PHARMACEUTICAL MANAGEMENT SYSTEM

Project report in partial fulfillment of the requirement of Software Engineering Lab
In
COMPUTER SCIENCE & ENGINEERING DEPT.

Submitted By

Jyoti Agarwal
University Roll No. 12018009019060
Section: CSE 3H, Class Roll No.: 28

Department of Computer Science,
UEM, Kolkata

Kinjal Raykarmakar
University Roll No. 12018009019439
Section: CSE 3H, Class Roll No.: 29

Department of Computer Science,
UEM, Kolkata

Mainak Bhowmick
University Roll No. 12018009019562
Section: CSE 3H, Class Roll No.: 30

Department of Computer Science,
UEM, Kolkata

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA
University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700 156

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this project is to develop an effective Pharmaceutical Management System. This system can store data and enable functionality that organizes and maintains the medication use process within pharmacies. There are multiple user levels server various sections like inventory management and medicine production.

1.2 Document Conventions

The headings have Times typeface of various font sizes ranging between 14pt to 18pt. The higher the font size, the higher the heading stands in the document hierarchy. The body has the Arial typeface unform at 11pt font size. Deeply nested subheadings are in a text style of strong and text in italics denotes emphasis.

1.3 Intended Audience and Reading Suggestions

This document is basically intended for a particular type of audience, consisting of new developers, project managers, customers, users. This documentation will also be useful for system and database administrators for understanding the workflow. Additionally, this documentation might serve as a manual to various end users too.

1.4 Project Scope

The scope of this project is to develop a pharmaceutical management system. Under this project, we have designed a web page which will mainly consist of three sections – the vendor section, the pharmacy section, and the customer section. The vendor section deals with the raw materials, chemicals and other items required for the production of the drugs and medicines. The pharmacy section deals with the manufacturing, packing and production of the drugs and medicines. The customer section deals with the sale and marketing of the medicines produced. The customers may consist of the individual buyers or patients, the medicine shopkeepers, the chemists, and druggists, hospitals and clinics, etc.

1.5 References

Google. (n.d.). *Material Design*. Retrieved from Material Design: https://material.io/design

Onuiri, E., & Sam-David, C. (2016, 05). Online Pharmaceutical Management System. *European Scientific Journal, 12*, 1857-7881. doi:10.19044/esj.2016.v12n12p139

PHP-FIG. (n.d.). *PHP Standards Recommendations*. Retrieved from The PHP Framework Interop Group: https://www.php-fig.org/psr/

W3C. (n.d.). *HTML 5.2*. Retrieved from World Wide Web Consortium (W3C): https://www.w3.org/wiki/The_web_standards_model_-_HTML_CSS_and_JavaScript

2. Overall Description

2.1 Product Perspective

This product is a completely new self-contained product. This product has been developed and designed in the time when online marketing, online business, online shopping, online banking and various other online daily-life activities are on the rise. Since it's the high time for these online businesses, so we have decided to come up with an online pharmaceutical management system, for the ease of use and greater smoothness of various sections of people including customers, manufacturers, vendors, retailers and other concerning working entities.

2.2 Product Features

Important features of this project:

- Data Retention: The data entered is retained in a secure and high responsive MariaDB Database using optimized SQL Queries. Same works for data fetching.
- User Management: Multiple classes of users can be managed at a time, thus making the system scalable.
- Inventory Management: In-built inventory management for chemicals and medicines. Keeps track of the quantity and lets the administrator produce medicines depending upon the availability.
- Transactions: Generates reports for customer and vendor transactions.

2.3 User Classes and Characteristics

This product utilizes three classes of users as described below: -

1. **Administrator**: An Administrator is a user type which will manage and integrate the whole product. Being the highest security level and to protect data access and manipulation, administrators cannot be added to the system through the UI. However, an administrator can login to the system and view the admin dashboard through the universal login system. An Administrator has full rights to the following actions:
    i. Add/Edit/Delete Chemicals
    ii. Add/Edit/Delete Medicines
    iii. Set/Unset Medicines for sale
    iv. Add/Edit/Delete Medicine Composition
    v. Trigger Medicine manufacture process and control quantity
    vi. View Vendor/Customer Transactions
2. **Vendors**: Vendors are generic users with no direct modification rights. However, their access level is slightly higher than that of Customers. Vendors can register themselves to the system by filling up the registration form. Vendors *should* login to view the list of chemicals to supply. They can just select and set quantity of chemicals to sell.

3. **Customers**: Customers are generic users too. Customers can register themselves to the system by filling up the registration form. However, customers can list available medicines by just visiting the homepage. To buy medicine(s), they need to log in with their credentials.

2.4 Operating Environment

- MySQL DB
- Client/Server system
- Operating System: Windows, OSX, Linux
- JS Enabled Web Browser

2.5 Design and Implementation Constraints

This project uses MySQL/MariaDB Database. Hence, the queries are solely written on SQL and are expected to run on any other distributions of SQL like MS SQL, SQL Server etc. Though it is unexpected to port this system to a NoSQL or a document-based DB system, primarily due to the existing DB structure. In such a case, the composite tables to link different tables should be avoided.

Keeping in mind the security considerations, Administrator cannot be directly added through the UI. However, by carefully modifying the server-side code in the register page, one can add Administrators. This is done, keeping in mind the consistency in the password hashing algorithm. If Administrators are added directly in the DB with a password, that might not match the hashing done throughout the system, therefore blocking the user from logging in.

This project was developed using Visual Studio Code with PHP IntelliSense. Throughout the code, 2 spaces indentation was maintained. Lines are no more than 80 characters.

2.6 User Documentation

Little or no user documentation is expected to be delivered along with this project. Most of the project will be self-explanatory, backed up with suitable prompts throughout the UI.

Administrators might need additional specialization using the software, which can be achieved through training and evaluation.

Following are the default credentials used during the development phase. The organizations might want to truncate the users table before starting their operations with this system.

User Type	*Email ID*	*Password*
Administrator	admin@pharma.com	admin
Vendor	vendor@pharma.com	vendor
Customer	customer@pharma.com	customer

2.7 Assumptions and Dependencies

It is assumed that an Administrator of the system has the sole authority to manage the whole application. Being the highest access level, an Administrator can control all the production of medicines within the application, even if they are/aren't produced physically in the organization.

This project is known to have no dependencies. It can run as a stand-alone software as long as enough resources are available and the file system has enough permissions to read/write.

3. System Features

The following System features are organized by the different user classes explained in Section 2.3

3.1 Administrator

3.1.1 Description and Priority

The administrator has the highest privilege and the highest priority in the system. An administrator can integrate the whole manufacturing process in a pharmaceutical organization through the system. The project provides a UI to the Administrator to manage the Chemicals and Medicines Inventory and control production.

3.1.2 Stimulus/Response Sequences

User Actions	***System Response***
Add a chemical	The system will accept input through a form and save the chemical data in the DB
Edit a chemical	The system will dynamically create a form using the current chemical data from the DB. Editing and submitting the form will lead to overwriting of the chemical data in the DB.
Delete a Chemical	The system will delete the chemical from the DB.
Add a medicine	The system will accept input through a form and save the medicine data in the DB
Edit a chemical	The system will dynamically create a form using the current medicine data from the DB. Editing and submitting the form will lead to overwriting of the medicine data in the DB.
Delete a Chemical	The system will delete the medicine from the DB.
Add composition	The system will accept input through a form and save the composition data in the DB. Also, the medicine price can be set through here
Edit	The system will dynamically create a form using the current

Composition	composition data from the DB. Editing and submitting the form will lead to overwriting of the composition data in the DB.
Delete Composition	The system will delete the composition from the DB.
Manufacture	The system prompts the administrator with the current availability of the chemicals required to manufacture a certain medicine. If the chemicals aren't available sufficiently, then the administrator cannot proceed with production.
View Transaction	The system will list all the vendor and customer transactions along with the total debit/credit

3.1.3 Functional Requirements

REQ-1: A user should be signed in as an Administration to use all the above features.
REQ-2: The chemical and medicine should be added before adding a composition linking them
REQ-3: Medicines using a chemical in its composition should be deleted before the chemical can be deleted.
REQ-4: Images should be reuploaded in edit forms
REQ-5: The chemical quantity required to produce a medicine should be available sufficiently to manufacture.

3.2 Vendor

3.1.1 Description and Priority

The vendor is a generic user with no direct privileges to edit the DB. Vendors can only list, view and sell chemicals stored in the DB.

3.1.2 Stimulus/Response Sequences

User Actions	***System Response***
List chemicals	All the chemicals stored in the DB will be listed along with the current quantity in the inventory
Sell a chemical	A vendor can select the quantity of chemical to sell

3.1.3 Functional Requirements

REQ-1: A user should be signed in as a Vendor to list and sell chemicals.
REQ-2: A vendor can sell maximum of 10 units of a chemical in a single transaction.

3.3 Customer

3.1.1 Description and Priority

The customer is a generic user with no direct privileges to edit the DB. Customers can only list, view and buy medicines stored in the DB.

3.1.2 Stimulus/Response Sequences

User Actions	***System Response***
List medicines	All the medicines stored in the DB will be listed along with the current quantity in the inventory. Note: Any end user can list medicines even without registering/logging in in the system.
Buy a medicine	A customer can select the quantity of medicines to buy

3.1.3 Functional Requirements

REQ-1: A user should be signed in as a Customer to buy chemicals

4. External Interface Requirements

4.1 User Interfaces

The UI of the total project is based on Material design through the usage of Materialize CSS framework. Distinctive shades of blue is used throughout the project to specify different states of interfaces. Full screen backdrop Modals are used to provide error messages display.

