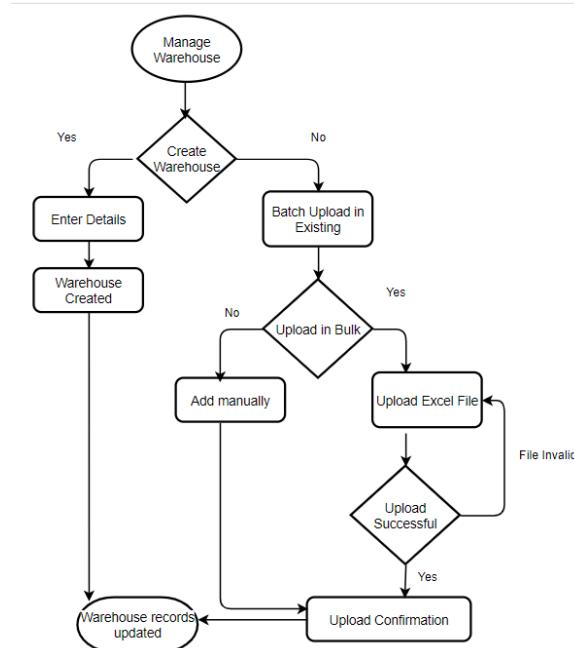
*Figure 4.11 - Transfer of Stocks Flow Diagram*

User logs in as manufacturer and clicks on Transfer Stocks Option. Here, user may want to send stock to its own warehouse, if not he can send it to a distributor to whom he is connected to on our

system. On selecting either of the options, a form opens up where information is required to be filled such as medicines (from autocomplete) and their quantity along with batch no and expiry date etc. After confirming the transfer, user is shown a message on success of the action performed. Manufacturer can only transfer the medicines that were either created by them or added by the admin under manage medicines.

#### **4.6.3.4 Description for Manage Warehouse**

Manage warehouse enables the logged in manufacturer to create new warehouse or upload batch of medicines to existing ones. This batch upload can be done manually or in bulk using an excel file to save time. Warehouse records are updated on all actions. Each Batch upload creates a token in blockchain.



*Figure 4.12 - Manage Warehouse Flow Diagram*

User logs in as manufacturer using the credentials provided by admin. User then goes to Manage Warehouse options. Here, either he can create a new warehouse or upload batches to an existing one. If user clicks on create new warehouse, he is required to fill a form with details of warehouse such as warehouse location. On filling the required valid information and saving it. Warehouse is

created with a confirmation message displayed on the screen. If user wants to upload batches, further two options are available that are either to add batch records manually or to simply upload an excel file with data in correct format. If file is valid and has correct format batch is successfully uploaded in bulk. Whereas, manual upload requires filling out forms. On successful upload, confirmation message is displayed.

#### ***4.6.4 Restrictions/ Limitations***

##### ***4.6.4.1 Performance Issues***

Supplychain performance might get slow with increasing number of transactions on blockchain.

##### ***4.6.4.2 Design Constraints***

Every action performed creates transactions on blockchain to ensure consistency of the supplychain. And only, medicines recognized by our application are allowed to be used.

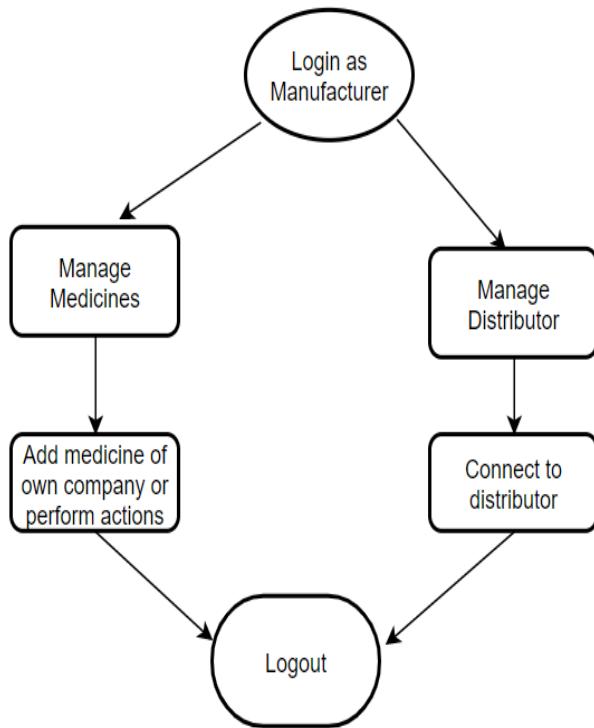
### **4.7 Description for Manufacturer Module**

A logged in manufacturer has various options available. Out of which, two major one are to manage medicines and manage distributers. Manage medicine deals with add, edit and delete operations on medicines whereas manage distributer lets manufacturer connect to the distributers for stocks transfer.

#### ***4.7.1 Manufacturer Module Scope***

Our system enables manufacturers to manage the medicine of their own company using our application. This also allows them to connect to distributers they already deal with or others that are present on our system.

#### **4.7.2 Manufacturer Module Flow Diagram**



*Figure 4.13 - Manufacturer Module Flow Diagram*

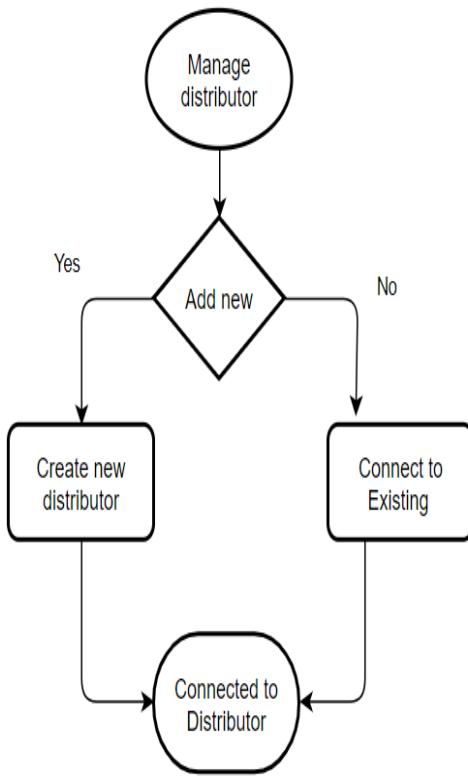
#### **4.7.3 Manufacturer Module sub-components**

Manufacturer module consists of two subcomponents:

- Manage distributor
- Manage Medicines

##### **4.7.3.1 Description for Manage Distributer**

Manufacturer manages distributors in terms of creating new distributors (those they already deal with) or make new connections with the already existing ones on our system.



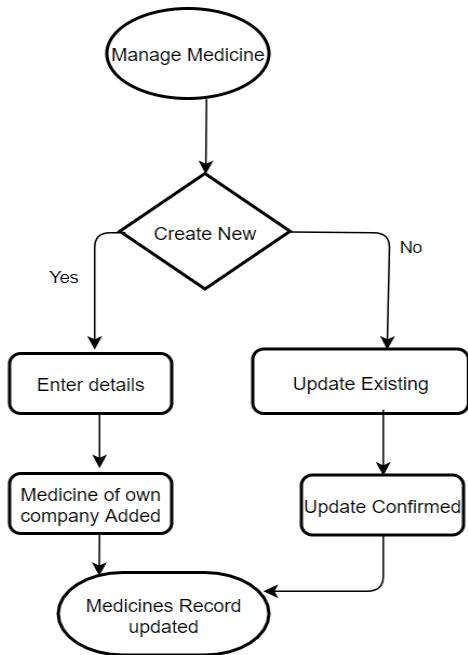
*Figure 4.14 - Manage distributor Flow Diagram*

User is required to login as manufacturer. He then clicks on Manage distributor option. User can either create new distributor which are not currently registered on our system, on choosing this option, user fills the required information which on being validated and confirmed will create the distributor on our system and create a connection between the manufacturer and the newly created distributor.

User can also select to connect with the distributor that are currently registered on our application. Confirmation message is shown on successful connection.

#### **4.7.3.2 Description for Manage Medicines**

Manufacturer can manage the medicines of their company. They may add any new medicine launched by their company into the market to our system or may edit the details of those who are already present. They can also remove any medicine if they want to.



*Figure 4.15 - Manage Medicines Flow Diagram*

User logs in as manufacturer using the provided credentials (email/password). He then selects to go to Manage medicines. Here, if he chooses to create a new medicine, a form is opened which on filling out with valid information in the fields adds the newly created medicine to the records. User can also view other medicine and update them by performing operations such as edit and delete, update is confirmed by the user. All the actions performed update the medicine records of our system and shows confirmation message to the user.

#### **4.7.4 Restrictions/ Limitations**

##### **4.7.4.1 Performance Issues**

As the no of medicine records will increase over time, the time for fetching the database might increase with the increase in medicine records.

##### **4.7.4.2 Design Constraints**

Medicines of a manufacturer's own company are only visible. Manufacturer user is restricted to view medicines manufactured by other companies.

## **4.8 Description for Admin Module**

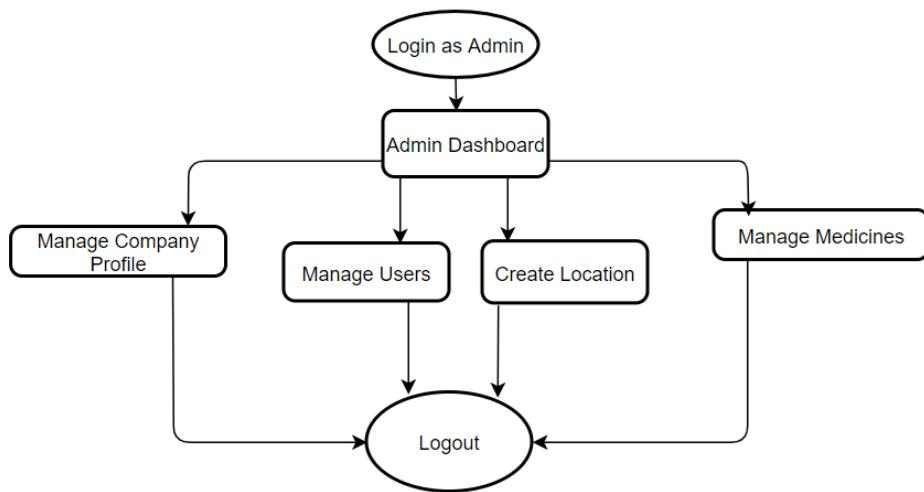
A logged in Admin user will be redirected to Dashboard where it is able to perform multiple operations which include management of all medicines existing on our system, management of other admin users to update the granted access or even create new or remove the existing ones. Admin can also update the company information and official documents. New locations can be created ( cities, states or countries).

### **4.8.1 Admin Module Scope**

Just like any other application, admin user is the administrator who has access to perform management operations such management of users and update the information related to application and our organization.

Moreover, admin can manage our domain-oriented entities such as medicines and locations.

### **4.8.2 Admin Module Flow Diagram**



*Figure 4.16 - Admin Module Flow Diagram*

### **4.8.3 Admin Module Subcomponents**

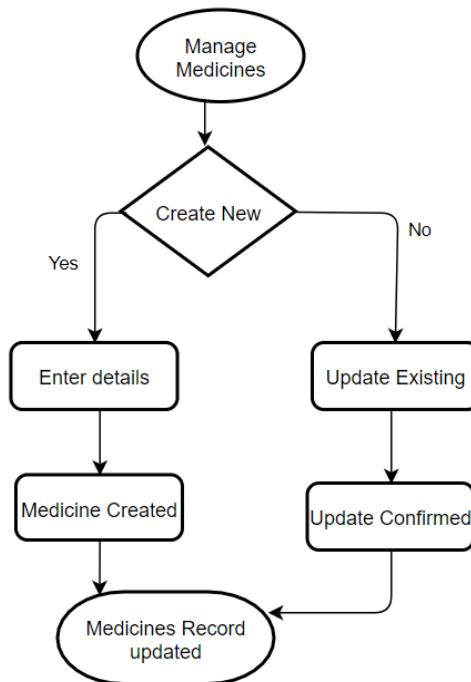
Admin module consists of following subcomponents:

- Manage Medicines
- Manage Users
- Manage Company profile

- Create Location

#### **4.8.3.1 Description for Manage Medicines**

Admin is able to view all the medicines currently present on the system. It has access to update any of these medicines by either editing the related information or deleting a medicine completely.

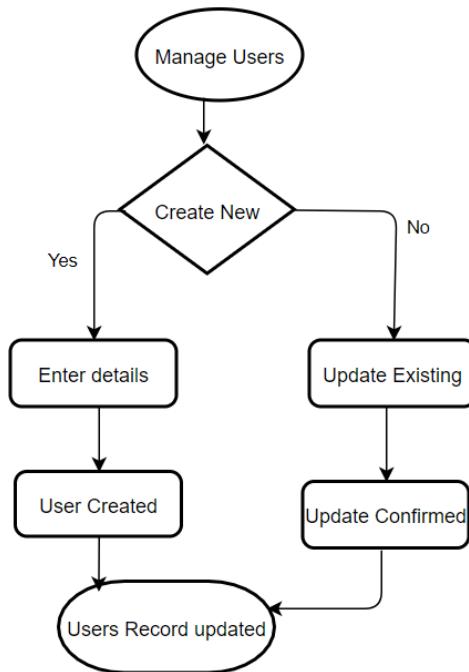


*Figure 4.17- Manage Medicines Flow Diagram*

User logs in as Admin using the credentials (email/password). He then selects to go to Manage medicines. Here, if he chooses to create a new medicine, a form is opened which on filling out with valid information in the fields adds the newly created medicine to the records. User can also view other medicine and update them by performing operations such as edit or delete; update is confirmed by the user. All the actions performed update the medicine records of our system and shows confirmation message to the user.

#### **4.8.3.2 Description for Manage Users**

Admin is able to manage other admin user currently present on the system. It has access to update any of these by either editing the related information or updating the granted access or even deleting a user completely.



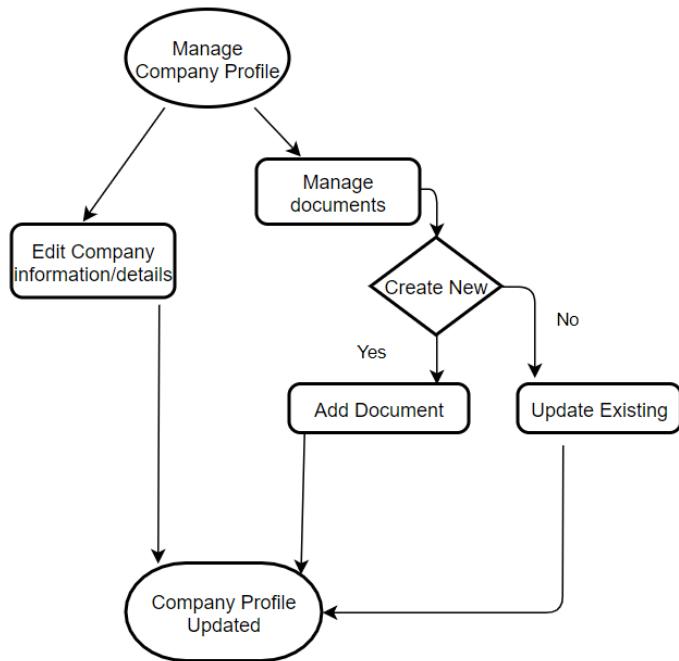
*Figure 4.18 - Manage Users Flow Diagram*

User logs in as Admin using the credentials (email/password). He then selects to go to Manage Users. Here, if he chooses to create a new admin user, a form is opened which on filling out with valid information in the fields adds the newly created admin user to the records. User can also view other users and update them by performing operations such as edit and delete; update is confirmed by the user. All the actions performed update the records of our system and shows confirmation message to the user.

#### **4.8.3.3 Description for Manage Company Profile**

Company Profile contains information that is related to our organization ‘MediQuick’.

Admin can update the initially entered details or upload official documents of our company. Multiple actions such as download, upload, edit are also available for the documents.



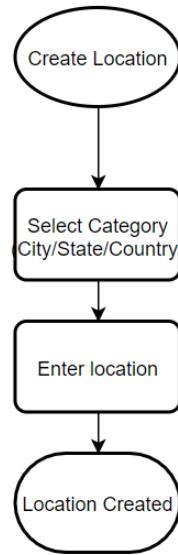
*Figure 4.19 - Manage Company Profile Flow Diagram*

Admin logs in to the application and goes to Company Profile. Here he can choose to click edit and update the available fields in the information form. User confirms the edit and the company profile gets updated.

Similarly, he can view the list of company documents, icons for available actions are available with each document such as download, edit, document type etc. User may choose any of these actions or update the document description or name. Add new document option is also available which on click, asks for the description, name and file attachment to upload the document file. For any update in the documents or profile information, user sees a confirmation message.

#### **4.8.3.4 Description for Create Location**

As our system will expand its coverage to include new cities, more locations will be added in our application. These locations are used for storing address information of our users or orders or stocks transfers.



*Figure 4.20 - Manage Location Diagram*

After logging in, admin will click on Create Location. A form modal is opened where he selects the category of location (city, state or country), then name is required to be filled from autocomplete. On saving the location, message for successful creation is displayed.

# **CHAPTER 5 - BEHAVIORAL MODEL AND DESCRIPTION**

---

## **5.1 Description for system behavior**

A detailed description of major events and states is presented in this section.