4.2 Hardware Interfaces

Standard Input devices like Keyboard and Mouse pointing device along with output Display device (monitor) with a minimum resolution of 1024×768 should be use. This project doesn't directly interface with any specialized hardware.

4.3 Software Interfaces

This project is a web application and was developed and debugged throughout using the Chrome Web Browser (v 90.0.4430.72). So, this project is intended to run in a web browser (preferably a chromium-based browser). However, using the project through ElectronJS might have undesirable behavior.
PHPMyAdmin using MySQL/MariaDB was used throughout the development process for the backend server and DB requirements.

4.4 Communications Interfaces

This project is indented to run of a web browser with JS enabled. Disabling JS is not recommended as it might lead to non-functioning components. The User Interface is served on the web over HTTP. All the electronic forms follow the HTTP standard POST request to send information to the server. The forms which include some file uploads are encoded as multipart/form-data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

This project works almost in real-time due to less to almost no complex system overheads. Data transferred over local HTTP is fast enough not to include any pre-loaders. However, images might need additional time to load depending on their size or hardware/software read/write times. Reading and Writing on the DB is real-time enabled by optimized queries and indexed fields.

5.2 Safety Requirements

Ideally, one Administrator should be logged in to the system, which can modify crucial DB entries. Multiple Administrators accessing the same data might be protected by the DB's internally locking system. Other safety measures should include protecting the user's DB from multiple email entries, which ideally should be incorporated with error prompts in the UI.

5.3 Security Requirements

Data should be protected with the utmost priority, and user credentials is one of the data leakages sources. Hence, passwords of the users should be hashed before saving in the DB, to protect from logging in as in imposter by the DB admin or DB user(s).

5.4 Software Quality Attributes

This product has a great degree of reliability, robustness and accuracy. Data has been presented in a simple way to the user, which facilitates easy and quick usage. Also, this application is in line with WAI-ARIA, that makes it more accessible (though aria-* labels aren't explicitly declared, modern HTML5 can already use the ARIA from attributed specified in the tags). Additionally, this software is highly maintainable due to the modular coding approach.

6. Other Requirements

Most of the features in this project are re-usable. This includes the authentication system, Database CRUD applications, sidebar navigation etc.
New sections pertinent to this system might include a session-based cart system and payment gateways integrations, auto-generated mailing system etc.

Other requirements for developing this application:

- Text Editor – For coding the application
- Materialize CSS – Material Design themes and responsive CSS for the website
- StarUML – Preparing the design documents

Appendix A: Glossary

Glossary of Terms used in this SRS Documentation:

- **Administrator** – The user intended to work with the backend of the system. Manages and controls the DB and medicine production.
- **Customers** – The user intended to work with a part of the frontend. Customers can buy medicines from the firm.
- **Organization** – The term ‘organization’ has been used interchangeably with ‘firm’, ‘company’, and ‘pharmaceutical company’ which collectively stands as the group of people that use the backend of the system.
- **Project** – The term ‘project’ has been used interchangeably with ‘product’, ‘software’, ‘system’ and ‘application’ which collectively stands as the outcome of development.
- **User** – End user i.e., the customer, unless otherwise specified.
- **Vendors** – The user intended to work with a part of the frontend. Vendors can sell medicines to the firm.

Acronym used in this SRS Documentation:

- **ARIA** – Accessible Rich Internet Application
- **DB** – Database
- **HTML** – Hyper Text markup Language
- **JS** – JavaScript
- **NoSQL** – Not Only SQL
- **PHP** – Hypertext Preprocessor
- **SQL** – Structured Query Language
- **UI** – User Interface