### **System behavior for End User / Patient:**

End user access the application to and on main page search medicines in nearby stores or view substitute medicines, user then selects a medicine and view store on a map. They can further reserve order for the medicine shown in result for which login or registering on the application will be required. They can also verify the authenticity of medicine by simply scanning the QR Code. Logged-in users can view order histories on their home page.

### **System behavior for Retailer:**

Retailers are required to login with credentials provided by admin or owners of our organization. After logging in, retailer is redirected to homepage where get notifications for orders reserved by users and orders' history with status will be displayed which can be updated. Apart from this, retailers also manage their own inventory and add edit or delete inventory items. They can also create sales orders to distributers to receive medicine stock and view the status and history.

### **System behavior for Distributer:**

Distributers are required to login with credentials provided by admin or owners of our organization. After logging in, distributer is redirected to homepage where get notifications for sales orders placed by retailer and orders' history with status will be displayed which can be updated. They can also create purchase orders to manufacturer to receive medicine stock of their company and view the status and history.

### **System behavior for Manufacturer:**

Distributers are required to login with credentials provided by admin or owners of our organization. After logging in, distributer is redirected to homepage where get notifications for purchase orders placed by distributer and orders' history with status will be displayed which can be updated. They can get connected to distributers on the system for stock transfers. The stock

transfer feature enables them to either send stock to warehouse or to some connected distributor. Apart from this, they manage their warehouse by creating new or updating the existing warehouse by uploading batches in bulk. Medicine Management is another feature that provides add and update (edit/delete) options for the medicine of their company.

### **System behavior for Admin:**

Admin will login to the account by entering credentials and land on Dashboard. Admin dashboard has multiple options including manage users, manage medicines and locations which enables the user to view, add or update(edit/delete) these. Company profile can also be updated by the user by editing information or uploading company documents.

#### ***5.1.1 Events/interrupts***

A listing of events (control, items) that will cause behavioral change within the system is presented.

- Search:

User enters the medicine name in search bar on the main page. If user selects the alternate medicine from results it stays on the results page, whereas on selecting the nearby store, map is opened to show the store location.

- Reserve Order:

User can reserve order for searched medicine for which a notification event for retailer is also triggered.

- Scan QR Code of medicine:

To verify the authenticity of medicine, user scans the QR Code on the medicine packaging.

- Register user:

To register on our application, user fills the registration form with required fields. If valid information is entered user accounts get created, else registration is failed and user is asked to retry by filling out correct information.

- Authentication:

Authentication event provides access to the application. This event is considered as the main event of the application in terms of security. Authentication event has two possibilities.

In case, the user enters valid credentials (email/password), the system will redirect the user to the homepage after logging in. If the user doesn't provide valid credentials, he will be asked to retry.

- Add, edit or delete inventory items:

Retailer user adds new items to the inventory, or edit the current items or delete them. User confirms the action performed.

- Place Sales Order:

Retailer creates sales order for medicine stock to a distributor for which a notification event for distributor is also triggered.

- Place Purchase Order:

Distributor creates purchase order for medicine stock to a manufacturer for which a notification event for manufacturer is also triggered.

- Connect to distributor:

User connects to distributor. There are two possibilities in this case, user either connects to existing distributor or creates a new distributor and connects to that.

- Transfer Stocks:

Manufacturer transfer stocks to either its warehouse or send it to some distributor. User then confirms its choice.

- Create Warehouse or update by batch uploading in bulk:

Manufacturer creates a new warehouse by entering information or update any warehouse by uploading medicine batch in bulk.

- Add or update users:

Admin user adds new users to the system, or update the current ones by edit or delete options. User confirms the action performed.

- Add or update locations:

Admin user adds new location to the system, or update the existing ones by editing or delete them. User confirms the action performed.

- Update Company Profile:

User updates the company profile by editing information or update documents by editing details or upload new ones. He also can select to download a document.

- Add or update medicines:

Admin user adds new medicine to the system, or view the current ones to update by editing or delete them. User confirms the action performed.

### **5.1.2 States**

- Search results shown:

Search is in result state on the next page, with alternate medicines and nearby stores having that medicine are displayed.

Location is shown on map for the selected store.

- Order reserved:

Order is in reserved state when user confirms the order. Retailer is notified when notification event is triggered.

- User logged in / Authenticated:

This state is achieved when the user triggers the authentication interrupt. The user stays in this state after entering the valid credentials i.e username and password and then the user will have the specified access to the functionality of the application.

- User registered:

This state is achieved when the user triggers the registration interrupt. The user gets into state after entering the valid information and successfully creating the account. User can now login to utilize the functionalities of the application.

- Medicine authenticated or unauthenticated:

QR Code scanned by the user is processed and back tracked on supply chain. If track is found, medicine is authenticated and all movements of supply chain are displayed. Otherwise medicine is unauthenticated.

- Inventory updated:

Inventory is in updated state when user confirms to add, edit or delete an inventory item.

- Sales order placed:

Sales order is in created state when user confirms the order. Distributer is notified when notification event is triggered.

- Purchase order placed:

Purchase order is in created state when user confirms the order. Manufacturer is notified when notification event is triggered.

- Connected to distributer:

Connected to distributer state is achieved when user confirms to create connection to an existing or newly created distributer.

- Stocks transferred:

When stocks transfer to warehouse event is triggered, bulk transfer state is achieved, whereas transferring to distributer state is changed to fulfilling purchase order state. Finally, stock is in transferred state on user's confirmation.

- Warehouse records updated:

Warehouse records are in updated state when user creates a new warehouse or when it uploads a medicine batch to a warehouse.

- User records updated:

User records are in updated state when user creates a new user or when it updates (edit/delete) it.

- Locations updated:

Locations updated state is achieved on when add or update(edit/delete) event is triggered on locations.

- Company profile updated:

Company profile goes into updated state is when add or update(edit/delete) event is triggered on company information or documents. A document is in downloaded state when download button is clicked.

- Medicines updated:

List of medicine records in the system is in updated state when admin adds or updates (edits/deletes) a medicine.

### 5.1.2.1 State Transition Diagrams

*Depict the overall behavior of the system.*

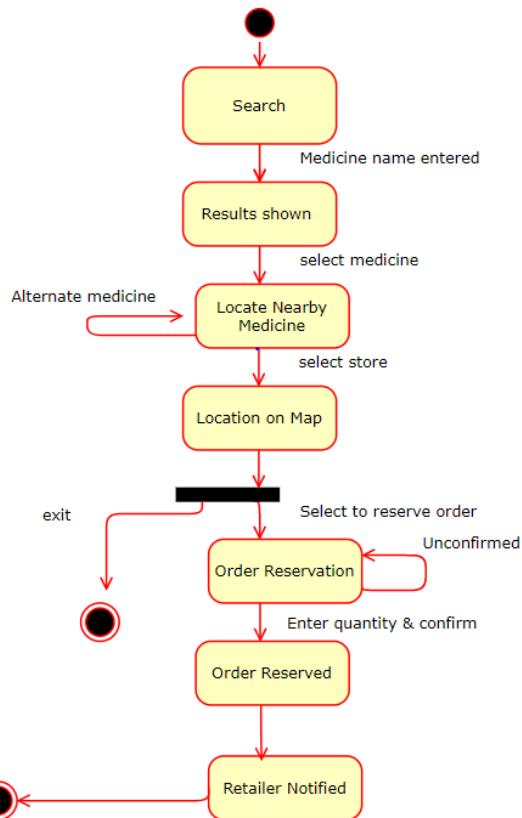


Figure 5.1 - Search and Order Reserved State Diagram

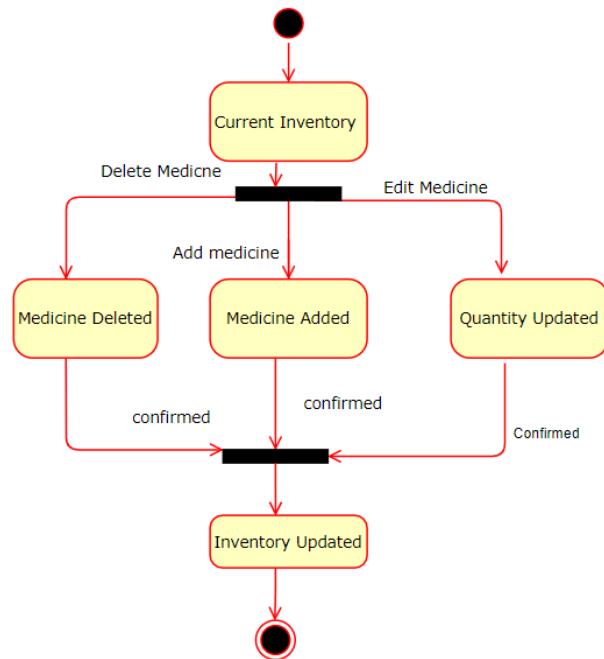


Figure 5.2 - Inventory Updated State Diagram

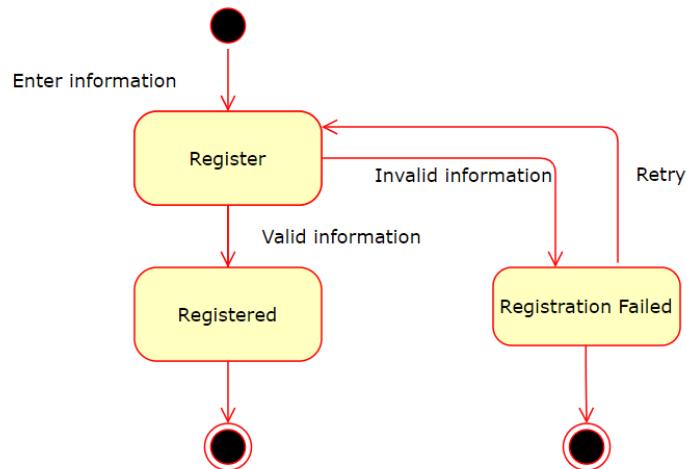
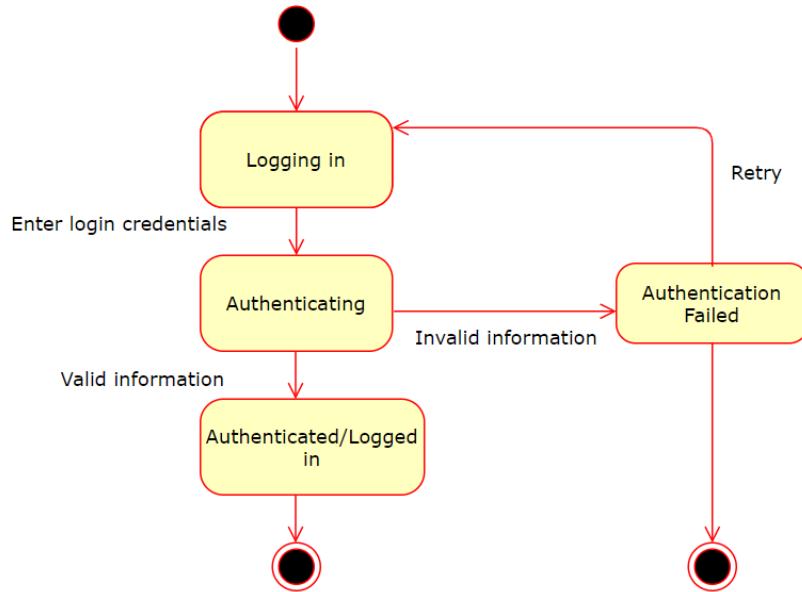
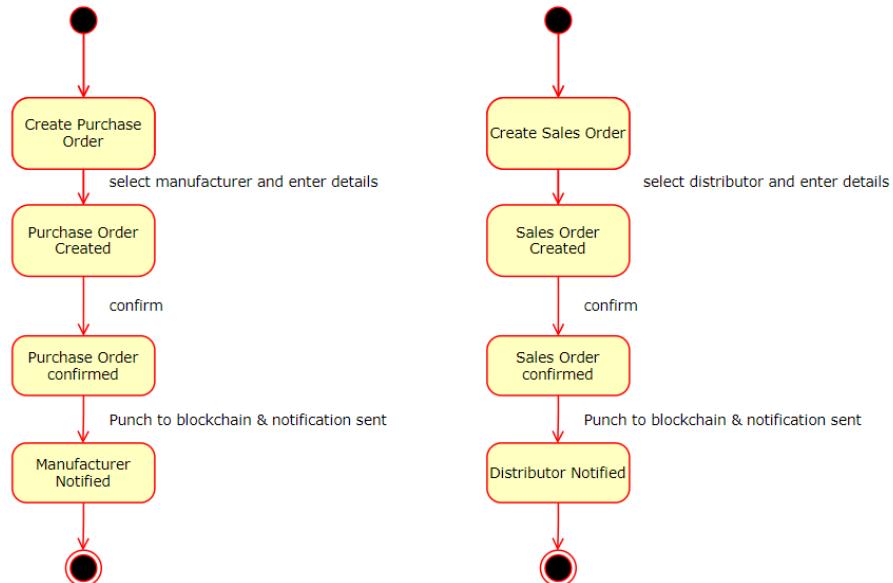