Appendix B: Analysis Models and Design Documents

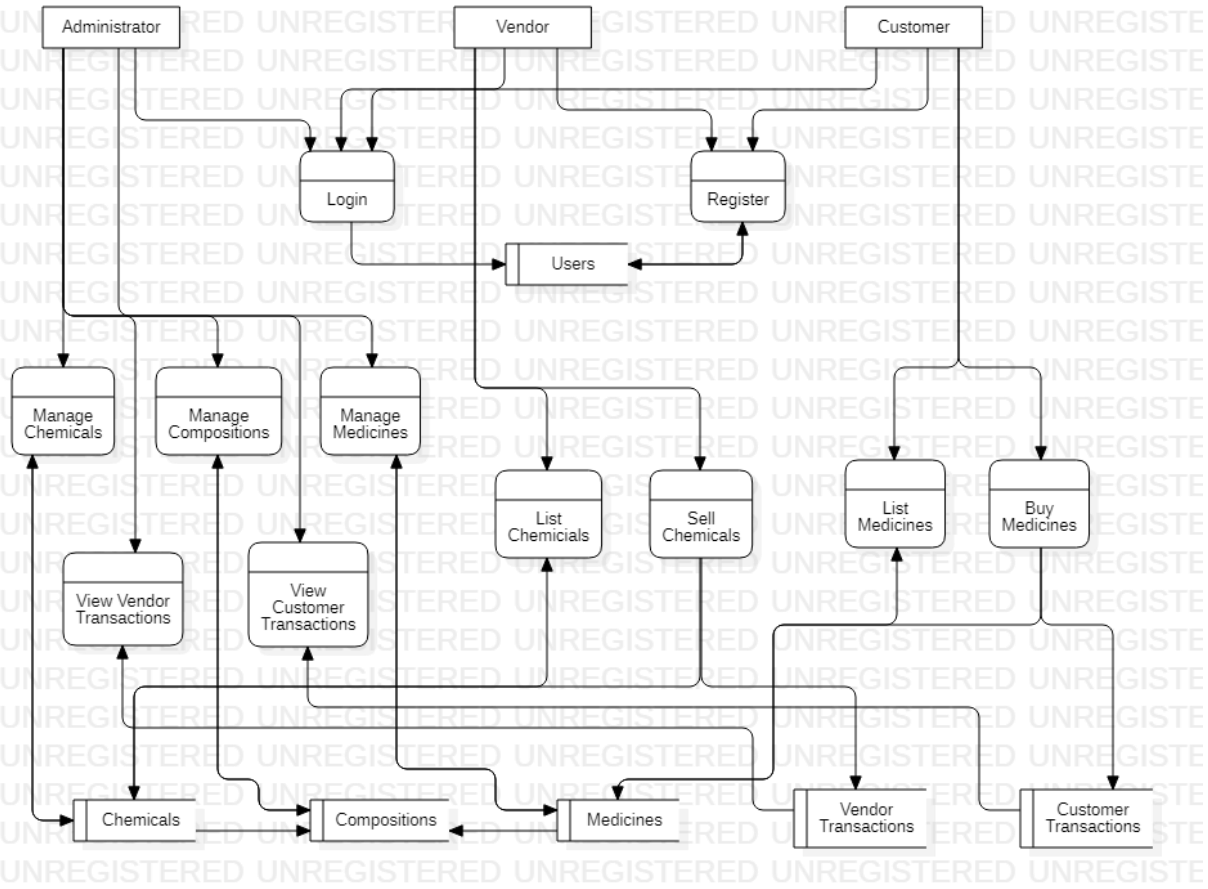

Figure 1 Data flow Diagram

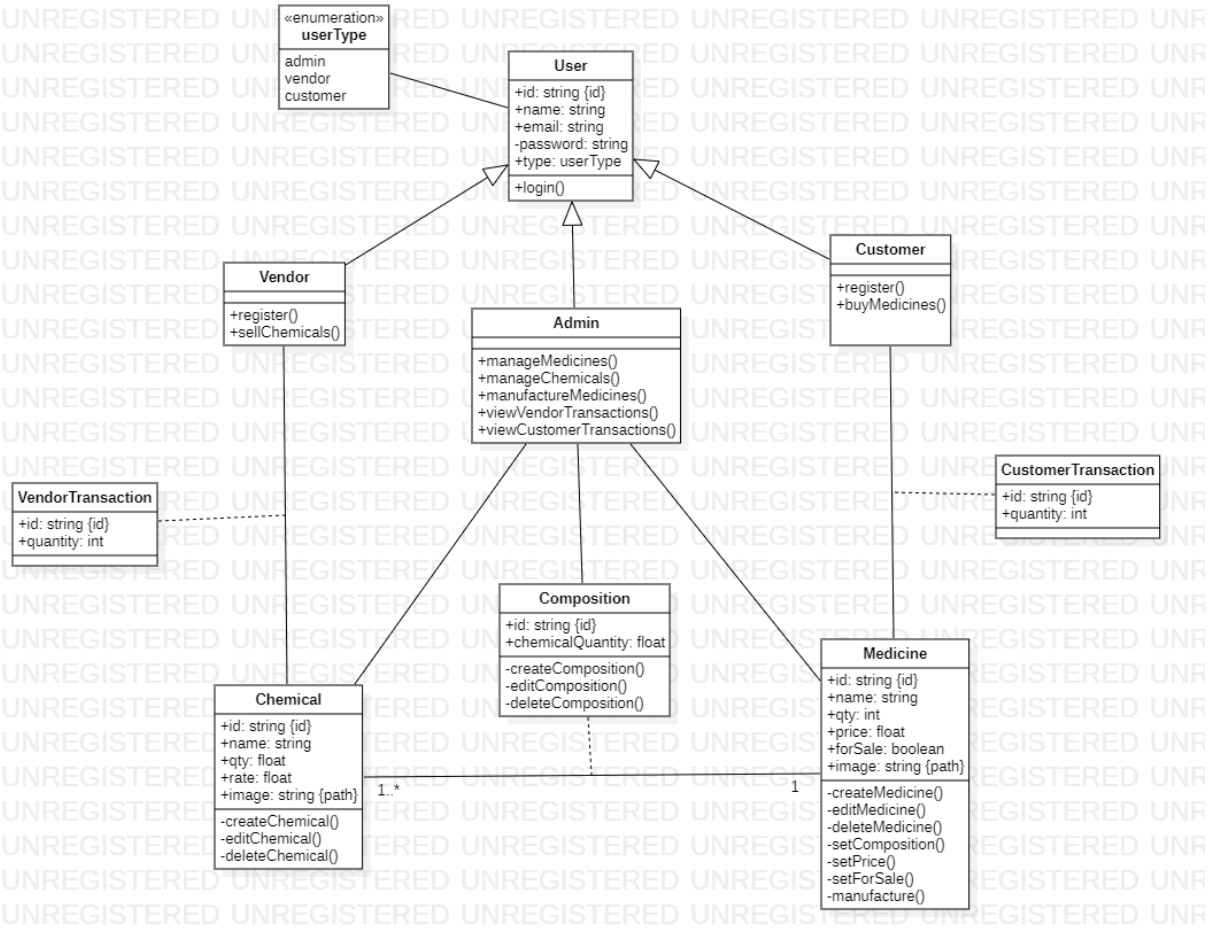

Figure 2 Class Diagram

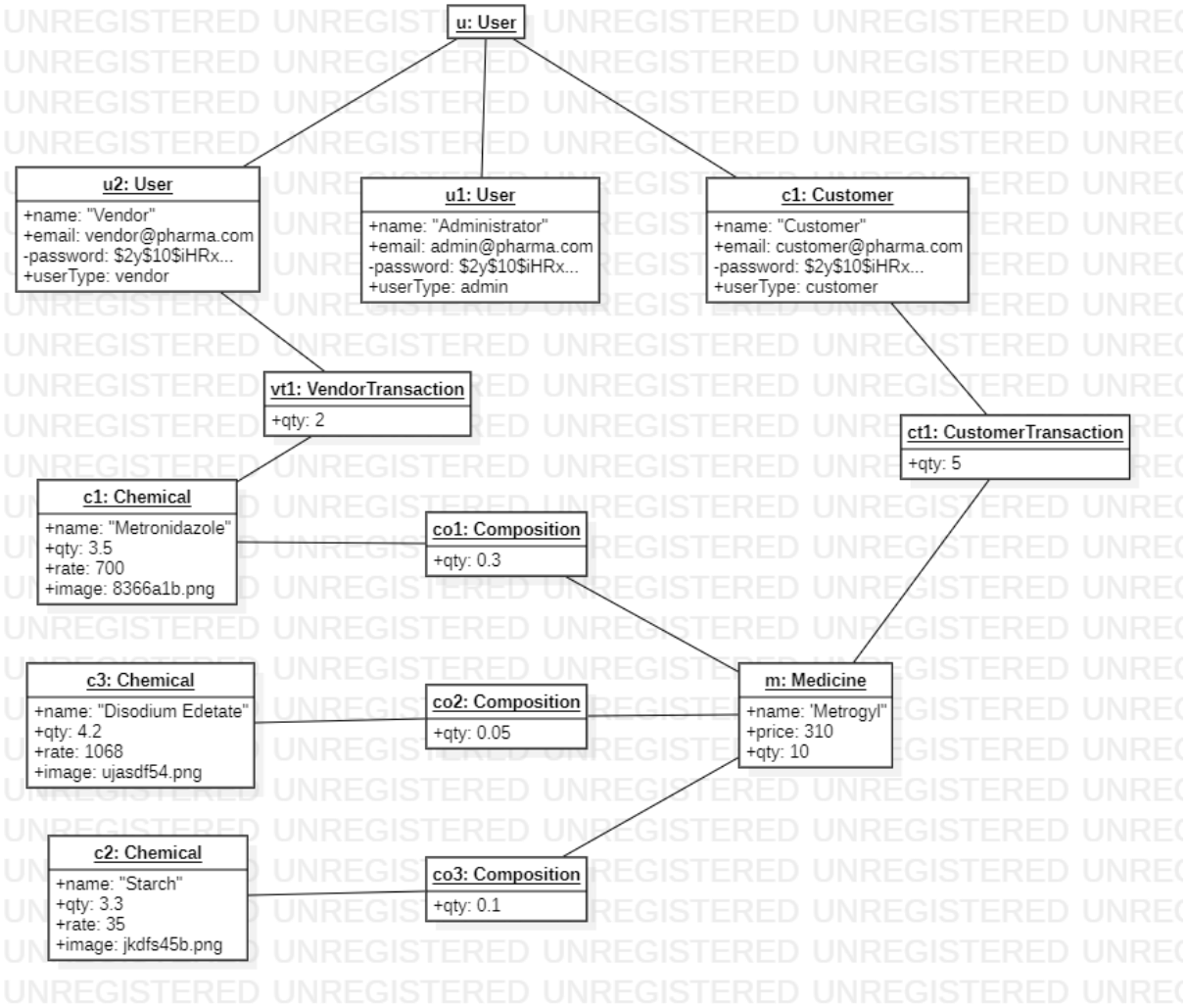

Figure 3 Object Diagram

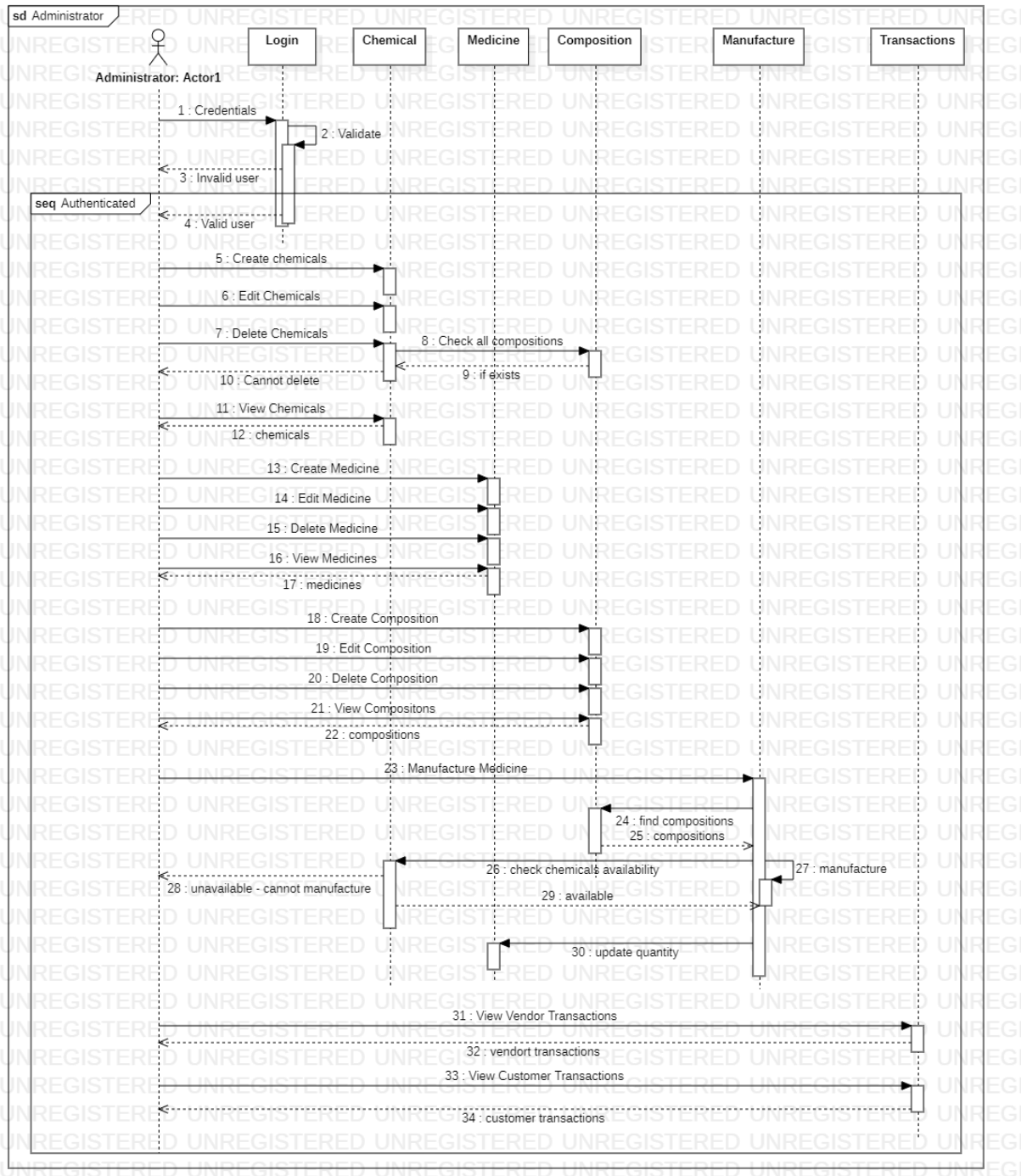