Figure 5.3 - Registered State Diagram



*Figure 5.4 - Authenticated State Diagram*



*Figure 5.5 -Purchase/ Sales order created State Diagram*

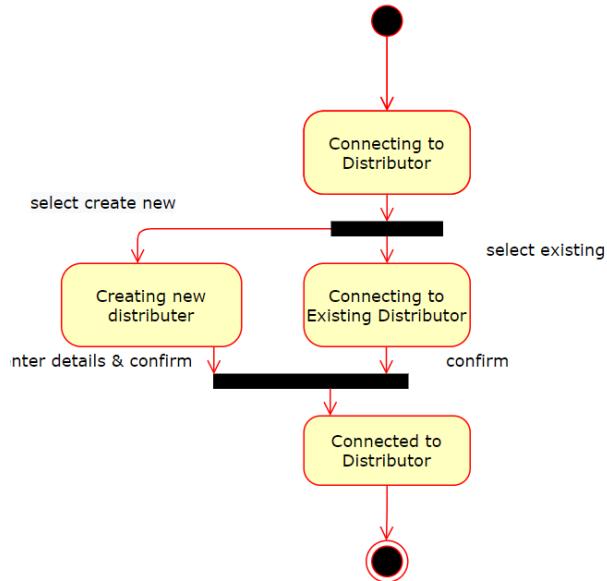


Figure 5.6 - Connected to distributor State Diagram

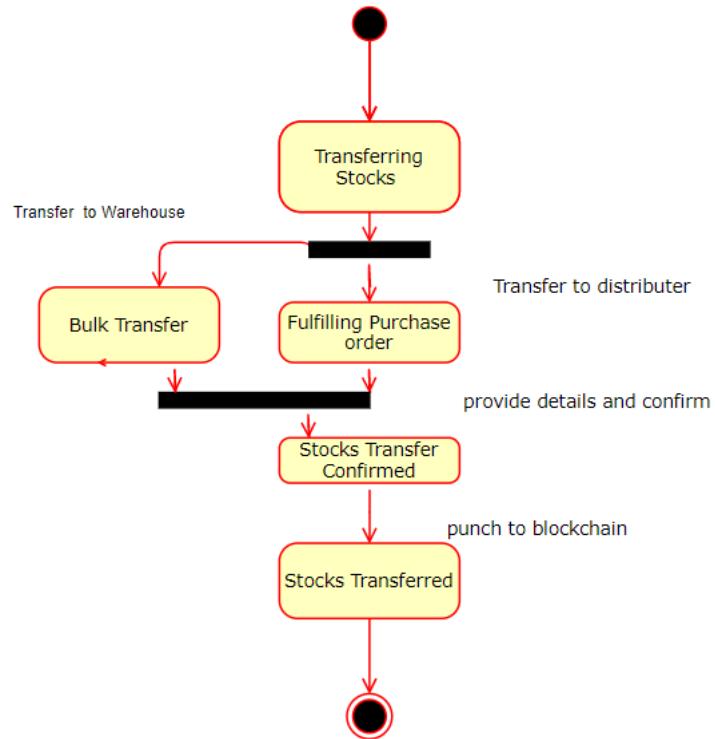


Figure 5.7 - Stocks Transferred State Diagram

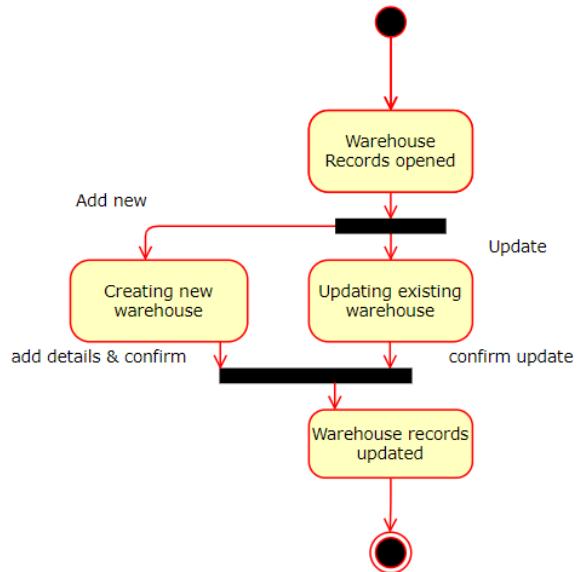


Figure 12 - Warehouse Records Updated State Diagram

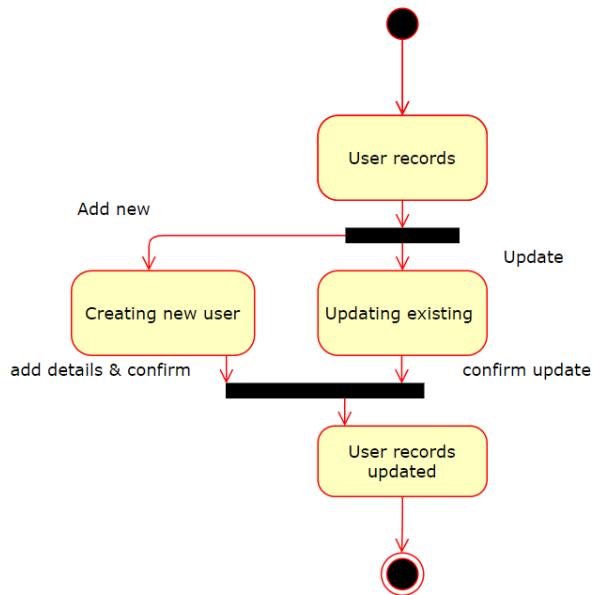


Figure 5.8 - User records Updated State Diagram

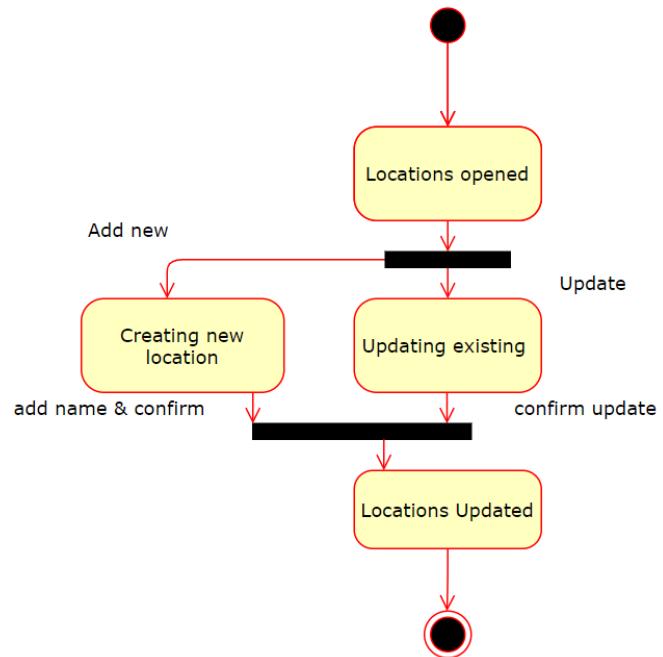


Figure 5.9 - Location Updated State Diagram

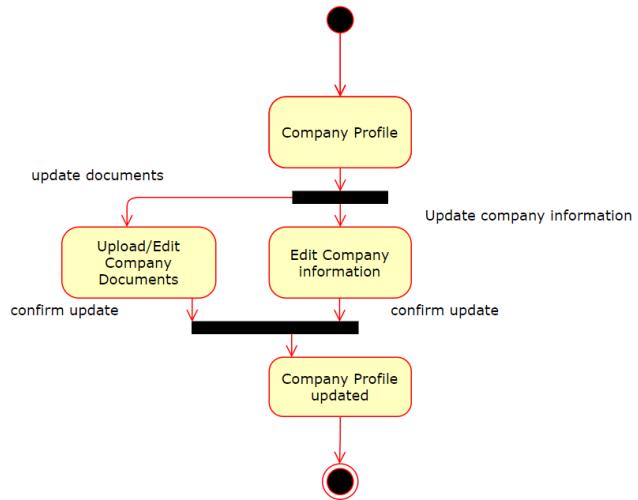


Figure 5.10- Company Profile Updated State Diagram

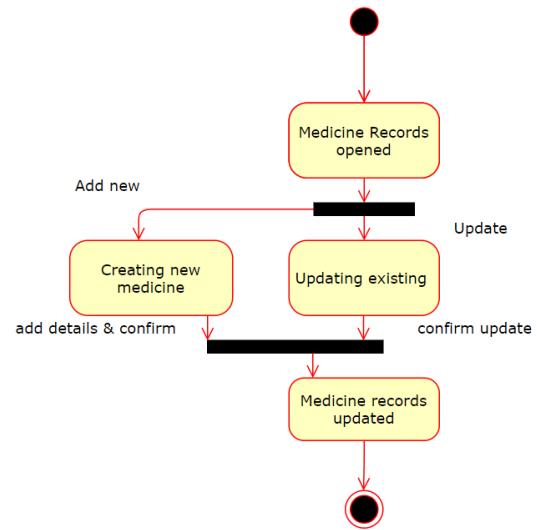


Figure 5.11 - Medicine Records Updated State Diagram

## **5.3 Control specifications**

### ***5.3.1 Control specifications for End user:***

System controls starts with searching medicines event, where it is transferred to Search results state. Now, selecting store event transfers it to location on map state, whereas selecting alternate medicine event returns control to search the result state. Now user can exit or trigger a reserve order event transfers the order in reserved state.

When user triggers registering or authentication events, control is transferred to registered, authenticated or failed state based on validity of credentials.

Scanning QR Code shifts control to authenticated or unauthenticated medicine states by passing through supply chain back tracking state.

### ***5.3.2 Control specifications for Retailer:***

When retailer triggers authenticating event, control is transferred to either authenticated or failed state based on validity of the entered credentials.

Once logged in, retailer can trigger events of add, edit or delete which changes the state of current inventory to the updated state.

Similarly, create sales order event cause the order be in created state.

### ***5.3.3 Control specifications for Manufacturer:***

When manufacturer triggers authenticating event, control is transferred to either authenticated or failed state based on validity of credentials.

Once logged in, manufacturer can trigger events of add, edit or delete on medicines or warehouses which changes the state of current records to the updated state.

If user clicks on transfer stocks, there are two possibilities; to go to either bulk transfer or transfer to distributor state. Both eventually goes to stocks transferred state.

#### ***5.3.4 Control specifications for Distributer:***

When distributer triggers authenticating event, control is transferred to either authenticated or failed state based on validity of credentials.

Once logged in, distributer can select to create purchase orders which on confirmation changes the order state as created.

#### ***5.3.5 Control specifications for Admin:***

On triggers authenticating event, control is transferred to authenticated or failed state.

Once logged in, retailer can trigger events of add, edit or delete on medicines, users, locations and company profile information which changes the state of current records to the updated state.

# **CHAPTER 6 - SYSTEM PROTOTYPE MODELLING**

---

## **6.1 Description of System Modelling Approach**

We have developed different abstract models to present different views or perspective of the system. We have represented the system modelling using graphical notations (in the form of UML diagrams in the previous chapters' i.e use-case diagrams, data flow diagrams, state-transition diagrams). This approach is used to get a basic understanding of the system behavior and interaction. System modelling approach has helped us to understand the functionality of the system.

## **6.2 Prototyping Requirements**

In the initial stage our project, we build a prototype of our web application. The prototype developed had the implementation of basic requirements of our application including search medicines, viewing availability of medicine and locating stores on map, register/login for users and inventory management.

Based on this prototype, we refined the requirements, improved our design and enhanced the implementation. Continuing the same flow, we eventually developed our finally application.

Our protype can be depicted by looking at the screenshots attached below.

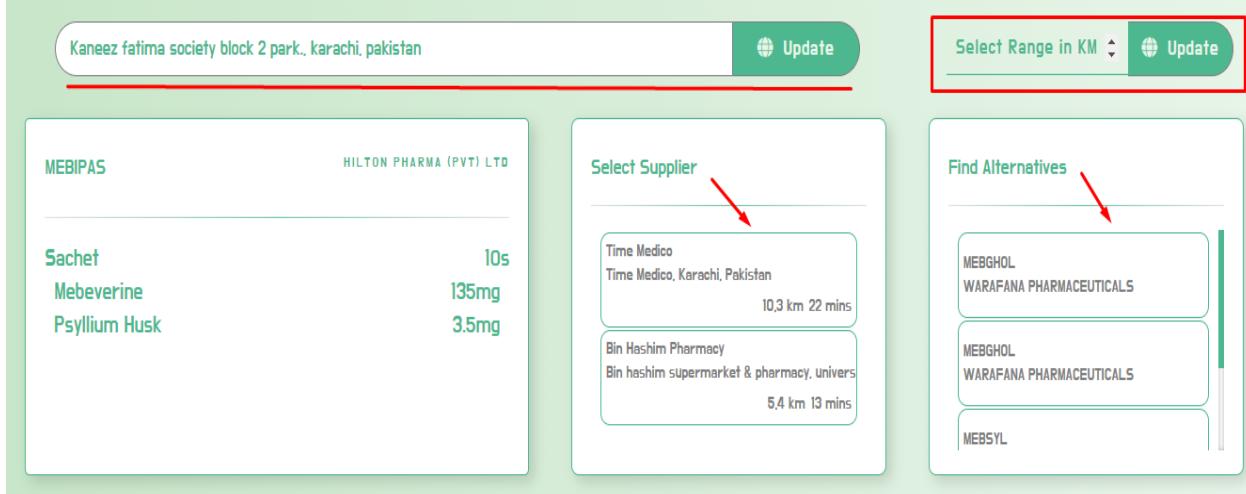


Figure 59 – Search Results

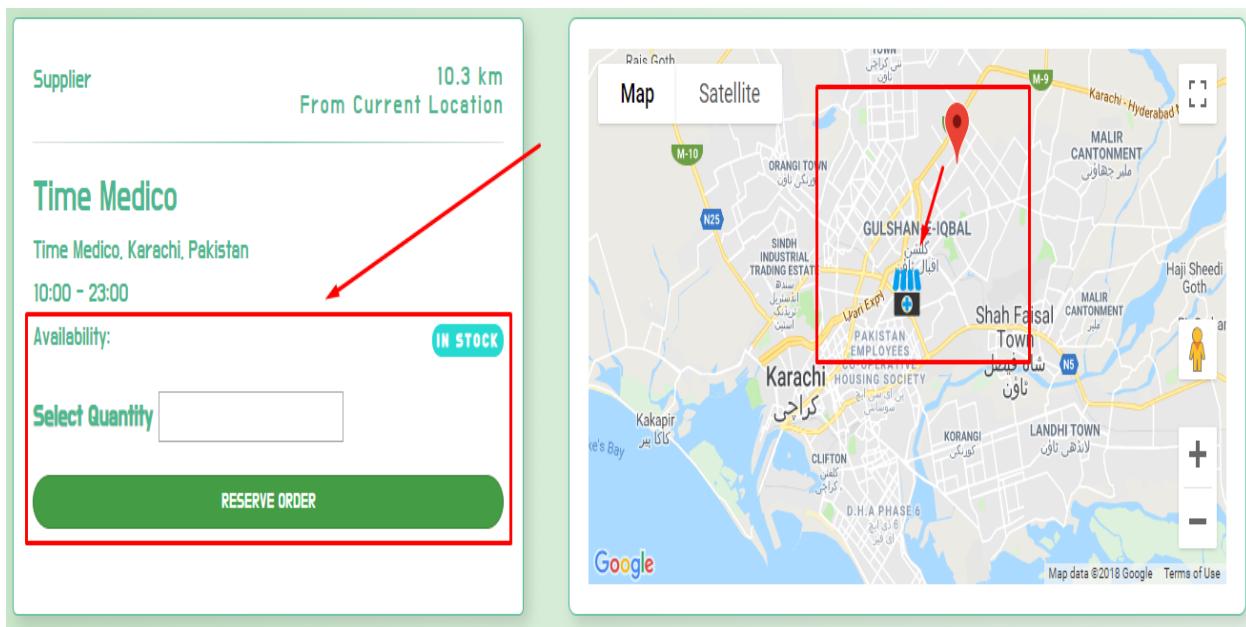


Figure 6.1 – Reserve Order

**Track Your Reservations**

All	Pending	Expired	Fulfilled	Unfulfilled
<b>All Orders</b>				
43 Adnan Medical Store	SPADIX TABROS PHARMA	4mix6s Injections	5	2018-09-14 12:34:16am
44 Time Medico	ANAFORTAN PLUS AGP (PVT) LTD	6sx4ml Injections	2	2018-08-14 08:08:40pm
45 Time Medico	ANAFORTAN PLUS AGP (PVT) LTD	6sx4ml Injections	2	2018-08-14 08:08:40pm

Figure 6.2 – Track Reservations

**MediQuick**

- HOME
- Dashboard
- APPS
- Analysis
- FEATURES
- Tables
- FORMS
- Inventory

15
 27
 1

Total Orders
Inventory
Customers

#### Recent Orders

#	Supplier	Product	Packing	Qty	Date	Status
1	Usman Sabir	BUSCOPAN PLUS MERCK PRIVATE LTD	10x10s Tablets	4	2018-09-26 09:32:13am	PENDING
2	Usman Sabir	MEBSYL NEUTRO PHARMA	10s Sachet	1	2018-09-15 12:08:37pm	PENDING
3	Usman Sabir	BUSCOPAN MERCK PRIVATE LTD	1mlx30s Injections	4	2018-08-21 01:54:39pm	PENDING