Figure 4 Sequence Diagram – Administrator

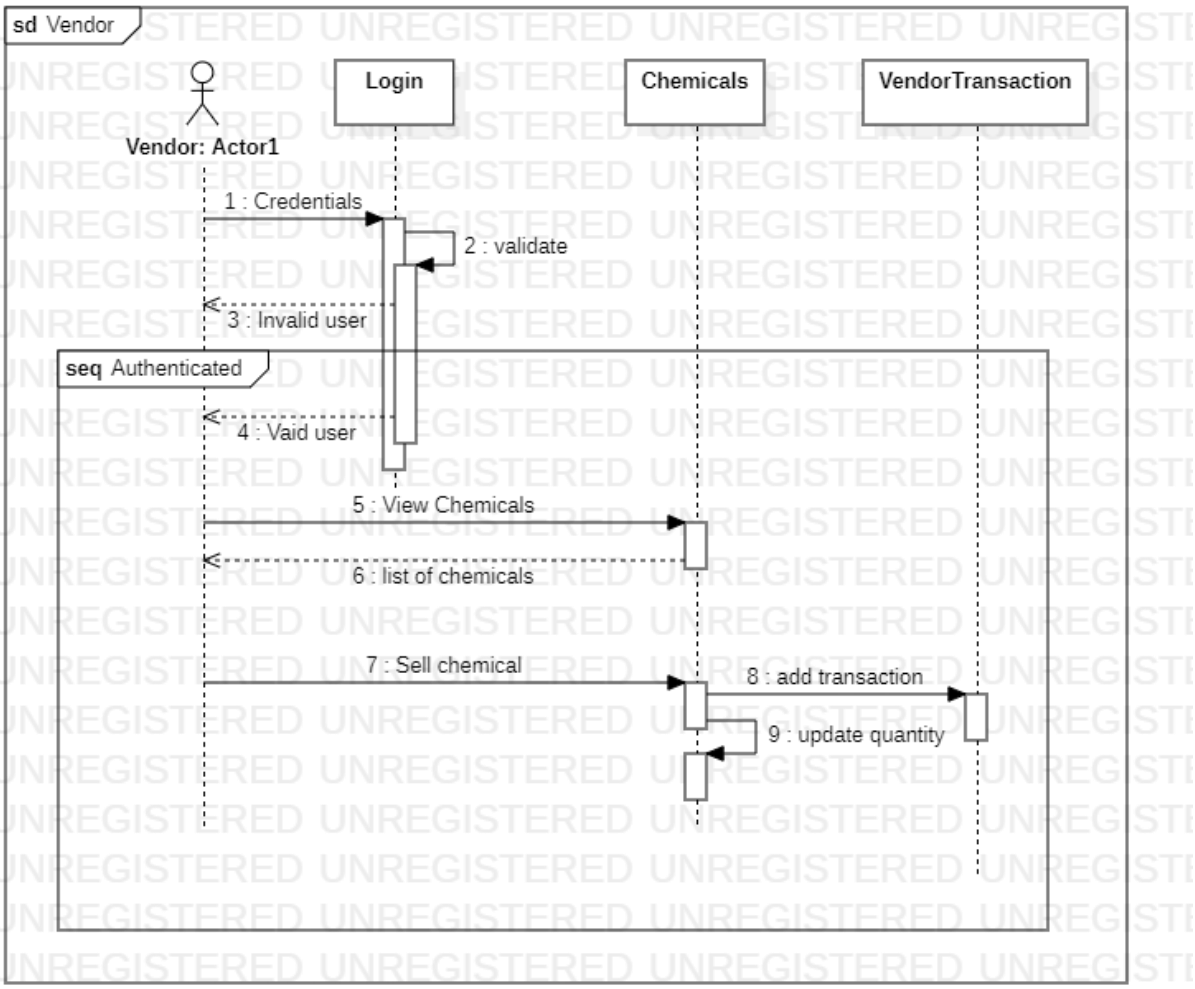

Figure 5 Sequence Diagram – Vendor

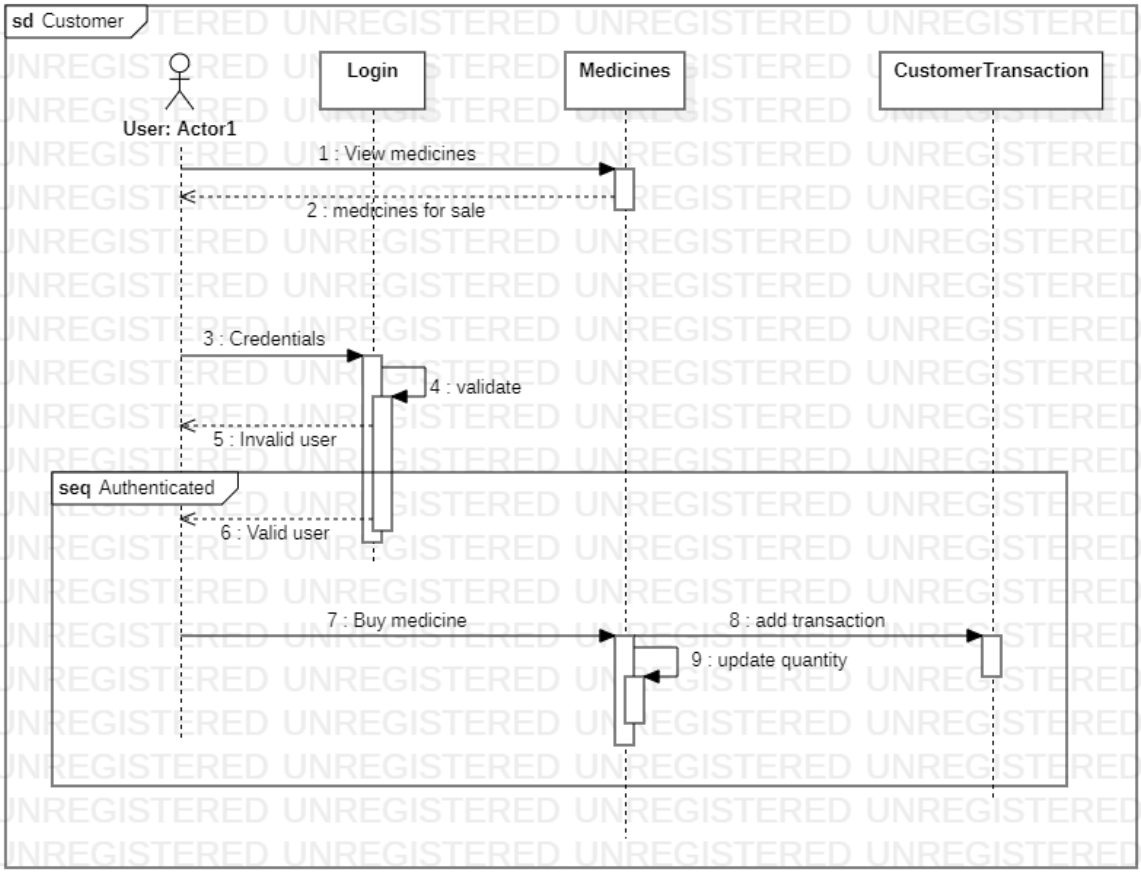

Figure 6 Sequence Diagram – Customer

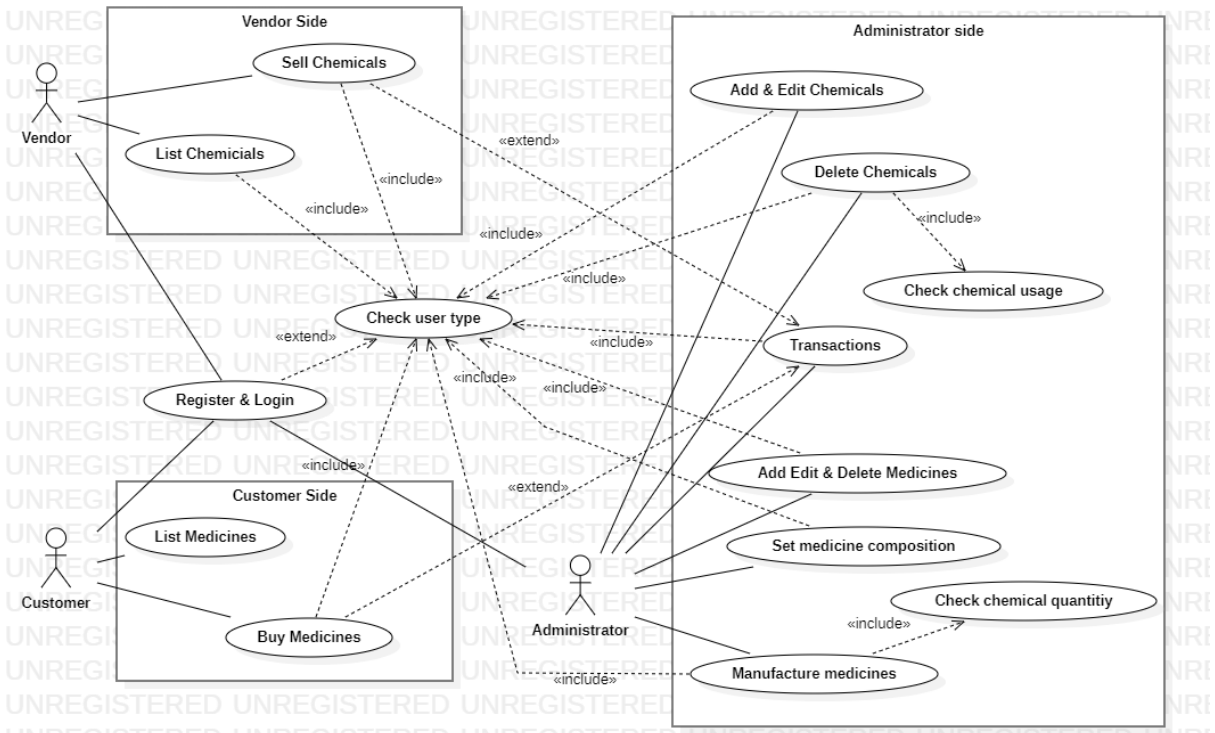

Figure 7 Use case Diagram