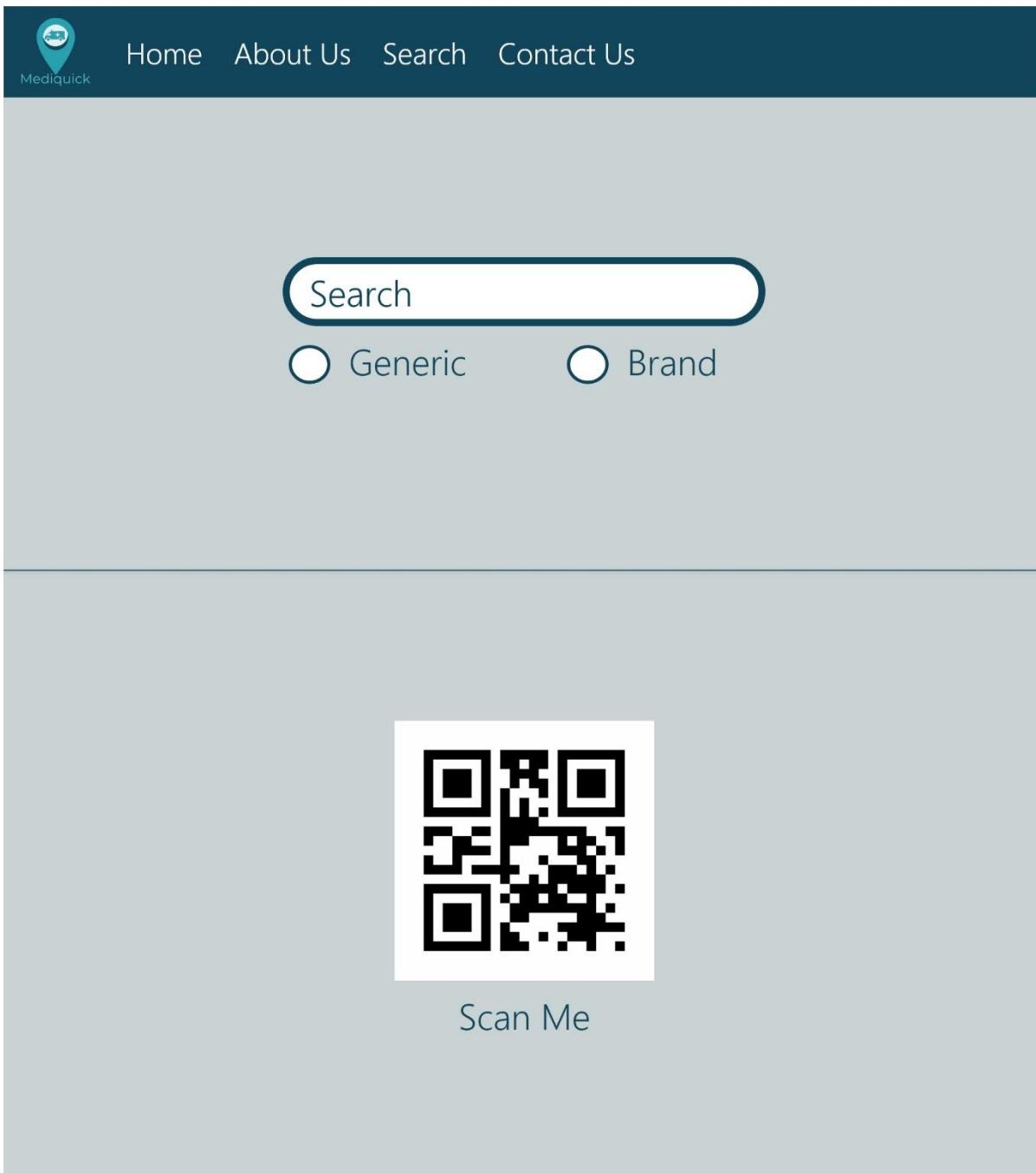
Fulfilled Orders
53%

Pending Orders
40%

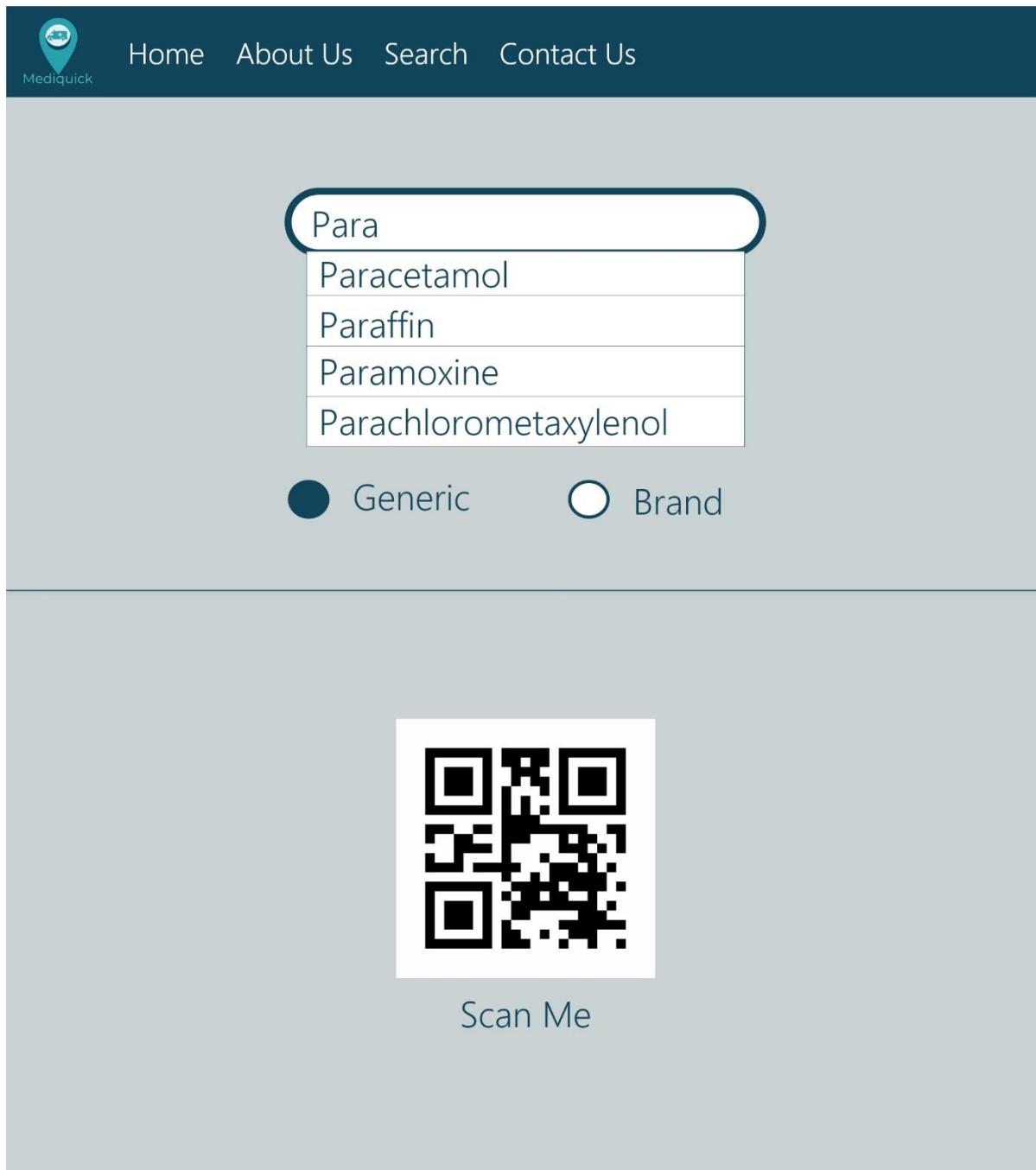
Unfulfilled Orders
0%

Expired Orders
6%

Figure 6.3 – Order History



*Figure 6.4 - Search*



*Figure 6.5 - Search*

The screenshot shows the Mediquick mobile application interface. At the top, there is a navigation bar with a logo on the left and links for Home, About Us, Search, and Contact Us. Below the navigation bar, the search term "Paracetamol" is displayed in a search bar. The main content area is titled "Available Brands" and lists several brands: Acepol (repeated four times) and Benamol. Below this, there are two sections: "Injections" and "Tablets". The "Injections" section shows packing details (2ml x 1s) and price (Rs. 100) for a product containing Lignocaine, Paracetamol, and Benzyl Alcohol. The "Tablets" section shows packing details (200s) and price (Rs. 100) for a product containing Paracetamol.

Packing	Price
2ml x 1s Lignocaine: 10mg/ml, Paracetamol: 150mg/ml, Benzyl Alcohol: 15mg/ml	Rs. 100

Packing	Price
200s Paracetamol: 500mg	Rs. 100

Figure 6.6 - Available Medicines

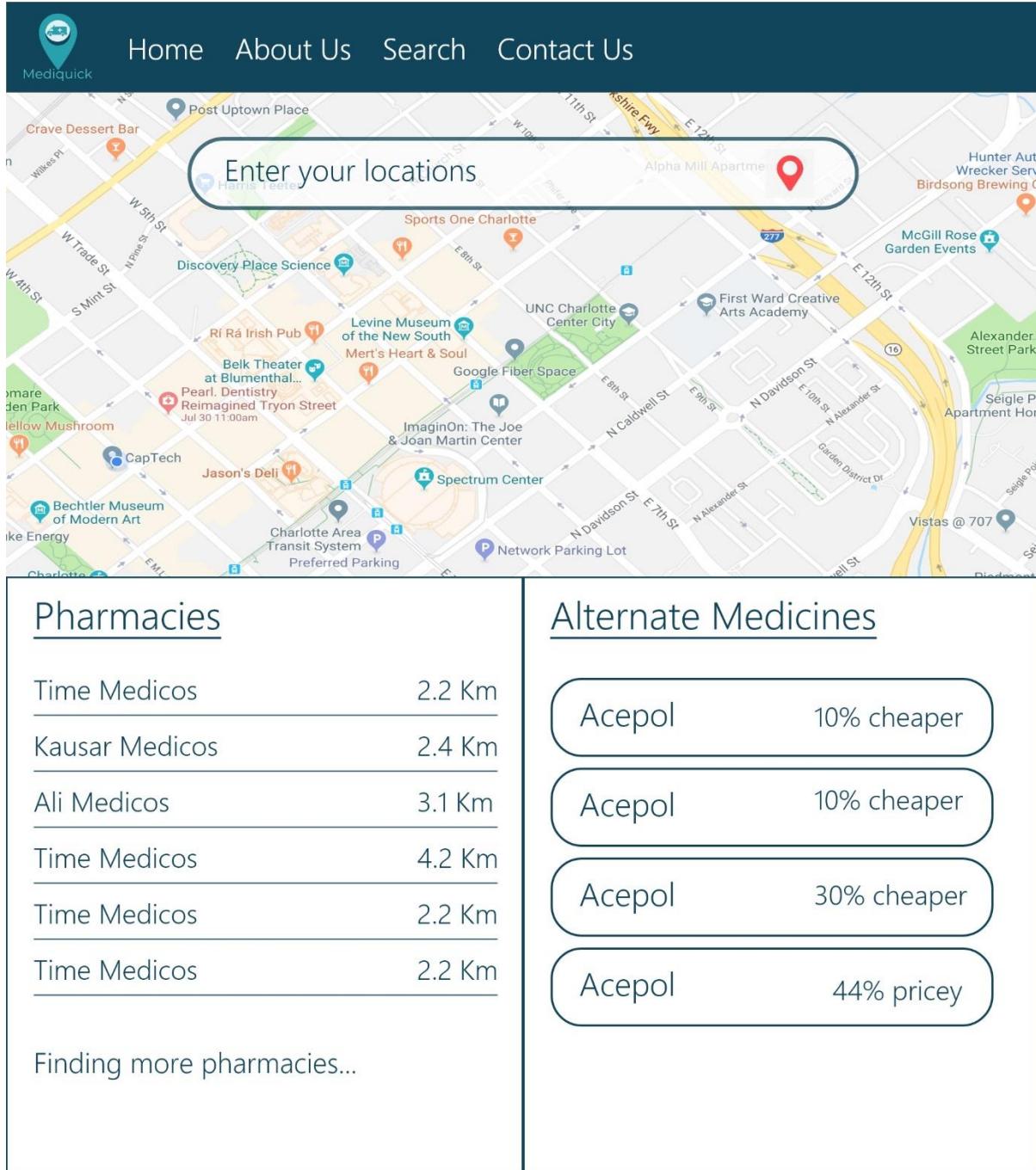


Figure 6.7 – Alternate Medicines



Home About Us Search Contact Us

**Reservation Details**

Benamol      Paracetamol: 500mg

---

Pharmacy: Kausar Medicos, Lucky One Mall

---

Timings: 9:30 am - 11:00 pm

Quantity:

**RESERVE**

Figure 6.8 – Order Reserve



Home About Us Search Contact Us

**Reservation Details**

All	Pending	Expired	Fulfilled	Unfulfilled
44 Adnan Medical Store	SPADIX Tabros Pharma	4mlx6x injections	5 2019-02-13 12:45 am	<b>Fulfilled</b>
44 Kausar Medicos	ANAFORTAN AGP (PVT) LTD	4mlx6x injections	5 2019-04-18 12:45 am	<b>Expired</b>
44 Adnan Medical Store	SPADIX Tabros Pharma	4mlx6x injections	5 2019-11-13 12:45 am	<b>Pending</b>
44 Adnan Medical Store	SPADIX Tabros Pharma	4mlx6x injections	5 2019-02-13 12:45 am	<b>Pending</b>

Figure 6.9 - Reservation Details

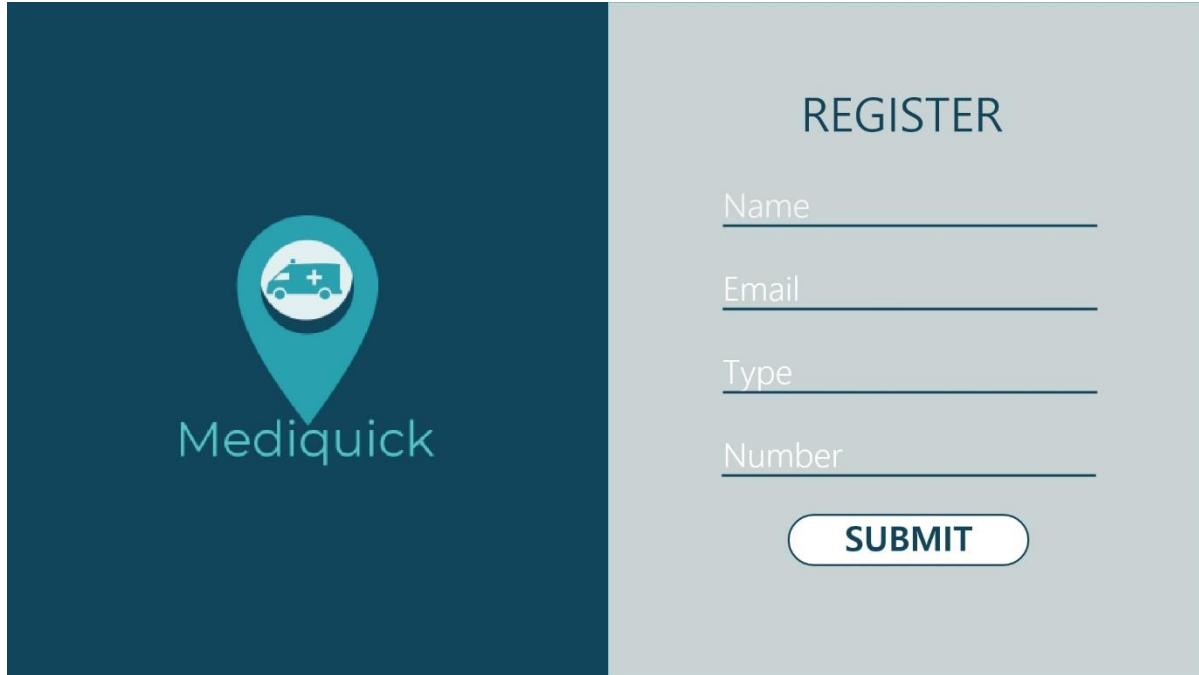


Figure 6.10 – Register

Create Purchase Order <span style="color: #007bff; font-size: 2em;">+</span>				
Purchase Order Number	Distributor Name	Order Amount	Order Date	Status
PO181920313213	TLX Distributor	Rs 70,000	25/12/2019	Fully Delivered
PO181920313213	TLX Distributor	Rs 70,000	25/12/2019	Partially Delivered
PO181920313213	TLX Distributor	Rs 70,000	25/12/2019	Fully Delivered
PO181920313213	TLX Distributor	Rs 70,000	25/12/2019	Fully Delivered
PO181920313213	TLX Distributor	Rs 70,000	25/12/2019	Fully Delivered

Figure 6.11 - Create Purchase Order

**Create PO >> Add**

<b>Basic Info</b>			
Distributer	Company	Status	
Order Date	Order Date Ship	Composition	
<b>Order Line Details</b>			
Product	Ordered Quantity	Price Per Unit	Net Amount
<b>Order Summary</b>			
Total Gross Amount Line Items	Order Net Amount		
GST			

Figure 6.12 - Create Purchase Order

Distributor	Warehouse				
Time Medicos	Karachi Warehouse				
Product Code	Batch Code	Brands	Products	Qty	Expiry Date
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28
CCED01	CCED01-1	Abbot	Augmentan	200	2019/12/28

Figure 6.13 - Create Purchase Order

Medicine		+					
Brand Name	Dosage Form	Manufacturer	Composition	Packaging	Strip per Packet	Price	
Panadol	Tablet	GSK	Paracetamol	500mg	20	Rs. 280	
Panadol	Syrup	GSK	Paracetamol	100ml	-	Rs. 80	
Dispirin	Tablet	Reckitt Benckiser	Aspirin	300mg	60	Rs.600	
Nuberol Forte	Tablet	Searle	Orphenadrine Paracetamol	50mg/650mg	3	Rs. 83.4	
Ponstan Forte	Tablet	Pfizer	Mefenamic Acid	500mg	20	R. 568.5	

Figure 6.14 – Add Medicine

Medicine >> Add

Basic Info	
Brand Name	Composition
Dosage Form	Packaging
Manufacturer	Strips Per Packet
Product Code	Price

Figure 6.15– Add Medicine

Inventory >> Add

**Basic Info**

Batch Code	Expiry Date
Medicine	Price
Quantity	

*Figure 6.16 – Add Inventory Item*

---

# **CHAPTER 7 - SYSTEM ESTIMATES AND ACTUAL OUTCOMES**

---

## **7.1 System Estimates and Actual Outcomes**

This section provides the cost estimates of our application MediQuick.

### ***7.1.1 Historical Data used for estimates***

We used the historical data provided by our industrial collaborator Techno Mites who specializes in developing similar natured products using ASP .NET Core technology stack with MS SQL SERVER database. We included some more complicated features and technology that is of blockchain and implemented our frontend on React which requires more expertise, man efforts and time.

### ***7.1.2 Estimation based on COCOMO II***

COCOMO (Constructive Cost Model) is a model which is used to estimate the cost, effort and schedule while planning any software project. COCOMO model estimates efforts based on the number of lines of code. Program size is expressed in estimated thousands of source lines of code (SLOC or KSLOC).

Effort and Schedule are the key parameters that define the quality of software project.

**Effort:** Amount of labor that will be required to complete a things. It is measured in person-months units.

**Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put.

### 7.1.3 Modes of COCOMOII

COCOMO-II can be applied to the three classes of software project and any of these three forms can be adapted according to the requirements. The hierarchy of these increasingly detailed and accurate modes can be given as:

- Organic projects
- Semi-detached projects
- Embedded projects

**Organic Projects** – A software project is said to be an organic type if the team size required is adequately small with good team experience working with less than rigid requirements, the problem is well understood and the deadline is not tight.

**Semi-detached Projects** – A software project is said to be a semi-detached type if the team size required is more than that of organic type. These are "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop.

**Embedded Projects** – It is a combination of organic and semi-detached projects. Such software projects requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

Our project MediQuick lie in the category of basic COCOMO. Hence the estimated cost of basic COCOMO equation may take the form below:

- 1)  $\text{Effort Applied} = A * (\text{KLOC})^B$  [man-months]
- 2)  $\text{Development Time} = C * (E)^D$  [count]
- 3)  $\text{Staff Size} = (\text{Effort}/\text{Development}) \text{ persons}$  [count]
- 4)  $\text{Productivity} = (\text{KLOC})/\text{Effort}$  [man-months]

Where, KLOC is defined as the estimated number of delivered lines (expressed in thousands) of code for project. The values of the coefficients A, B, C and D for the organic, semi-detached and embedded projects are given in the following table:

Software Project	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table-01 Values of COCOMO Coefficients

Software Size      Sizing Method

SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New <input type="text" value="35000"/>						
Reused <input type="text" value="8000"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="60"/>	<input type="text" value=""/>		
Modified <input type="text" value="6500"/>	<input type="text" value="70"/>	<input type="text" value="80"/>	<input type="text" value="60"/>	<input type="text" value="5"/>	<input type="text" value="50"/>	<input type="text" value="1"/>

Software Scale Drivers

Precedentness <input type="button" value="Low ▾"/>	Architecture / Risk Resolution <input type="button" value="High ▾"/>	Process Maturity <input type="button" value="High ▾"/>
Development Flexibility <input type="button" value="Low ▾"/>	Team Cohesion <input type="button" value="Very High ▾"/>	

Software Cost Drivers

Product	Personnel	Platform
Required Software Reliability <input type="button" value="Very High ▾"/>	Analyst Capability <input type="button" value="High ▾"/>	Time Constraint <input type="button" value="Nominal ▾"/>
Data Base Size <input type="button" value="High ▾"/>	Programmer Capability <input type="button" value="Very High ▾"/>	Storage Constraint <input type="button" value="Nominal ▾"/>
Product Complexity <input type="button" value="Very High ▾"/>	Personnel Continuity <input type="button" value="High ▾"/>	Platform Volatility <input type="button" value="Low ▾"/>
Developed for Reusability <input type="button" value="High ▾"/>	Application Experience <input type="button" value="Low ▾"/>	Project <input type="button" value=""/>
Documentation Match to Lifecycle Needs <input type="button" value="Nominal ▾"/>	Platform Experience <input type="button" value="Low ▾"/>	Use of Software Tools <input type="button" value="Very High ▾"/>
	Language and Toolset Experience <input type="button" value="Low ▾"/>	Multisite Development <input type="button" value="Nominal ▾"/>
		Required Development Schedule <input type="button" value="Low ▾"/>

Maintenance

Software Labor Rates

Cost per Person-Month (Dollars)

Figure-7.1 COCOMO Estimation

---

### Results

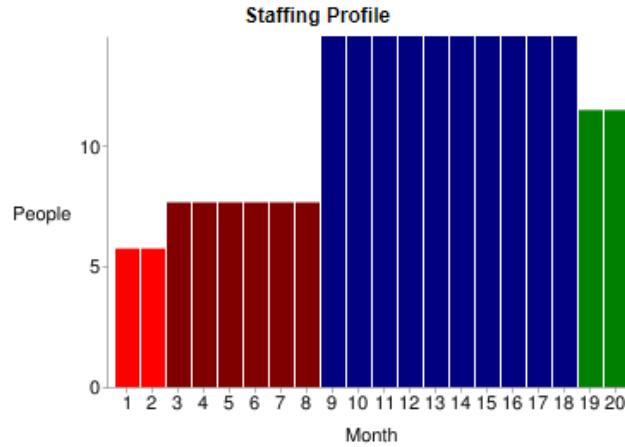
**Software Development (Elaboration and Construction)**

Effort = 207.4 Person-months  
 Schedule = 17.4 Months  
 Cost = \$0

Total Equivalent Size = 44565 SLOC  
 Effort Adjustment Factor (EAF) = 1.21

**Acquisition Phase Distribution**

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	12.4	2.2	5.7	\$0
Elaboration	49.8	6.5	7.6	\$0
Construction	157.6	10.9	14.5	\$0
Transition	24.9	2.2	11.5	\$0


**Software Effort Distribution for RUP/MBASE (Person-Months)**

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	1.7	6.0	15.8	3.5
Environment/CM	1.2	4.0	7.9	1.2
Requirements	4.7	9.0	12.6	1.0
Design	2.4	17.9	25.2	1.0
Implementation	1.0	6.5	53.6	4.7
Assessment	1.0	5.0	37.8	6.0
Deployment	0.4	1.5	4.7	7.5

*Figure-7.2 COCOMO Estimation Results*

## 7.2 Actual Results and Deviation from Estimates

The final cost for the project (at this point in time) is presented here. A deviation chart should be presented to illustrate the comparison between actual cost of system and estimated cost.

The estimates mentioned above are generated considering the ideal scenario but as the scope of our project is limited so there is a huge variation in the results as compared to the estimated results.

The final cost for the project (at this point in time) is presented here. A deviation chart has been presented to illustrate the comparison between actual cost of MediQuick and estimated cost of the MediQuick via COCOMOII.

FACTORS	COCOMO TECHNIQUE RESULTS	ACTUAL RESULTS
Efforts Applied [man-month]	207	220
Development Time [months]	17.4	10

Table-02 Actual and Estimated Results

## 7.3 System Resources (Required and Used)

People, hardware, software, tools, and other resources proposed to build the software are discussed in this section. On the basis of which cost estimations were performed.

### 7.3.1 Man-force Required and Used

5 to 6 members were required for this project each for the front-end development of web app and web-panel, back-end development of admin panels and blockchain integration, testing or quality assurance of project and reporting/documentation. The work would have distributed in the following manner if the required individuals would have been there:

<i>Developer</i>	<i>Work Description</i>
Developer 1	Front-end Developer of Mobile application.
Developer 2	Back-end Developer of Mobile application.
Developer 3	UI/UX Designer
Developer 4	Blockchain Developer
Developer 5	Tester/ Quality Assurance Analyst
Developer 6	Reporting or documentation.

Table 03- Man-force Required

Instead of 6 developers, the system was developed by only three individuals.

### **7.3.2 Hardware Required and Used**

To develop MediQuick system, following hardware were required and used.

<i>Hardware Resource</i>	<i>Description</i>
<i>Device</i>	PC/Laptop, Android Mobile
<i>Operating System</i>	Windows 8, Windows 10
<i>Processor</i>	Core i5 4200U, 1.60 GHz, 2.30 GHz
<i>Memory</i>	4GB , 8GB
<i>Wifi/ Internet</i>	Good Speed

*Table 04- Hardware Required and Used*

### **7.3.3 Software/Tools Required and Used**

To develop MediQuick system, following software/tools were required and used.

<i>Software Resources</i>	<i>Description</i>
<i>Code Editor</i>	Visual Studio, VS Code, Remix IDE
<i>Database</i>	MS SQL SERVER
<i>Testing Tool</i>	Postman for API Testing
<i>Version Control System</i>	Git and BitBucket
<i>Graphical Tools</i>	Star UML, Microsoft Visio, Draw.io
<i>Blockchain Node</i>	Geth Node and Infura
<i>Blockchain Testnet</i>	Ropsten Network
<i>Frontend Testing</i>	Redux DevTools

*Table 05- Software/Tools Required and Used*

---

## **CHAPTER 8 - TEST PLAN**

---

This section describes the overall testing strategy and issues that are required to properly execute effective tests.

Testing is a critical phase in a software's life cycle. A software System is tested in parallel to development and continued to be tested after development and even after deployment system. Therefore, a proper planning has to be done before actually starting the testing the system

Testing is done to ensure the quality of a software by constructing a proper test plan and following it in an organized, efficient and timely manner. Testing enables to identify bugs of higher priorities so that they can be removed and fixed before deploying the software.

A test plan is documented with scope, aims and objective of a software. It contains description of testing approaches, strategies and techniques to be followed and estimates for cost and schedule of testing and the related risks.

A testing procedure consists of the following phases:

- Requirement analysis
- Test planning
- Test case preparation
- Test environment Setup
- Test case execution
- Test closure

### **8.1 System Test and Procedure**

This system describes the overall testing procedure of our software application.

We performed testing in the phases mentioned below:

- The requirements were analyzed in order to get an idea about conflicting, missing and ambiguous requirements and possible testcases were identified.
- GUI of our application was reviewed to test user interface and user experience in terms of ease of use and robustness.

- The registration and authentication functionalities were tested so that users can register on our application without facing any trouble and all types of users (manufacturer, distributor, retailer, admin and end-user) can login to the application by entering valid credentials.
- Management of medicines, inventory items, warehouses, locations and clients modules were tested for CRUD operations. API testing and testing through front-end forms were done.
- Search module was tested to ensure results were being generated
- Testing of Google map was done for proper navigation of medical stores.
- Database was tested if create, update or deleting operations on the data were being saved correctly in a consistent manner in the database and required data fetching with least delay time possible.
- Testing of all other features was done.

## **8.2 Testing Strategy**

The overall strategy for software testing is described.

The testing strategy of our application followed three levels of testing which are as follows:

Unit Testing

Integration Testing

System Testing

### **8.2.1 Unit Testing**

The strategy and brief procedure for unit tested is described.

A software application is tested on unit level during the implementation or development phase i.e. in parallel to coding. Each unit of code is tested separately to verify proper functioning. Both manual and automated testing is done at this level.

Unit under the test, is executed to check if variables are storing and returning correct required values or not to ensure producing of a desired output

In our application ‘MediQuick’, we have performed the unit testing in the similar manner. We have tested the different modules as described below:

#### **8.2.1.1 Registration Module:**

We started testing with this module. Registration module was tested to ensure validations working properly on form fields and user is getting registered on providing valid information. And, finally the registered user is getting stored into our database with password stored as hash.

It was also tested that the system doesn’t allow the user with the same email address to register more than once.

#### **8.2.1.2 Authentication Module:**

Authentication module was tested to ensure that the user is successfully logged into the system on providing valid credentials. On entering one or both of the fields (email and password) incorrect user is not logged in. It was tested that user has option to retry the process.

#### **8.2.1.3 Medicine Management:**

Medicine Management was tested using both API testing and front-end forms to verify the basic CRUD operations were being performed successfully and medicines created or updated were stored in the database. Similarly, on deleting medicine it was removed from the database and correct data was retrieved for viewing.

#### **8.2.1.4 Inventory Management:**

In the similar manner as medicine management, inventory management was tested to verify the basic functionalities for adding new medicine items into inventory, updating the quantity and deleting the record with consistency maintained in the database.

#### **8.2.1.5 Location Management:**

Location management module was also tested for CRUD operations.

#### **8.2.1.6 Search Module:**

Search module was tested to ensure the brand or generic name searched and the brand or generic name of the medicine showed in results were same.

Along with this, alternate medicines proposed were verified to have the same drug composition as the original searched medicine.

#### **8.2.1.7 Locating Stores on Map:**

It was verified that the system is showing the nearest stores for availability of medicines.

It was tested that medical stores were located correctly on the map and the correct time and distance was calculated for the route from given starting point to the destination.

***8.2.1.8 Reserve Order module:***

This module was tested to verify on reserving order, order is created with status set to ‘pending’ and it was listed in order history.

***8.2.1.9 Blockchain module:***

It was checked that blockchain token are generated and transactions are created for the required actions .

***8.2.1.10 Sales/Purchase Order:***

These were tested to verify on placing order, order is created with status set to ‘pending’ , assigned to the receiver and order history is maintained.

***8.2.1.11 Warehouse Management***

This module was tested to verify creation of warehouse. Apart from this, it was verified that bulk upload was successful on uploading excel file. On uploading file with wrong data format or file type , system rejected it.

***8.2.1.12 Transfer Stocks:***

Transfer of stocks module was tested that stocks were transferred to warehouse or distributers.

***8.2.1.13 QR Code Scan***

It was tested that user was able to scan QR Code from the packaging and system was able to read it.

***8.2.1.14 Medicine Authentication from Supply Chain:***

Movement of supply chain of the medicine is tracked for the authentic QR Code. If QR Code is not genuine, system displays the medicine is unauthentic.

This behavior was tested.

***8.2.1.14 Manage distributer module:***

It was verified that a manufacturer was able to select distributers from our system and connect to them. Manufacturer is also able to create new ones.

***8.2.1.15 Admin Dashboard:***

Admin dashboard was tested.

**8.2.1.16 Client Creation Module:**

It was tested that client users (retailer, manufacturer and distributor) are created on system and synced to the database on providing valid information.

**8.2.1.17 Manage Users module:**

Functionalities of updating admin users' information and adding new admin users were tested.

**8.2.1.18 Company Profile:**

It was tested that information is only added once and then can only be edited. New information cannot be created and the present can not be deleted.

**8.2.1.18 Company Documents:**

It was tested that document and images files are easily uploaded and stored. Their name and description can later be updated or deleted. Downloading documents was also tested.

**8.2.1.19 Front-end Module (Web pages and components):**

Front end forms were tested for UI and reading correct input from the user. Navigation among the pages were tested. Responsiveness of the application was tested.

Buttons, clickable icons and dialogues were also tested to be functional.

**8.2.2 *Integration Testing***

Integration testing is done after unit testing when all or few units are integrated. A technique out of three major techniques for Integration testing can be opted. Bottom up, Top-down or hybrid are followed.

If units are working correctly individually, they might not continue to do so when combined with the other modules of the system. To ensure the correct behavior, modules who are directly related in terms of navigation or dependency for data or actions are integrated to combine and then tested. For integration testing of our application, we also combined unit or modules and integrated them to check correct behavior.

We integrated authentication module with the homepage to validate that on entering correct credentials and logging in, user lands on to home page (or dashboard for admin user).

We validated, that on uploading excel files (for bulk upload), file is not only read and stored successfully but also data is pushed to our database, and the uploaded data is visible on our system (to the user having access).

For supply chain related modules, we validated that on performing actions such as Creation of sales/ purchase order or transfer of stocks transactions are actually created on our blockchain.

Integration testing was done for all type of orders and notification generation to verify that on placing order, respective receivers got the notification alert on their homepage.

Integration of QR Code Scanned by user with blockchain was done to test that the code scanned by user is read successfully and then back-tracked on blockchain to display movements of supply chain

### ***8.2.3 Validation Testing***

Validation testing is performed to compare expected and actual results. It is performed hand in hand with development as well as after completion of the project to ensure the output generating is as per the specified or expected behaviour.

It validates that the product offers what it actually intended to.

We have done validation testing during the development for data retrieving operations of our modules. We performed it again once our application was built completely and deployed.

For instance, we validated the alternate medicines returned as a result on Searching a medicine have the exact same drug composition as the one which was searched by the user. Or, the medical stores shown as a nearby store are actually near to user's location.

### ***8.2.4 High-order Testing (System Testing)***

Software is tested on system level to ensure the correct as a whole system as per the requirements specifications (SRS). It is a black box testing where user is unaware of internal details and just test the system for functionalities specified in user requirements.

When performing system testing, we traced the outputs to the business requirements we have defined earlier. It included the testing for the following aspects:

- Major services that are provided to the users (Search medicines, reserve order, verifying medicine authenticity and supply chain management)
- Actual outputs are same as desired for a particular input.
- User experience of our application, aesthetic interface and ease in understanding of use.
- Validation on form fields.
- Confirmation messages for each action performed.

#### ***8.2.4.1 Security Testing***

Security is a type of testing done to ensure there is no point of breaching into the application. Avoiding unauthorized access to crucial data or breaking into the transactions being done on the system.

It's never a good idea to leave an application untested for backdoor entry points.

To achieve security in our system, we implemented JWT Authentication in our application. JSON Web Tokens (JWT's) are used for authorization of users. JWT is an open standard (RFC 7519) that provides a secure method of transmitting data and information between parties as a JSON object. Authentication was tested as per the tokens created.

In-order to protect the user credentials we store user passwords using a one-way cryptographic hash function. If, in case, a hacker is able to hack our database, the person can never get hold of the user credentials because it is impossible to reverse a one-way hash. We tested the hashing implementation to make sure it is working as desired.

#### ***8.2.4.2 Stress Testing***

Stress testing was done for load of number of transactions on blockchain. It was also done for uploading data in bulk using excel file to verify the system is robust enough to process the heavy files with large size /amount of data.

## **8.3 Testing Resources and Staffing**

Apart from the testing tools, no other resources were required such as staff. Team members were sufficient to complete the process of testing.

## **8.4 Test Metrics**

In software testing, Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. In other words, metrics helps estimating the progress, quality and health of a software testing effort.

Following modules of our application are evaluated using standard metrics:

- Authentication Module
- Client Creation Module
- Medicine Management
- Inventory Management
- Location Management
- Search Module
- Locating Stores on Map
- Reserve Order module
- Blockchain module
- Sales/Purchase Order
- Warehouse Management
- Transfer Stocks
- Medicine Authenticity Module
- Manage distributor module
- Admin Dashboard
- Manage Users module
- Location Management
- Company Profile:

## **8.5 Testing Tools and Environment**

Ample of resources/tools are available for manual and automated testing both. For manual testing of REST APIs, we have used Postman, a well-known tools for API creation, management and testing.

For automated testing from a user's perspective, we have used Selenium. For testing different forms such as registration, login or creation forms. We opted this to make the testing process faster in an easy way without the doubt of human errors.

Along with this, we manually tested our application features when time was not enough to write test scripts.

## **CHAPTER 9 - FUTURE ENHANCEMENTS AND RECOMMENDATIONS**

---

We have depicted acute diligence in delivering our proposed system functionalities and features. We covered the highlighting features of our application that is to solve the problem of unavailability of drugs and eradicate the issue of counterfeit drugs through blockchain technology. We have been successful in deploying the solution to the core problems and are now looking forward to implementing the advancements and improving the user experience and underlying software efficiency.

We plan to migrate the blockchain architecture from public blockchain to a private blockchain as Ethereum lacks the performance and efficiency in mining the transactions. The throughput of ethereum transaction is quite low for a use case as complex as that of a medicine supply system. With a private permissioned blockchain architecture, we plan to improve the transactional efficiency of our system.

Furthermore, AI and Data Analytics can be employed to provide users with better notifications and data oriented decision making. This can help bridge the gap between the supply and demand of the medicines.

---

## **CHAPTER 10 - CONCLUSION**

---

The main purpose and motivation of this project was to introduce a game changing technology of blockchain in solving the persistent problem of counterfeit drugs and lack of proper mechanisms of authenticating the supply chain of the purchased pharmaceutical products. Moreover, a centralized platform connecting all the nearby pharmacies to fulfill the immediate drug needs was missing. This software system enables the patients to quickly locate their desired medicines in nearby pharmacies, reducing their efforts and cost of searching.

The challenges encountered were that of choosing the right technology stack for developing such a critical, reliability focused and efficient software system. The recommended technology stack that we employed is quite complex and we had no grasp of it until the start of our project. It was a crucial learning experience and we managed to mitigate all the risks associated with it.