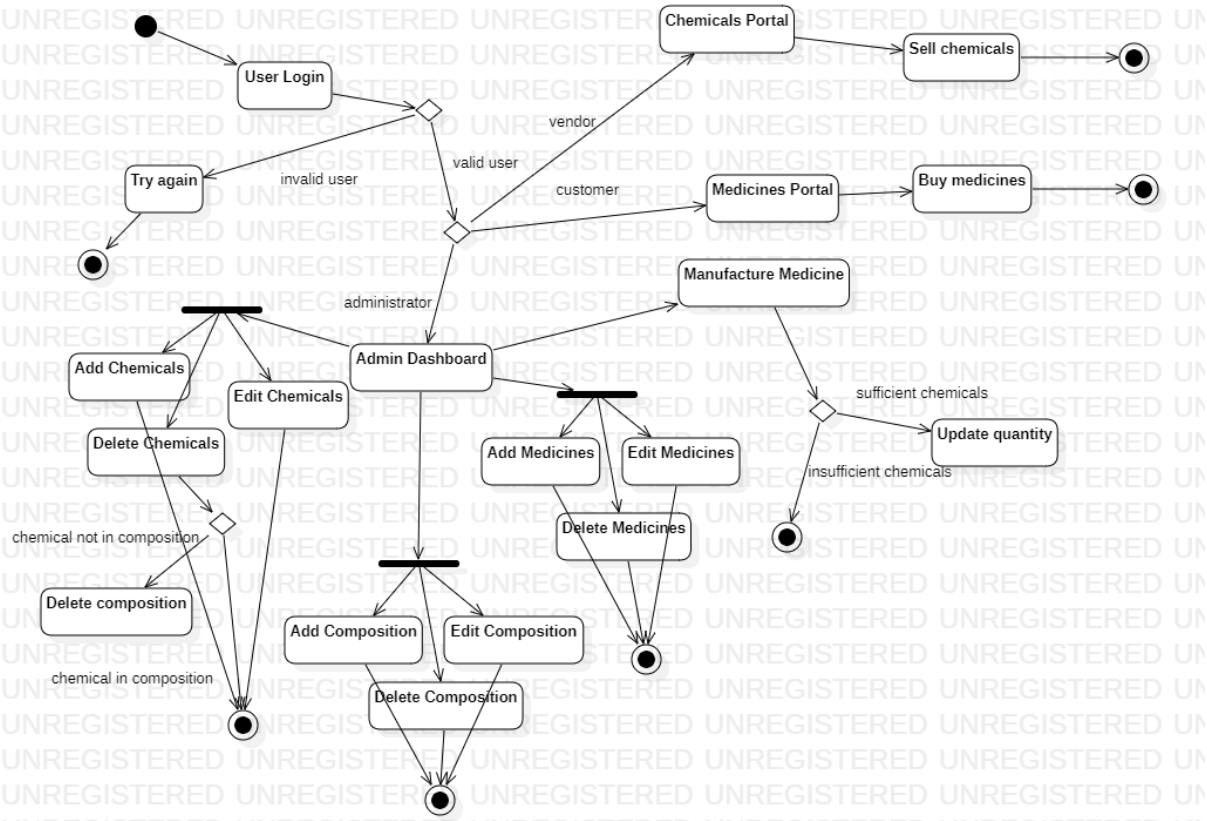

Figure 8 Activity Diagram

Appendix C: Implementation

Figure 9 Index Page

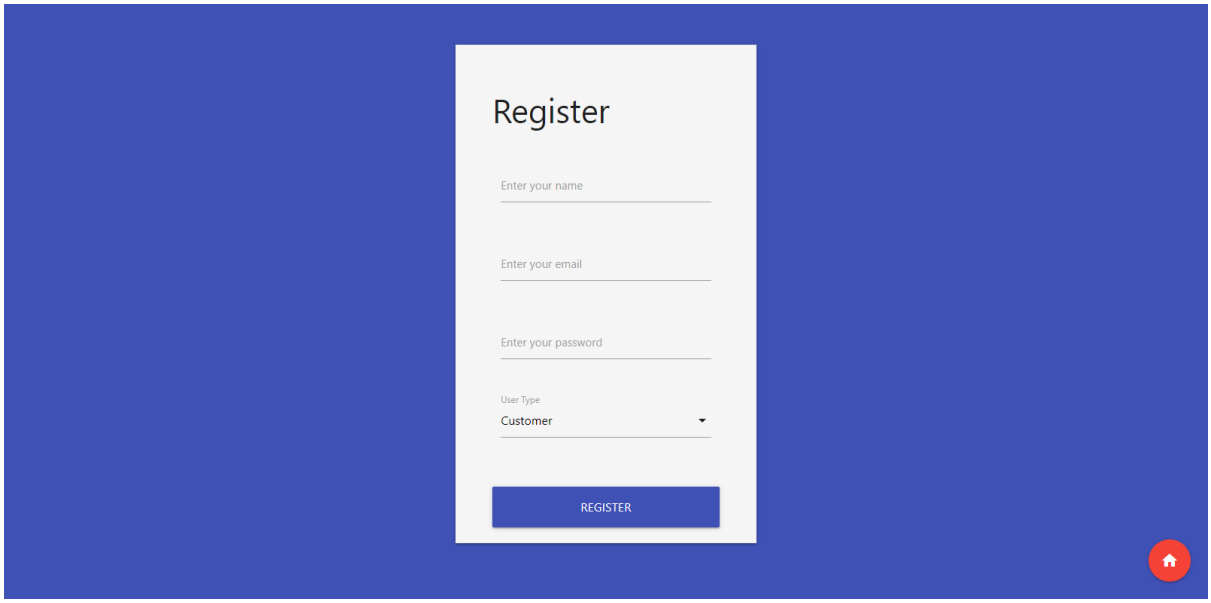

Figure 10 Register Page

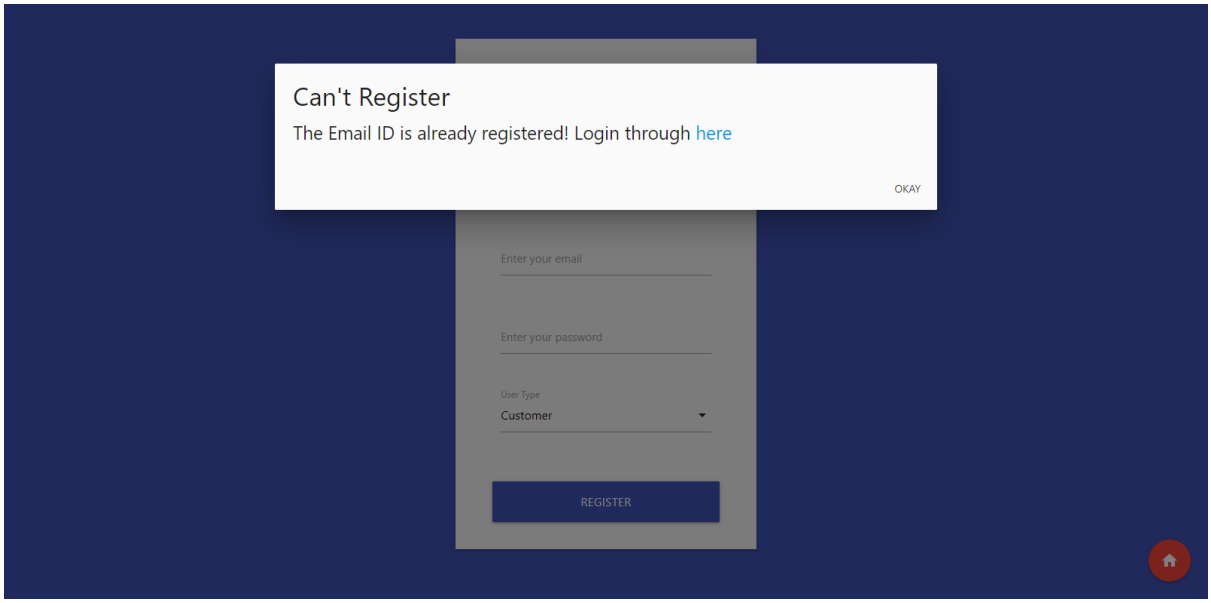

Figure 11 Error Prompt: Trying to register with existing email id