We developed a user friendly application that a person can use to search a medicine, locate a pharmacy nearby and reserve the medicine within seconds. The admin panel abstracts the layman from underlying blockchain implementation and provides a user friendly interface for performing their business activities. On the other hand, our system provides complete audit details for the regulatory authorities for transparency and quality control which serves our purpose.

## A. APPENDICES

### i. Project Schedule

This section gives the overview of project schedule, project milestones, task distribution and the output of project scheduling tool for our application MEDIQUICK.

#### Timeline Chart/ Gantt Chart

## Project Planner



*Gantt Chart of MediQuick Application*

### **Project Group Organization / Workload Distribution**

We had three members in our project so we divided our work evenly so that all the members get equal chance to learn and explore things. The division between members was as following:

TEAM MEMBER	WORK ASSIGNED
Muhammad Usman Sabir	Blockchain architecture designing Smart Contract development Block chain integration Backend architecture designing Database designing Front-end development System integration Research & Meetings
Yusra Wasi	Backend architecture design Backend development Database designing API development and testing Bulk Upload and QR Code Generation Testing Report Writing
Muneeb Ahmed Khan	Front-end architecture designing Front-end development Secure Authentication implementation QR Code Scanning Front-end integration Research & Meetings

## ii. Working Session/ Snapshots of Deployed System

Below attached are the screenshots of MEDIQUICK running application.

The screenshot shows the 'Create Purchase Order' form. On the left, there's a sidebar with 'MANAGEMENT' and 'Warehouse Management' sections, including 'Warehouses'. The main area has a title 'Create Purchase Order'. It includes fields for 'ABC' (dropdown), 'Select Manufacturer' (dropdown), 'Status' (dropdown set to 'Created'), and date pickers for 'Order Date' (18/Sep/2020) and 'Shipping Date' (18/Sep/2020). Below these is a section for 'Order Line Details' with a '+' button. A message box says 'You have no selected medicines. Please select medicines by clicking on the add icon'. At the bottom right is a 'Submit' button.

*Create Purchase Order*

This screenshot is similar to the first one but includes a modal calendar overlay. The calendar shows the month of September 2020, with the date 'Fri, Sep 18' highlighted in blue. The rest of the interface is identical to the first screenshot, with the 'Create Purchase Order' form and its various fields.

*Create Purchase Order*

Warehouse	Code	Lat	Long	Address	City	State	Country	Actions
ABC Wre...	ABCWH	22.4	67.5	Gulshan	Karachi	Sindh	Pakistan	
ABC Wre...	ABCWH2	44.2	65.4	Highway	Hyderabad	Sindh	Pakistan	

Add Warehouse

Add City

Address

Name \_\_\_\_\_ Code \_\_\_\_\_

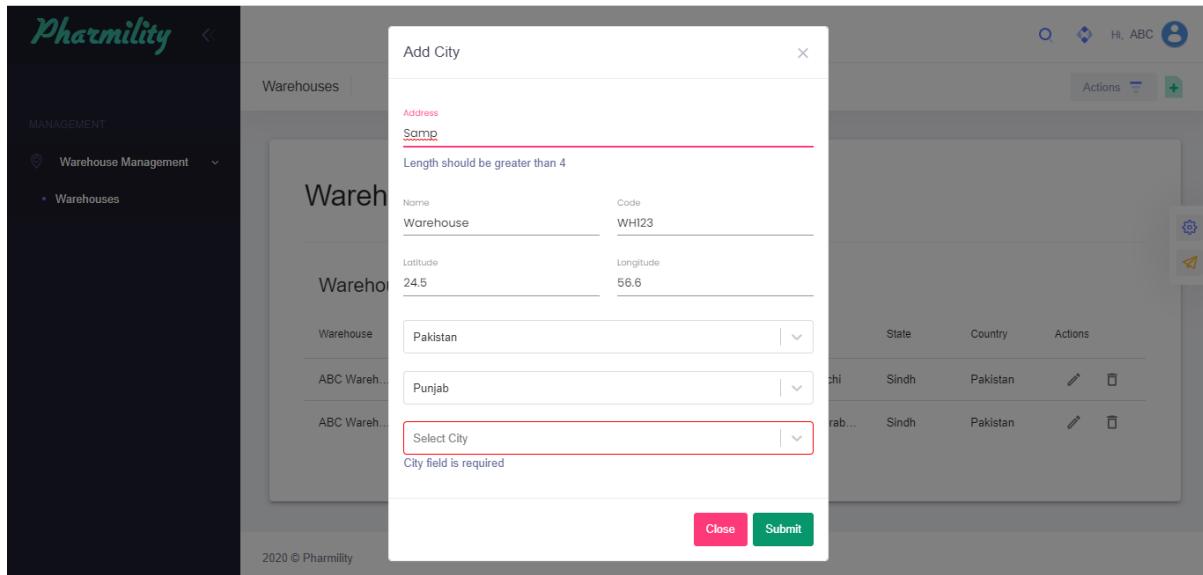
Latitude \_\_\_\_\_ Longitude \_\_\_\_\_

Select Country

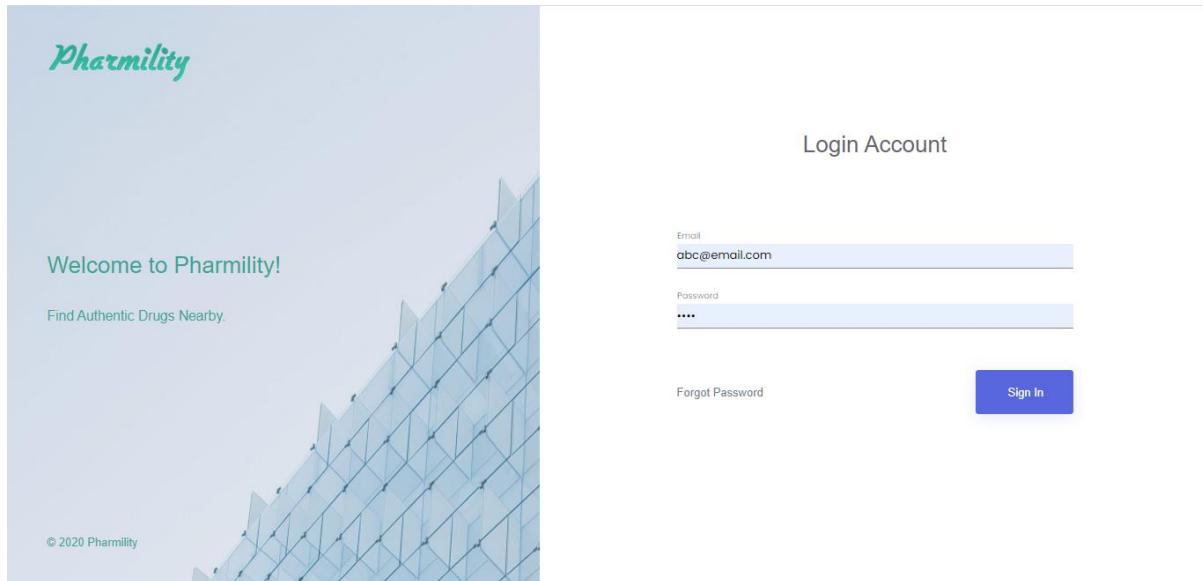
Select Province

Select City

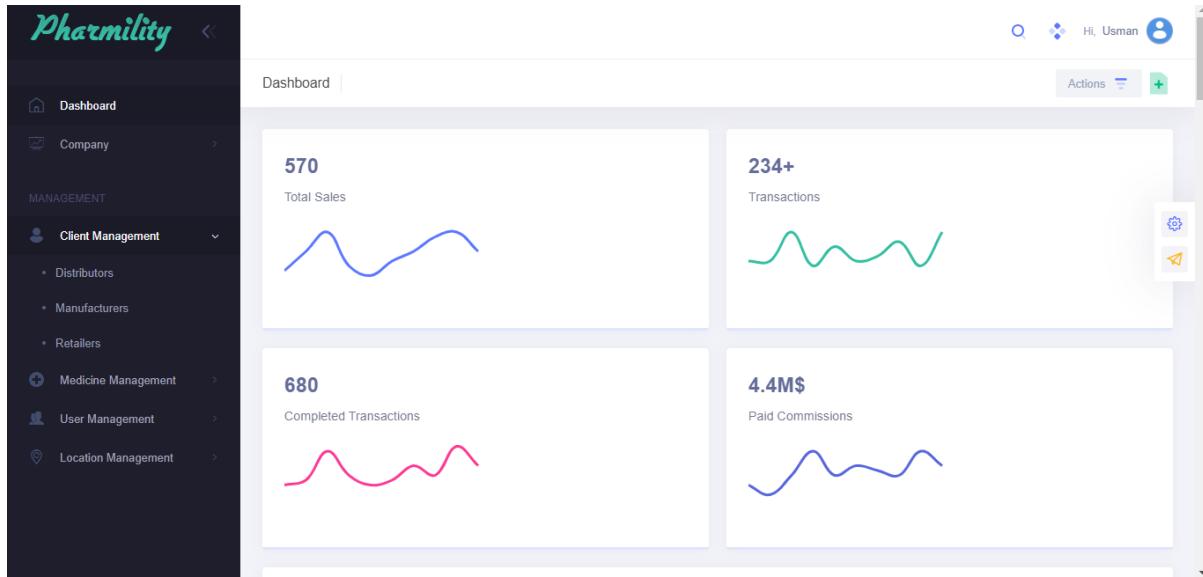
Add Warehouse



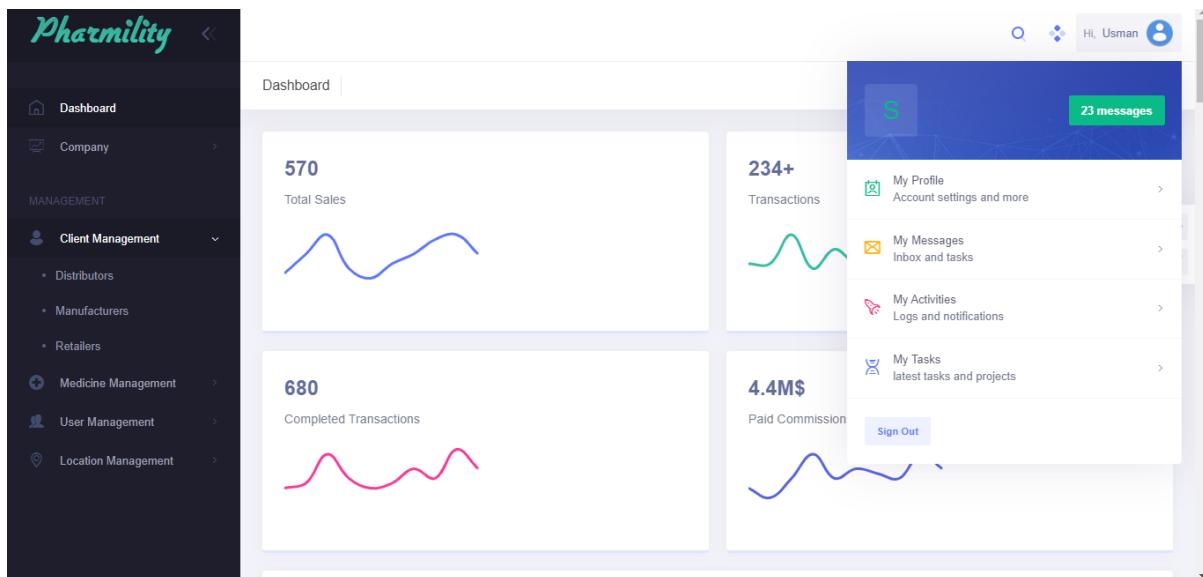
*Add Warehouse*



*Login Account*



*Admin Dashboard*



*Admin Dashboard*

The screenshot shows the 'Company Profile' section of the Pharmility application. The left sidebar has a dark theme with the 'Company' section expanded, showing 'Profile' and 'Documents'. The main content area has a light background with a header 'Company Profile' and the 'Pharmility' logo. It displays the following information in a grid:

Company Name Pharmility	Company TRN No 123456464778975
Company License No 123456789012346	License Expiry 25/Nov/2020
Company Address Karachi Pakistan	
Company Contact 1 03030388456	Company Contact 2 03348289867

Actions buttons are visible at the top right.

### Company Profile

The screenshot shows the 'Company Profile' section of the Pharmility application. The left sidebar has a dark theme with the 'Company' section expanded, showing 'Profile' and 'Documents'. The main content area has a light background with a header 'Profile' and the 'Pharmility' logo. It displays the following information in a grid:

Company Address Karachi Pakistan	
Company Contact 1 03030388456	Company Contact 2 03348289867
Company Email Address contact@pharmility.com	Company Website Address www.pharmility.com
Latitude 43.21	Longitude 55.3

An 'EDIT' button is located at the bottom right of the form area. The footer shows '2020 © Pharmility'.

### Company Profile

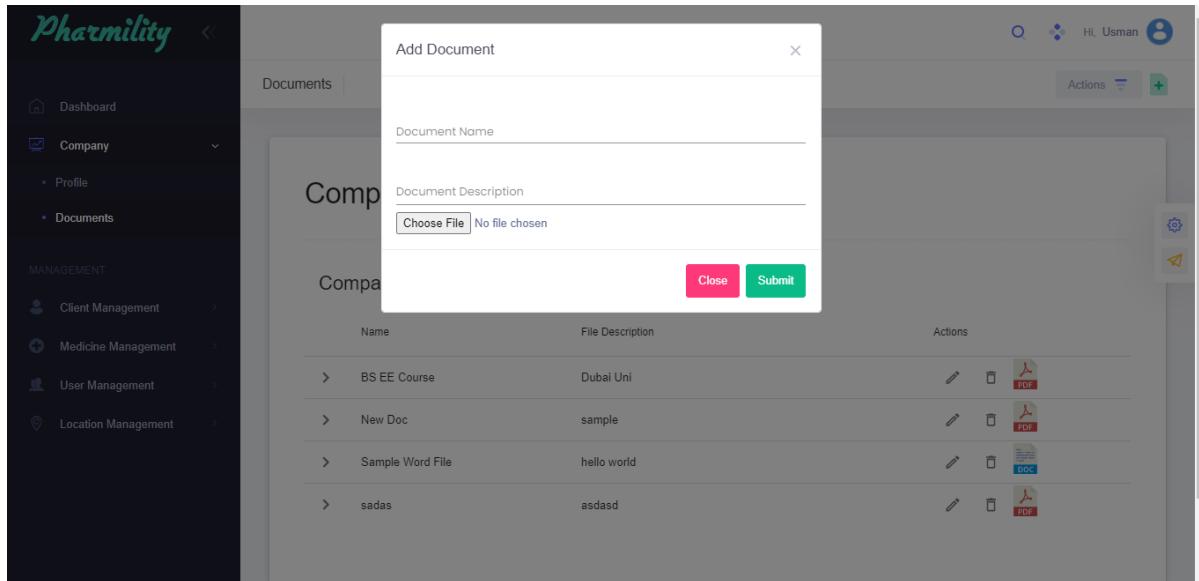
The screenshot shows the 'Company Profile' section of the Pharmility application. The left sidebar has 'Company' selected under 'Profile'. The main content area displays the 'Company Profile' heading and a summary card with the Pharmility logo. Below this, there are four data fields: 'Company Name' (Pharmility), 'Company TRN No.' (123456464778975), 'Company License No.' (123456789012346), and 'License Expiry' (25/Nov/2020). Further down are 'Company Address' (Karachi Pakistan), 'Company Contact 1' (03030388456), and 'Company Contact 2' (03348289867). A 'Remove' button is located at the top right of the summary card.