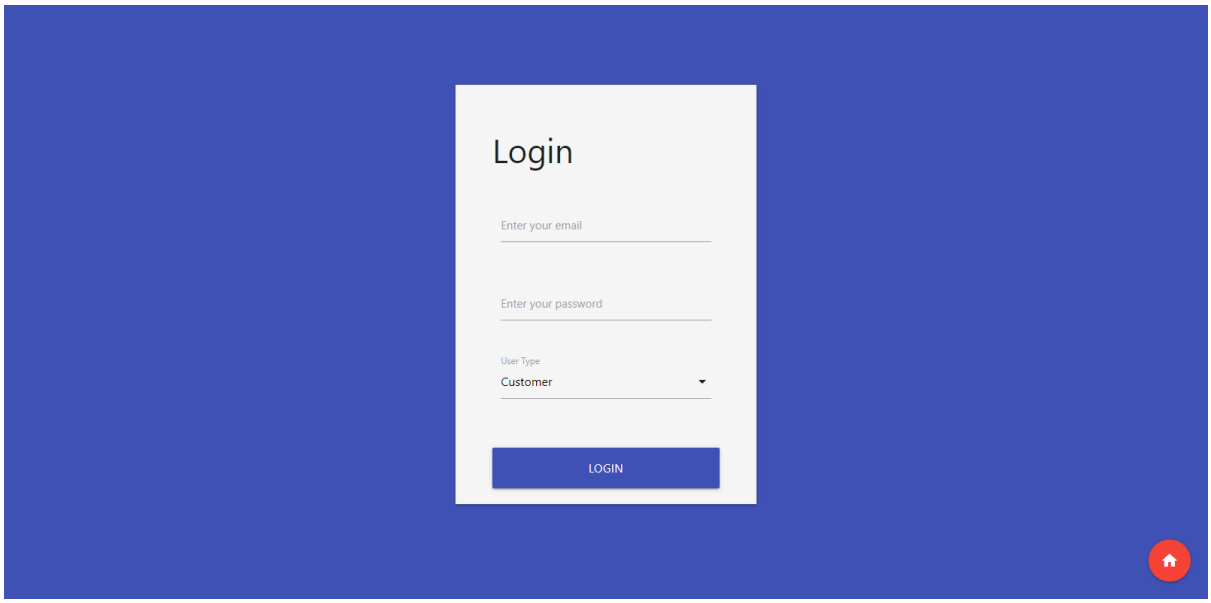

Figure 12 Login Page

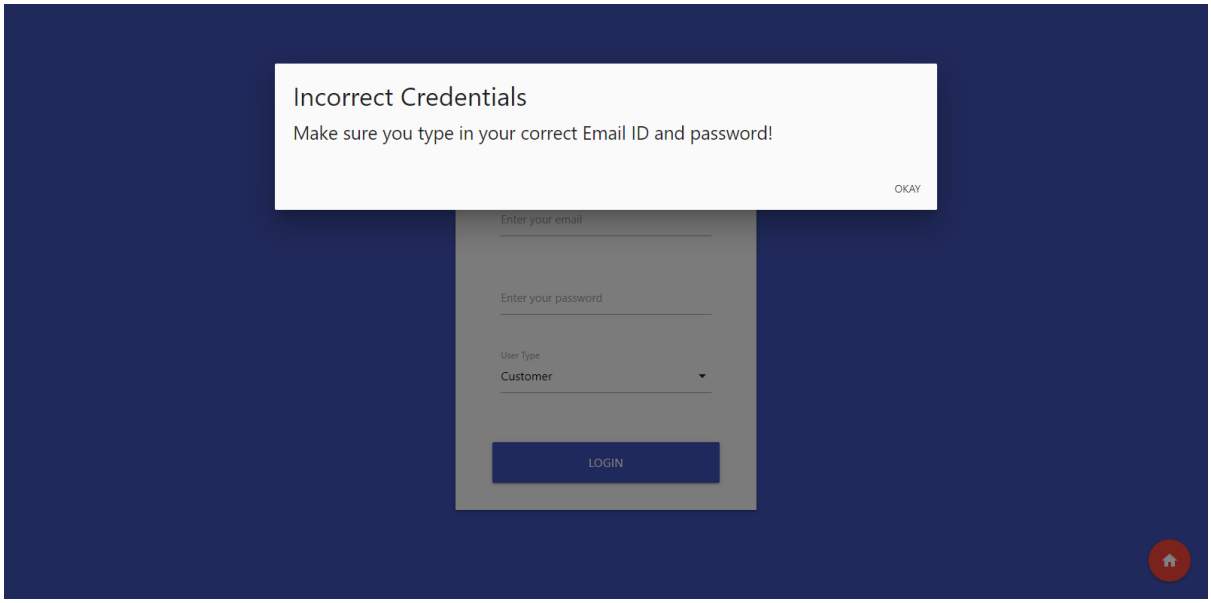

Figure 13 Error Prompt: Trying to logging in with incorrect credentials

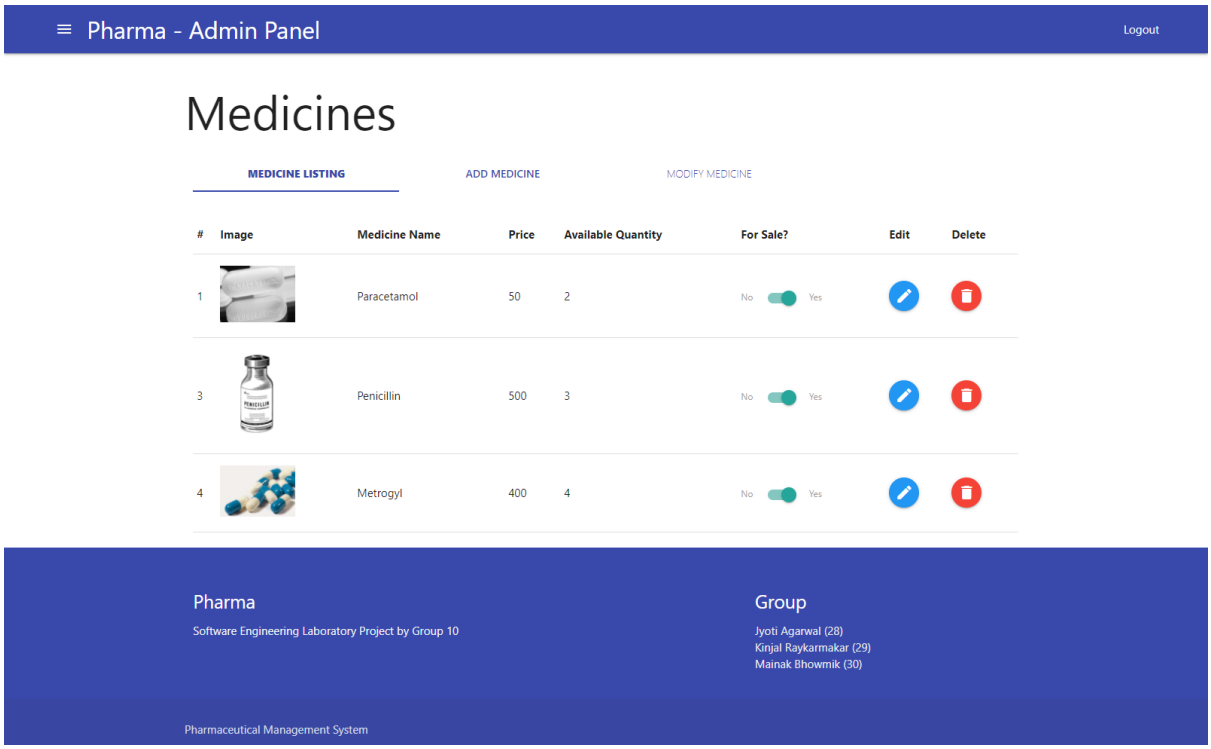

Figure 14 Admin Dashboard + Medicine Listing