*Company Profile*

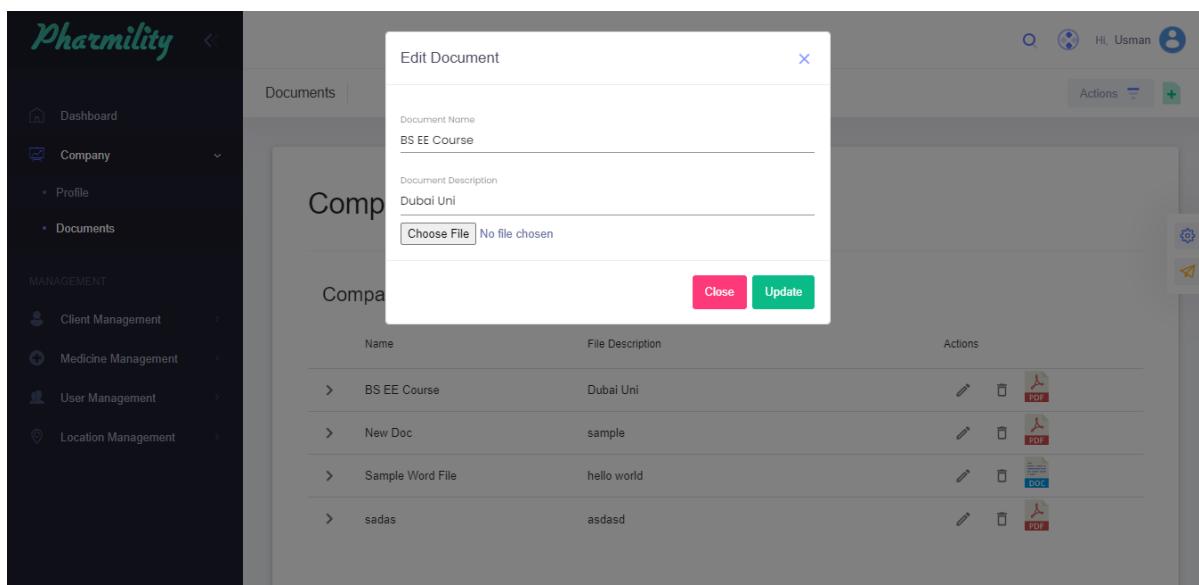
The screenshot shows the 'Company Documents' section of the Pharmility application. The left sidebar has 'Company' selected under 'Documents'. The main content area displays the 'Company Documents' heading and a table. The table has columns for 'Name', 'File Description', and 'Actions'. It contains four rows:

Name	File Description	Actions
BS EE Course	Dubai Uni	[Edit] [Delete] [PDF]
New Doc	sample	[Edit] [Delete] [PDF]
Sample Word File	hello world	[Edit] [Delete] [DOC]
sadas	asdasd	[Edit] [Delete] [PDF]

*Company Documents*



*Company Documents*



*Company Documents*

The screenshot shows the 'Company Documents' section of the Pharmility application. On the left, there's a dark sidebar with the 'Pharmility' logo at the top. Below it are sections for 'Dashboard', 'Company' (with 'Profile' and 'Documents' sub-options), and 'MANAGEMENT' (with 'Client Management', 'Medicine Management', 'User Management', and 'Location Management'). The main content area has a title 'Company Documents' with a blue '+' button. A modal window titled 'Delete Document?' is open, asking 'Are you sure you want to delete this document?'. It contains 'NO' and 'YES' buttons. Below the modal is a table with columns 'Name' and 'Actions'. The table data is as follows:

Name	Actions
BS EE Course	[Edit] [Delete] [PDF]
New Doc	[Edit] [Delete] [PDF]
Sample Word File	[Edit] [Delete] [DOC]
sadasd	[Edit] [Delete] [PDF]

*Company Documents*

The screenshot shows the 'Distributors' section of the Pharmility application. The left sidebar is identical to the one in the previous screenshot. The main content area has a title 'Distributors' with a blue '+' button. Below it is a section titled 'Distributors List' with a table. The table has columns 'Name', 'Email', 'Contact No.', 'City Name', and 'Actions'. The data is as follows:

Name	Email	Contact No.	City Name	Actions
ABC	abc@email.com	03333122112	Karachi	[Edit] [Delete]
XYZ	xyz@email.com	03333148012	Lahore	[Edit] [Delete]

*Add Distributor*

**Add Distributor**

Distributor Name	Contact Number	City
ABC	Distributor Address	Distributor License Number
XYZ	Distributor TRN Number	
License Expiry 18/Sep/2020		

**Actions**

City Name	Actions
Karachi	
Lahore	

**Close** **Submit**

### Add Distributor

**Manufacturers**

Name	Email	Contact No.	Actions
GSK	admin@gsk.com		
Pfizer	admin@pfizer.com	03333148012	

**Actions**

### Add Manufacturer

**Add Manufacturer**

Name	Contact No.	Actions
GSK	33148012	
Pfizer		

### Add Manufacturer

**Retailers**

Name	Email	Contact No.	City Name	Actions
Time Medicos	admin@timemedicos.com	03333148282	Karachi	

### Add Retailer

**Edit Retailer**

Retailer Name: Time Medicos

Email: admin@timemedicos.com

Contact Number: 0333148282

City: Karachi

Retailer Address: Gulshan

Latitude:

Longitude:

Retailer License Number: 213123211231231

Retailer TRN Number: 213123211231231

License Expiry: 25/Sep/2020

**Actions**

City Name	Actions
Karachi	

**Close** **Update**

*Add Retailer*

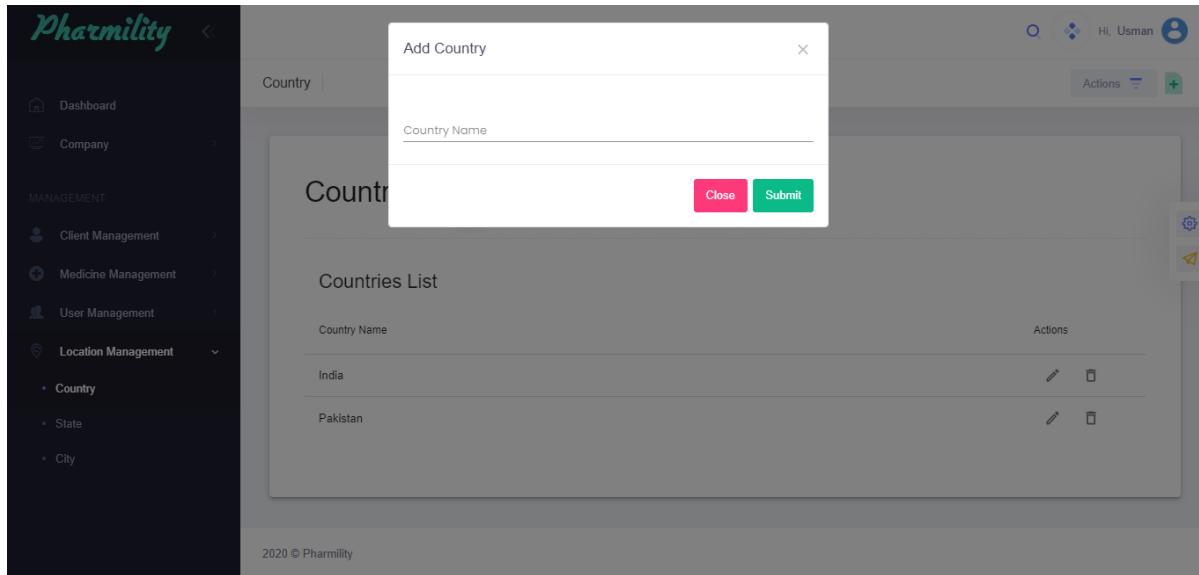
**Countries**

**Countries List**

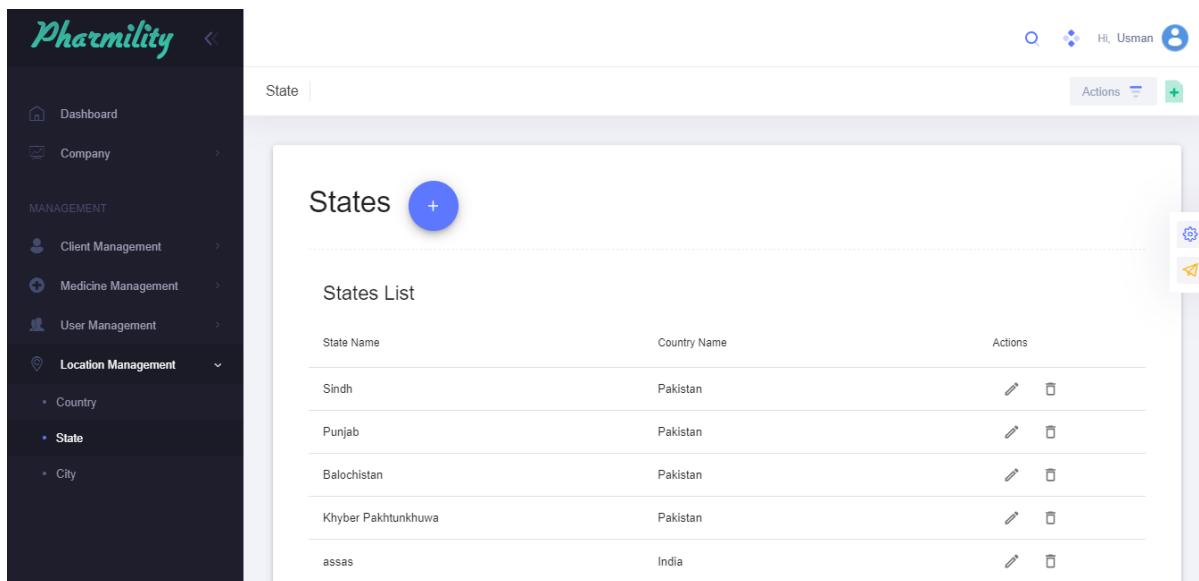
Country Name	Actions
India	
Pakistan	

**Actions**

*Add Country*



Add Country



Add State

The screenshot shows the Pharmility application interface. On the left, there is a dark sidebar with navigation links: Dashboard, Company, MANAGEMENT (Client Management, Medicine Management, User Management), and Location Management (Country, State, City). The main content area has a title 'State' and a sub-section 'States'. A modal window titled 'Add State' is open, showing a dropdown for 'Country' set to 'Pakistan' and a text input field for 'State Name' containing 'assas'. Below the modal is a table titled 'States List' with columns 'State Name', 'Country Name', and 'Actions'. The table contains five rows: Sindh (Pakistan), Punjab (Pakistan), Balochistan (Pakistan), Khyber Pakhtunkhuwa (Pakistan), and assas (India).

State Name	Country Name	Actions
Sindh	Pakistan	[Edit] [Delete]
Punjab	Pakistan	[Edit] [Delete]
Balochistan	Pakistan	[Edit] [Delete]
Khyber Pakhtunkhuwa	Pakistan	[Edit] [Delete]
assas	India	[Edit] [Delete]

Add State

The screenshot shows the Pharmility application interface. The sidebar is identical to the previous screenshot. The main content area has a title 'City' and a sub-section 'Cities'. A modal window titled 'Cities' is open, showing a table titled 'Cities List' with columns 'City Name', 'State Name', 'Country Name', and 'Actions'. The table contains four rows: Karachi (Sindh, Pakistan), Hyderabad (Sindh, Pakistan), Islamabad (Punjab, Pakistan), and Lahore (Punjab, Pakistan).

City Name	State Name	Country Name	Actions
Karachi	Sindh	Pakistan	[Edit] [Delete]
Hyderabad	Sindh	Pakistan	[Edit] [Delete]
Islamabad	Punjab	Pakistan	[Edit] [Delete]
Lahore	Punjab	Pakistan	[Edit] [Delete]

Add City

The screenshot shows the Pharmility application interface. On the left is a dark sidebar with navigation links: Dashboard, Company, MANAGEMENT (Client Management, Medicine Management, User Management, Location Management - Country, State, City), and Warehouse Management. The main area has a title 'City' and a sub-section 'Cities'. A modal window titled 'Add City' is open, showing a dropdown for 'Country' set to 'Pakistan' with a dropdown menu listing Sindh, Punjab, Balochistan, and Khyber Pakhtunkhwa. Below this is a 'City Name' input field and two buttons: 'Close' (red) and 'Submit' (green). In the background, a table titled 'Cities List' displays four rows of data: Karachi (Sindh, Pakistan), Hyderabad (Sindh, Pakistan), Islamabad (Punjab, Pakistan), and Lahore (Punjab, Pakistan). Each row has edit and delete icons in the 'Actions' column.

Add City

The screenshot shows the Pharmility application interface. The sidebar includes: Dashboard, MANAGEMENT (Medicine Management - Medicine, Brand, Manufacturer, Generics, Dosage Form), and Warehouse Management. The main area has a title 'Medicine' and a sub-section 'Medicines'. A large blue circular button with a '+' sign is visible. Below it is a section titled 'Manage Medicines' with a search bar. A table lists two medicine entries: Buscopan (Created At: 2020-07-29, Brand Name: Buscopan, Dosage Name: Drops, Manufacturer: Pfizer, Drugs: Montelukast: 50mg, Packing: 100ml, Code: EXL10) and Exlant (Created At: 2020-08-24, Brand Name: Exlant, Dosage Name: Tablets, Manufacturer: Pfizer, Drugs: Montelukast: 20mg, Packing: 10s, Code: EXL10). The table has columns for Created At, Brand Name, Dosage Name, Manufacturer, Drugs, Packing, Code, and Actions. At the bottom, there are pagination controls for 'Rows per page: 10' and '1-2 of 2'.

Add Medicine

Add Medicine

Pfizer Select Brand

Select Dosage Form Packing

GTIN Code

Drugs +

Select Drug Strength Remove

Close Submit

### Add Medicine

Add Medicine

Pfizer Buscopan

Drops Packing  
Packing field is required

GTIN  
GTIN field is required

Code  
Code field is required

Drugs +

Select Drug Strength Remove  
Drug field is required Strength field is required

Close Submit

### Add Medicine

The screenshot shows the Pharmility software interface. The left sidebar has a dark theme with the logo 'Pharmility'. The 'Management' section under 'Medicine Management' includes 'Brand' which is currently selected. Other options like 'Medicine', 'Manufacturer', and 'Generics' are also listed. The main content area is titled 'Medicine Brands' and shows a list of brands with columns for 'Creation Date' and 'Name'. Three entries are visible: Arinac (17 Mar 2020, 12:00:00 AM), Exlant (17 Mar 2020, 12:16:44 PM), and Buscopan (17 Mar 2020, 1:02:48 PM). A search bar and actions buttons are at the top right of the list.

Creation Date	Name	Actions
17 Mar 2020, 12:00:00 AM	Arinac	
17 Mar 2020, 12:16:44 PM	Exlant	
17 Mar 2020, 1:02:48 PM	Buscopan	

### Add Medicine Brands

The screenshot shows the Pharmility software interface. The left sidebar has a dark theme with the logo 'Pharmility'. The 'Management' section under 'Medicine Management' includes 'Generics' which is currently selected. Other options like 'Medicine', 'Brand', 'Manufacturer', and 'Dosage Form' are also listed. The main content area is titled 'Medicine Generics' and shows a list of generics with a column for 'Name'. Five entries are visible: Paracetamol, Montelukast, Ibuprofen, testing, and asdadas. The entry 'Montelukast' is highlighted with a gray background. A search bar and actions buttons are at the top right of the list.

Name	Actions
Paracetamol	
Montelukast	
Ibuprofen	
testing	
asdadas	

### Add Medicine Generics

The screenshot shows the Pharmility software interface. On the left, there is a dark sidebar with the Pharmility logo at the top. Below it are navigation items: Dashboard, MANAGEMENT (with sub-options: Medicine, Brand, Manufacturer, Generics, and Dosage Form), and Warehouse Management. The 'Dosage Form' option under MANAGEMENT is currently selected. The main content area has a title 'Medicine Dosage Forms' with a blue '+' button. Below this is a section titled 'Dosage Forms' with a table:

Name	Actions
Tablets	
Drops	
Capsules	
Syrup	
Susensions	

*Add Medicine Dosage Forms*

The screenshot shows the Pharmility software interface. On the left, there is a dark sidebar with the Pharmility logo at the top. Below it are navigation items: Dashboard, MANAGEMENT (with sub-options: Medicine Management and Warehouse Management), and Warehouse Management (with sub-options: Warehouses and Batch). The 'Batch' option under Warehouse Management is currently selected. The main content area has a title 'Batch Upload' with a blue '+' button. Below this is a section titled 'Batches' with a table:

BatchNo	Medicine Code	Manufacturing Date	Expiry Date	Actions
B11	AETest1	12 Jul 2019, 12:00:00 AM	12 Jul 2020, 12:00:00 AM	
B12	AETest2	12 Jul 2019, 12:00:00 AM	12 Jul 2020, 12:00:00 AM	
B13	AETest3	12 Jul 2019, 12:00:00 AM	12 Jul 2020, 12:00:00 AM	
B14	AETest4	12 Jul 2019, 12:00:00 AM	12 Jul 2020, 12:00:00 AM	
B15	AETest5	12 Jul 2019, 12:00:00 AM	12 Jul 2020, 12:00:00 AM	

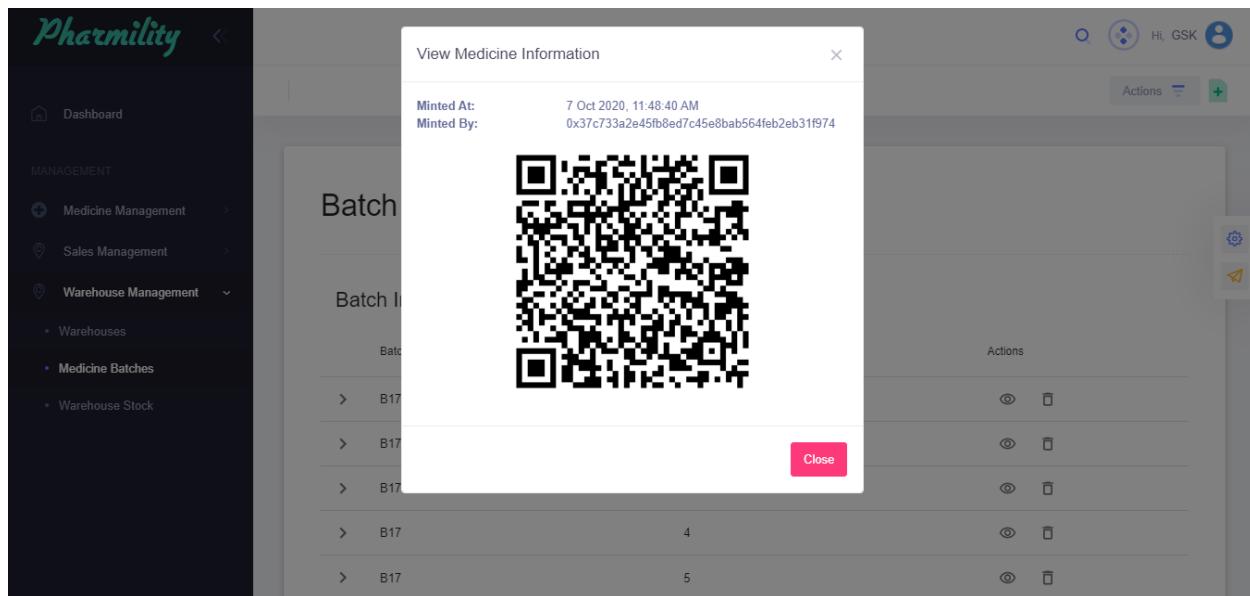
*Batch Upload*

The screenshot shows the Pharmility software interface. On the left, there is a dark sidebar with the Pharmility logo at the top. Below it are navigation items: Dashboard, MANAGEMENT (with Medicine Management and Warehouse Management sub-items), Warehouses, and Batch. The main content area has a title "Batch" and a sub-section "Batch Upload". A modal window titled "Add Excel File" is open, containing a "Choose File" button and a message "No file chosen". Below the modal is a table titled "Batches" with columns: BatchNo, Medicine Code, Manufacturing Date, Expiry Date, and Actions. The table contains five rows, each with a BatchNo from B11 to B15, a Medicine Code starting with AEtest, and manufacturing and expiry dates in July 2019 and 2020 respectively. Each row has edit and delete icons in the Actions column.

*Batch Upload*

The screenshot shows the Pharmility software interface. The sidebar on the left is identical to the previous screenshot. The main content area has a title "Batch Medicines" and a sub-section "Batch Info". A table titled "Batch Info" is displayed with columns: Batch No, Serial No, and Actions. The table contains five rows, all with a Batch No of B11 and Serial Nos 1 through 5. Each row has edit and delete icons in the Actions column.

*Batch Medicines*



*Batch Information – Token Information From Blockchain and QR Code*

ID	Manufacturer	Distributor	Order Date	Ship Date	Status	Actions
21	GSK	ABC	10/7/2020	10/7/2020	Delivered	
20	GSK	ABC	10/7/2020	10/7/2020	Delivered	
19	GSK	ABC	10/7/2020	10/7/2020	Delivered	
18	GSK	ABC	10/7/2020	10/7/2020	Delivered	
17	GSK	ABC	10/7/2020	10/7/2020	Delivered	

*Purchase Orders*

The screenshot shows the Pharmility software interface. On the left is a dark sidebar with the Pharmility logo at the top. Below it are navigation items: Dashboard, MANAGEMENT (with Medicine Management, Sales Management, and Warehouse Management), and a central Sales Management section with Purchase Orders and Deliveries. The main area is titled "Edit Purchase Order". It displays form fields for "ABC" and "GSK" in dropdown menus, and "Status" set to "Created". Below these are "Order Date" (04/Oct/2020) and "Shipping Date" (10/Apr/2020). A table titled "Order Line Details" shows one row: Batch B11, Brand Arinac, Dosage Tablets, Packing 10s, Mfg 7/12/2019, Expiry 7/12/2020, Qty 3, Deliver 3, and Action with a pencil icon.

### Fulfill Purchase Orders

The screenshot shows the Pharmility software interface with the "Add Products" dialog box open. The sidebar on the left is identical to the previous screenshot. The dialog box has a search bar and a table with columns for "Batch Code" and "Serial Numbers". There are five rows: the first is not checked; the second, third, and fourth are checked and highlighted in purple, all labeled "B11"; the fifth is not checked and labeled "B11". At the bottom of the dialog are buttons for "Close" and "Save Changes". Below the dialog, a preview table shows the selected products: B11, Arinac, Tablets, 10s, 7/12/2019, 7/12/2020, 3, 3, and Action with a pencil icon. A "Create Delivery" button is visible at the bottom right of the dialog.

### Fulfill Purchase Orders – Select Medicines

Purchase Order Deliveries

ID	Supplier	Recipient	Order Date	Ship Date	Status
20	GSK	ABC	10/7/2020	10/7/2020	Delivered
19	GSK	ABC	10/7/2020	10/7/2020	Delivered
18	GSK	ABC	10/7/2020	10/7/2020	Delivered
17	GSK	ABC	10/7/2020	10/7/2020	Delivered
16	GSK	ABC	10/6/2020	10/6/2020	Delivered
15	GSK	ABC	10/5/2020	6/10/2020	Delivered
13	GSK	ABC	10/4/2020	5/10/2020	Delivered
12	GSK	ABC	10/4/2020	4/10/2020	Delivery Cre...
11	GSK	ABC	10/4/2020	4/10/2020	Delivery Cre...
10	GSK	ABC	10/4/2020	4/10/2020	Delivery Cre...

### Receive Deliveries – Distributors

Create Sales Order

Product	Retailer	Status
ABC	Select Retailer	Created

Order Date: 09/Oct/2020      Shipping Date: 09/Oct/2020

Selected Medicines

Batch	Brand	Dosage	Packing	Mfg	Expiry	Qty
B11	Arinac	Tablets	10s	7/12/2019	7/12/2020	0
B12	Buscopan	Drops	20ml	7/12/2019	7/12/2020	0
B16	Arinac	Tablets	10s	7/12/2019	7/12/2020	0

### Create Sales Orders – Distributors

**Edit Purchase Order**

Batch	Brand	Dosage	Packing	Mfg	Expiry	Qty	Deliver	Action
B11	Arinac	Tablets	10s	7/12/2019	7/12/2020	1	0	

**Create Delivery**

### Fulfill Sales Orders - Retailers

**Products**

Time Medicos

You have no selected medicines. Please select medicines by clicking on the add icon

Activate Windows  
Go to Settings to activate Windows.

### Inventory Management

Medicine ID	Brand Name	Manufacturer Name	Dosage Form	Packing
4	Panadol	Abbott Pharma	Tablets	10
6	Panadol	Abbott Pharma	Injections	ml
7	Rigix	GetsPharma	Tablets	10
8	Rigix	GetsPharma	Tabs DS	10
9	Myteka	Abbott Pharma	Drops	Bottle

Showing 1 to 5 of 8 Results

0 1 Next

Close Save Changes

### *Inventory Management*

Medicine ID	Brand Name	Manufacturer Name	Dosage Form	Packing
4	Panadol	Abbott Pharma	Tablets	10
6	Panadol	Abbott Pharma	Injections	ml
9	Myteka	Abbott Pharma	Drops	Bottle

Showing 1 to 3 of 3 Results

0

Close Save Changes

Activate Windows  
Go to Settings to activate Windows.

2020 © Pharmility

### *Inventory Management*

Medicine ID	Brand Name	Manufacturer Name	Dosage Form	Packing
4	Panadol	Abbott Pharma	Tablets	10
6	Panadol	Abbott Pharma	Injections	ml
7	Rigix	GetsPharma	Tablets	10
8	Rigix	GetsPharma	Tabs DS	10
9	Myteka	Abbott Pharma	Drops	Bottle

Showing 1 to 5 of 8 Results

**Actions** +

**Close** **Save Changes**

Activate Windows  
Go to Settings to activate Windows.

### Inventory Management

Medicine	Manufacturer	Generics	Qty
Panadol Tablets, 10	Abbott Pharma	Paracetamol : 500mg	0
Panadol Injections, ml	Abbott Pharma	Paracetamol : 500mg	0
Rigix Tabs DS, 10	GetsPharma	Anastrazole : 50mg Folic Acid : 60mg	0

**Actions** +

**Submit**

Activate Windows  
Go to Settings to activate Windows.

### Inventory Management

The screenshot shows the Pharmility web application. On the left, a dark sidebar titled 'MANAGEMENT' has a 'Inventory Management' section with 'Products' and 'Inventory' options. The main area is titled 'Inventory' and shows a table of inventory items. The table columns are: Medicine, Manufacturer Name, Generics, Quantity, and Actions. The data in the table is as follows:

Medicine	Manufacturer Name	Generics	Quantity	Actions
Panadol Tablets, 10	Abott Pharma	Paracetamol : 500mg	100	<a href="#">Edit</a> <a href="#">Delete</a>
Panadol Injections, ml	Abott Pharma	Paracetamol : 500mg	10	<a href="#">Edit</a> <a href="#">Delete</a>
Rigix Tablets, 10	GetsPharma	Baclofen : 50mg	10	<a href="#">Edit</a> <a href="#">Delete</a>
Rigix Tabs DS, 10	GetsPharma	Anastrazole Folic Acid : 50mg : 60mg	0	<a href="#">Edit</a> <a href="#">Delete</a>
Myteka Drops, Bottle	Abott Pharma	Meltdormin Clonidine : 50mg 100mg	0	<a href="#">Edit</a> <a href="#">Delete</a>

A blue circular button with a '+' sign is located at the top center of the table area. In the bottom right corner of the table, there is a message: 'Activate Windows' and 'Go to Settings to activate Windows.' There are also gear and search icons on the right side of the table.

### Inventory Management

This screenshot is similar to the one above, showing the 'Inventory Management' section. A modal window titled 'Enter Medicine Quantity' is open in the center. It has a single input field labeled 'Quantity' containing the value '0'. At the bottom of the modal are two buttons: 'Close' (pink) and 'Submit' (green). The background table data is identical to the first screenshot.

### Inventory Management

## Orders

<b>Time Medicos</b>	Stadium Road 2020-09-20T00:00:00 Panadol, Tablets	Qty: 10
<b>Maseeha</b>	Block-7, Gulshan-e-Iqbal 0001-01-01T00:00:00 Panadol, Tablets	Qty: 5
<b>Maseeha</b>	Block-7, Gulshan-e-Iqbal 2020-09-20T00:00:00 Panadol, Tablets	Qty: 2
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-09-20T00:00:00 Buscopan, Tablets	Qty: 2
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-01T00:00:00 Myteka, Drops	Qty: 1
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 0
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 10
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 56
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 5
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 1
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 2
<b>Maseeha</b>	Block-7, Gulshan-e-Iqbal 2020-10-04T00:00:00 Rigix, Tabs DS	Qty: 1
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-05T00:00:00 Buscopan, Tablets	Qty: 2

## Orders

Orders		
<b>Time Medicos</b>	Stadium Road 2020-09-20T00:00:00 Panadol, Tablets	Qty: 10
<b>Maseeha</b>	Block-7, Gulshan-e-Iqbal 0001-01-01T00:00:00 Panadol, Tablets	Qty: 5
<b>Maseeha</b>	Block-7, Gulshan-e-Iqbal 2020-09-20T00:00:00 Panadol, Tablets	Qty: 2
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-09-20T00:00:00 Buscopan, Tablets	Qty: 2
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-01T00:00:00 Myteka, Drops	Qty: 1
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 0
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 10
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 56
<b>Al-Waheed Pharmacy</b>	Paradise bakery, Block-3 2020-10-04T00:00:00 Buscopan, Capsules SR	Qty: 5

## Orders

The screenshot shows the Pharmility website's search interface. At the top, there is a navigation bar with links for Home, About, Search, Order History, Contact, and Logout. Below the navigation bar, the word "Pharmility" is displayed in a large, bold, black font. A search bar is present with the placeholder text "Search Medicine" and a dropdown menu showing "Paracetamol". To the right of the search bar is a green search button with a magnifying glass icon. Below the search bar, there are two radio buttons: "Brand" (unchecked) and "Generic" (checked). The background of this section has a gradient from light green at the top to light blue at the bottom.

This screenshot displays detailed product information for Panadol Extra. It includes sections for Tablets (10 tablets, Paracetamol: 500mg) and Injections (MI, Paracetamol: 500mg). The background is white with a light gray header and footer.

## Search Medicine

The screenshot shows a map of Karachi, Pakistan, with a red marker indicating the location of Gulshan-e-Iqbal. The map includes labels for various neighborhoods like Orangi Town, North Nazimabad, and Jinnah International Airport. A search bar at the top of the map allows users to enter a location. The background is a standard Google Maps interface.

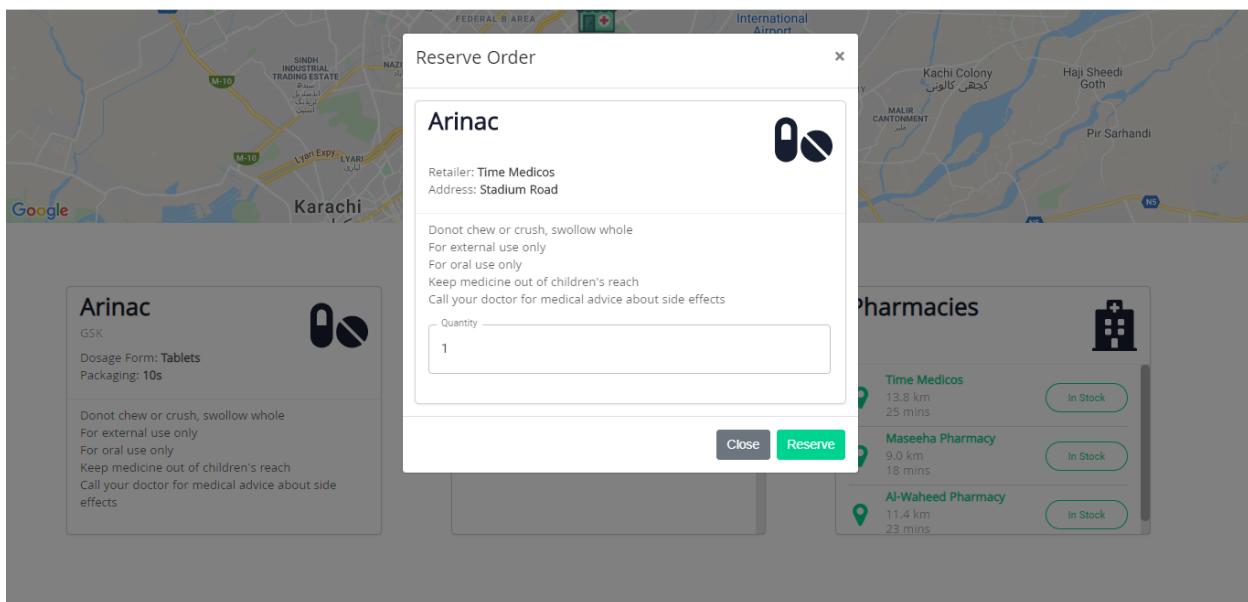
This screenshot shows product details for Panadol Extra. It includes the brand name "GetsPharma", dosage form "Tablets", packaging "10", and instructions: "Don't chew or crush, swallow whole. For external use only. For oral use only. Keep medicine out of children's reach. Call your doctor for medical advice about side effects." There is also a small image of two tablets.

This screenshot shows product details for Alternate Medicines. It includes the brand name "Panadol Abbott Pharma" and a small image of two tablets.

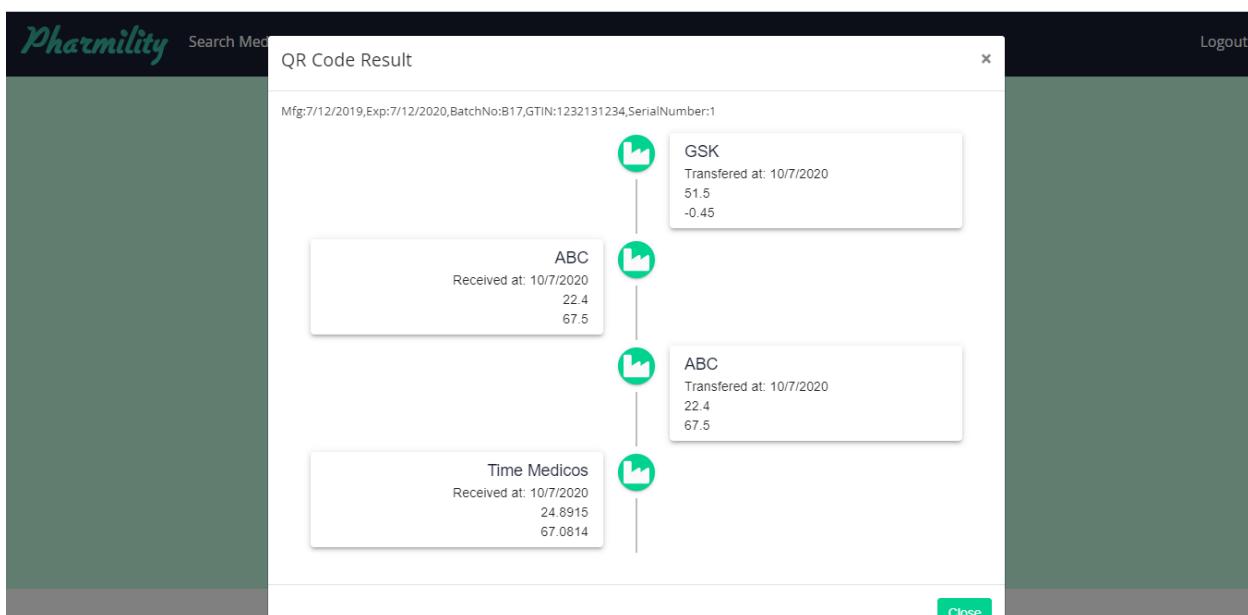
This screenshot lists nearby pharmacies with their names, distances, and delivery times. The data is as follows:

Pharmacy	Distance	Delivery Time	Status
Maseeha	2.8 km	8 mins	In Stock
Time Medicos	9.0 km	17 mins	Out of Stock
Al-Waheed Pharmacy	3.6 km	10 mins	In Stock

## Search Medicine Results



### Search Medicine Results



*QR Code Scanner – Supply Chain Authentication of Medicine*