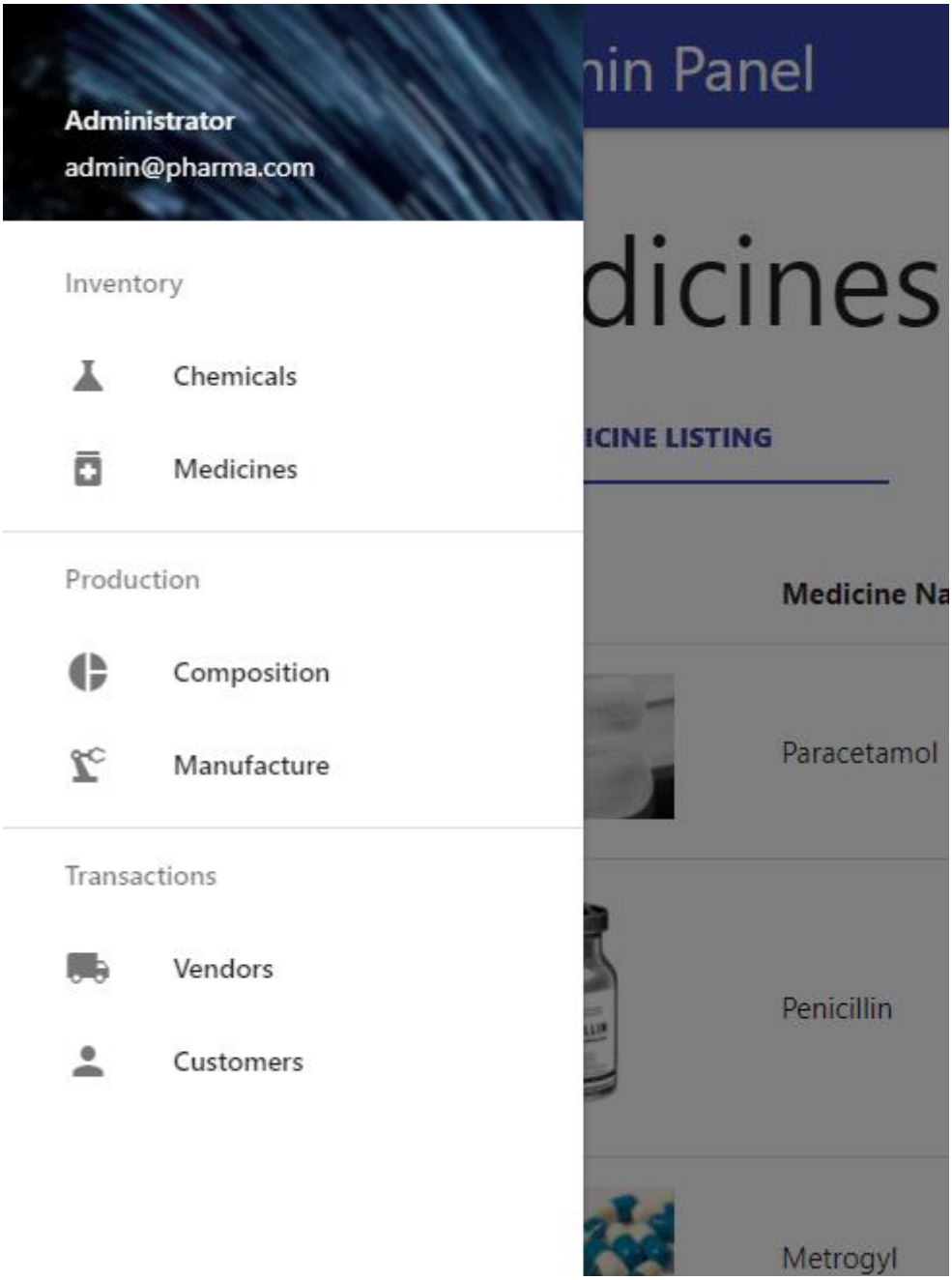

Figure 15 Sidebar for navigation

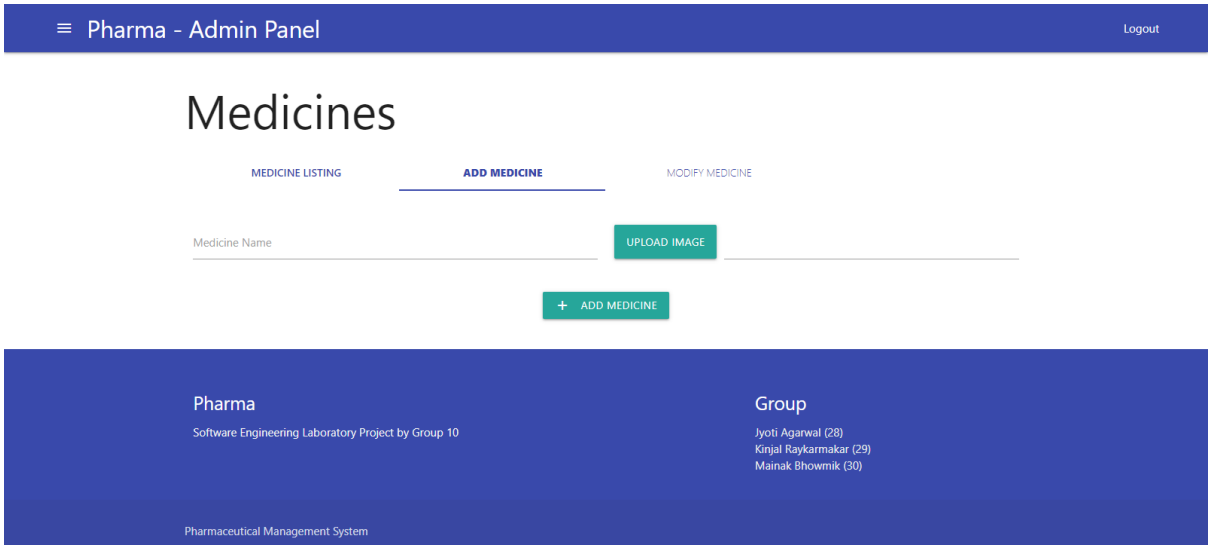

Figure 16 Add Medicine page

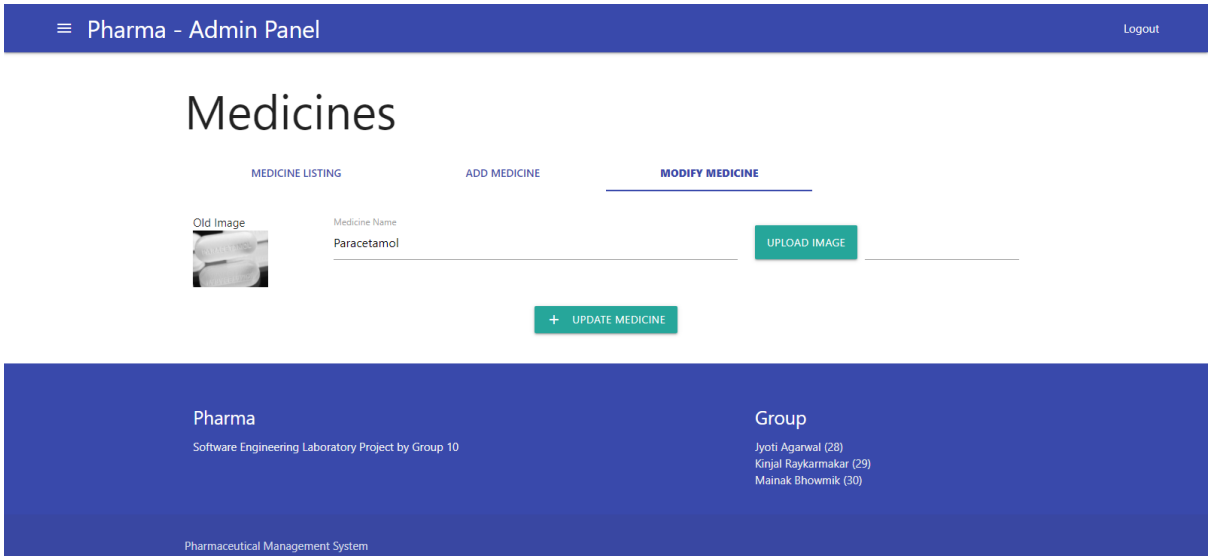

Figure 17 Modify Medicine Page

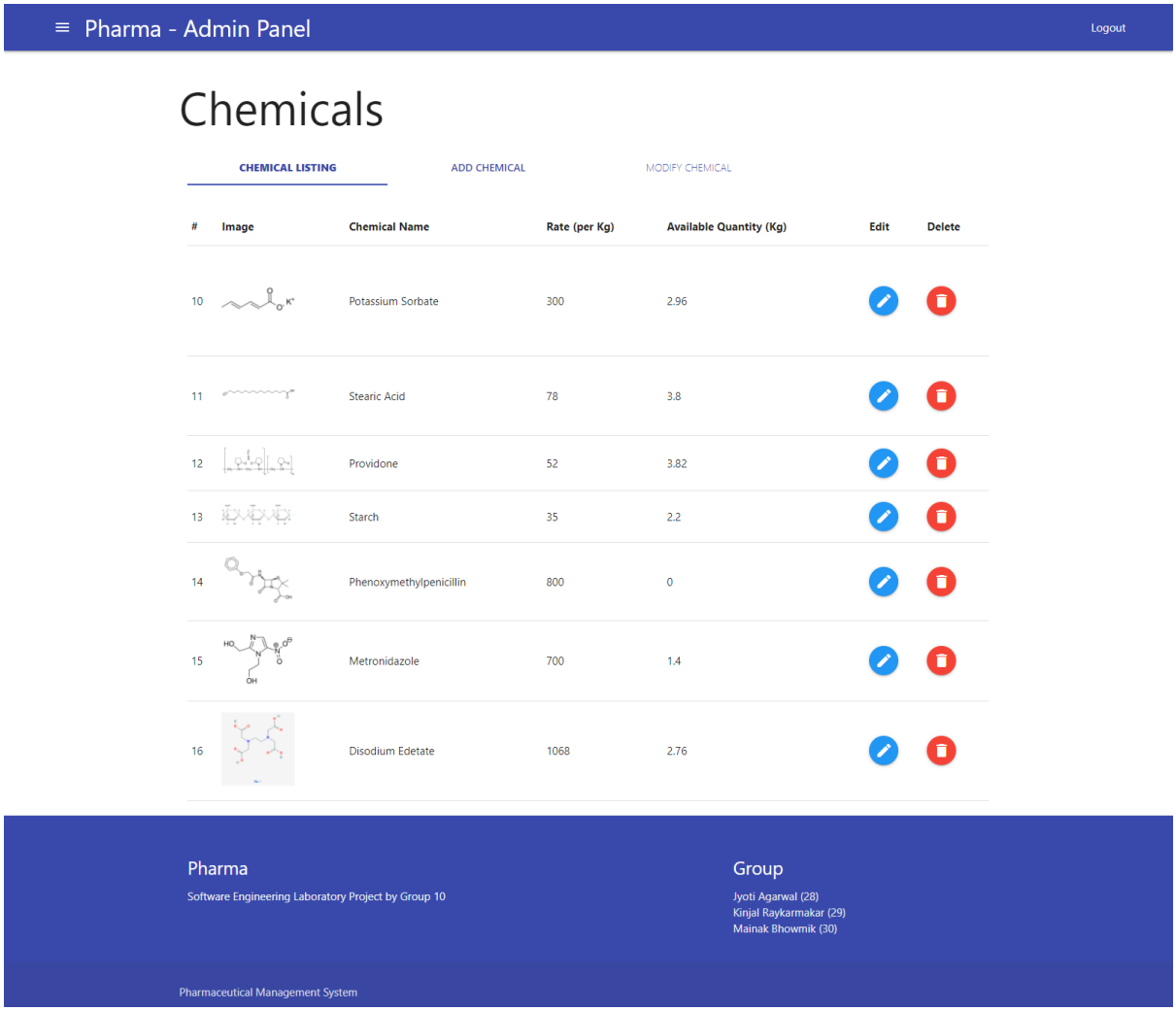

Figure 18 Chemical Listing Page

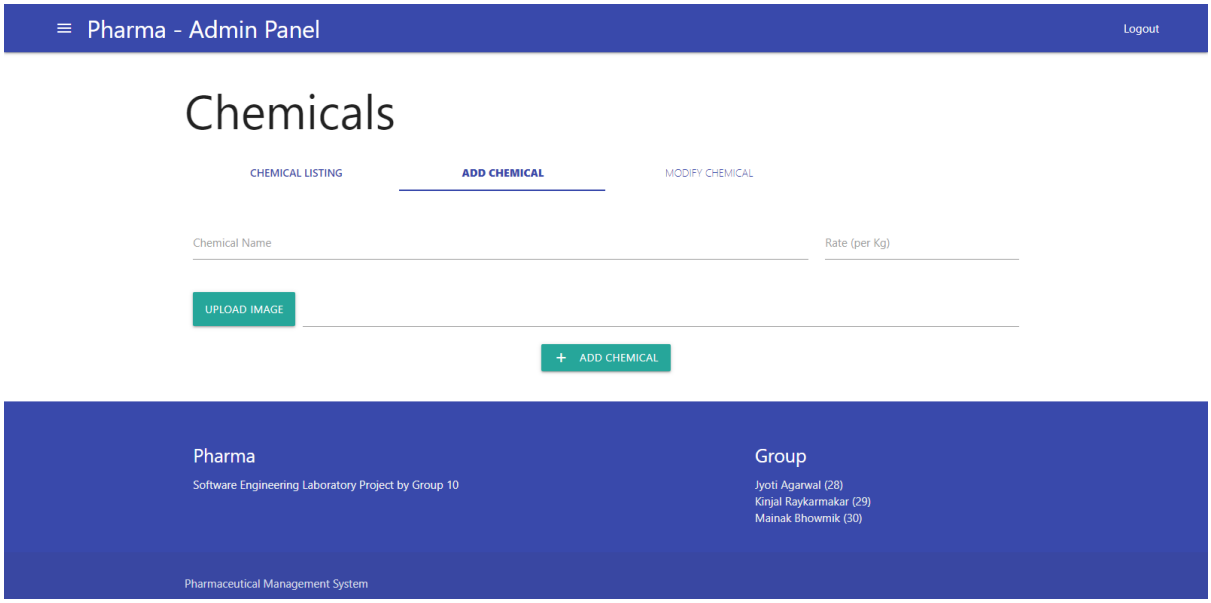

Figure 19 Add Chemical Page

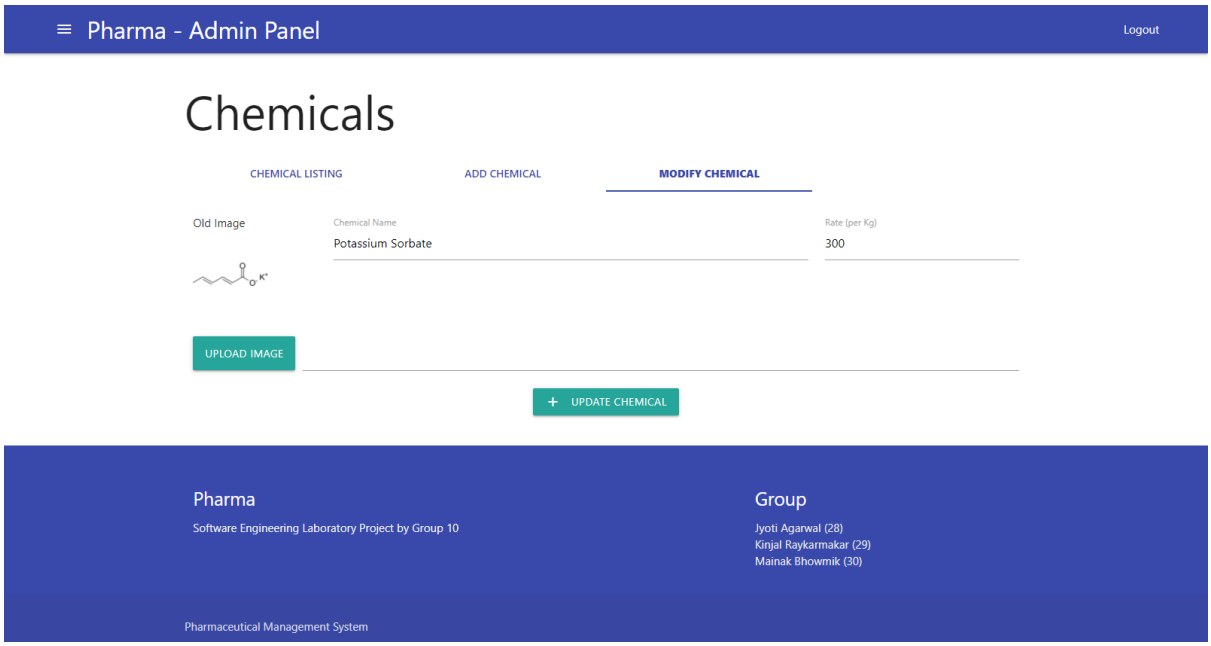

Figure 20 Modify Chemical Page

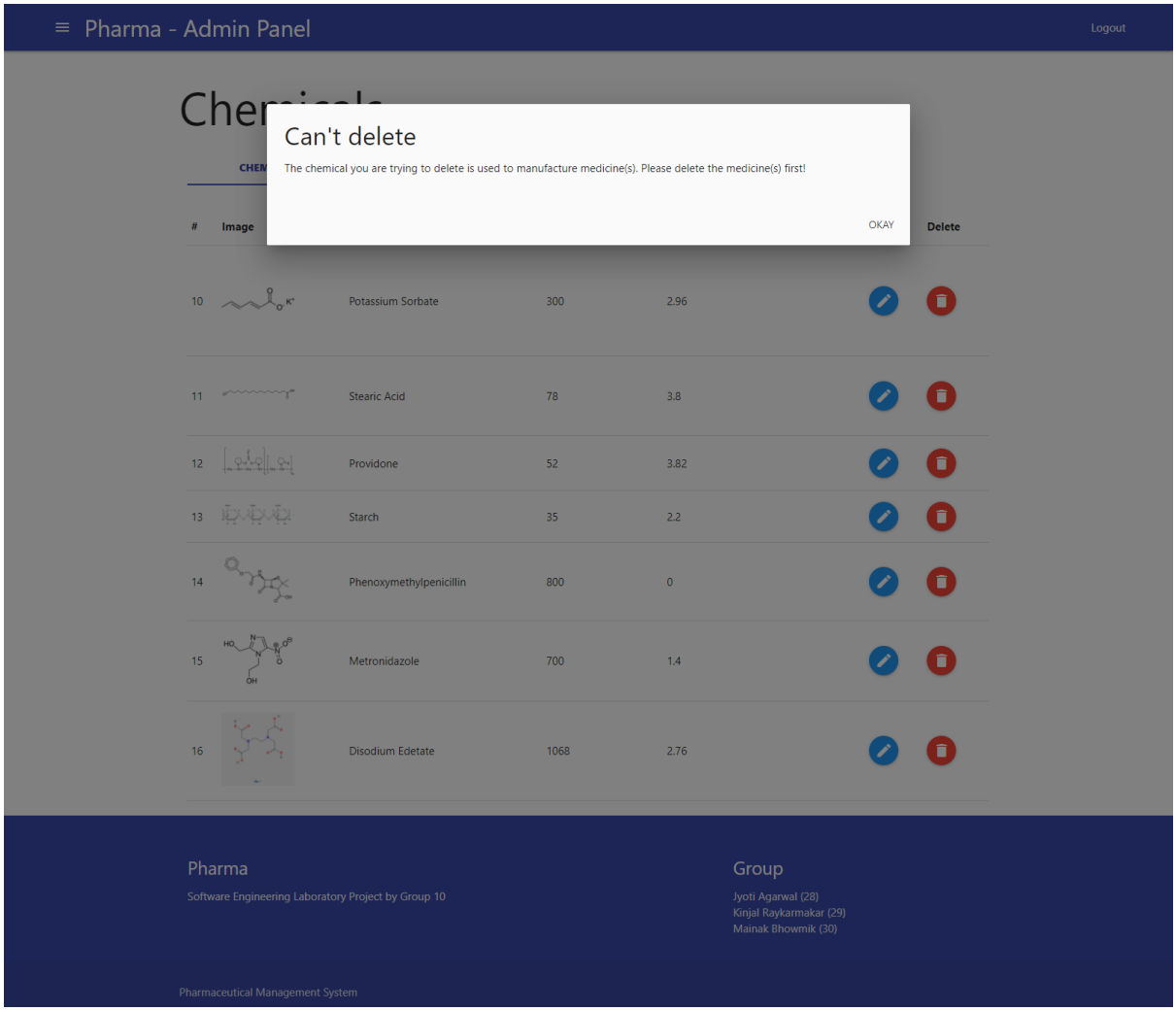

Figure 21 Error Prompt: Trying to delete a medicine which is used in a medicine

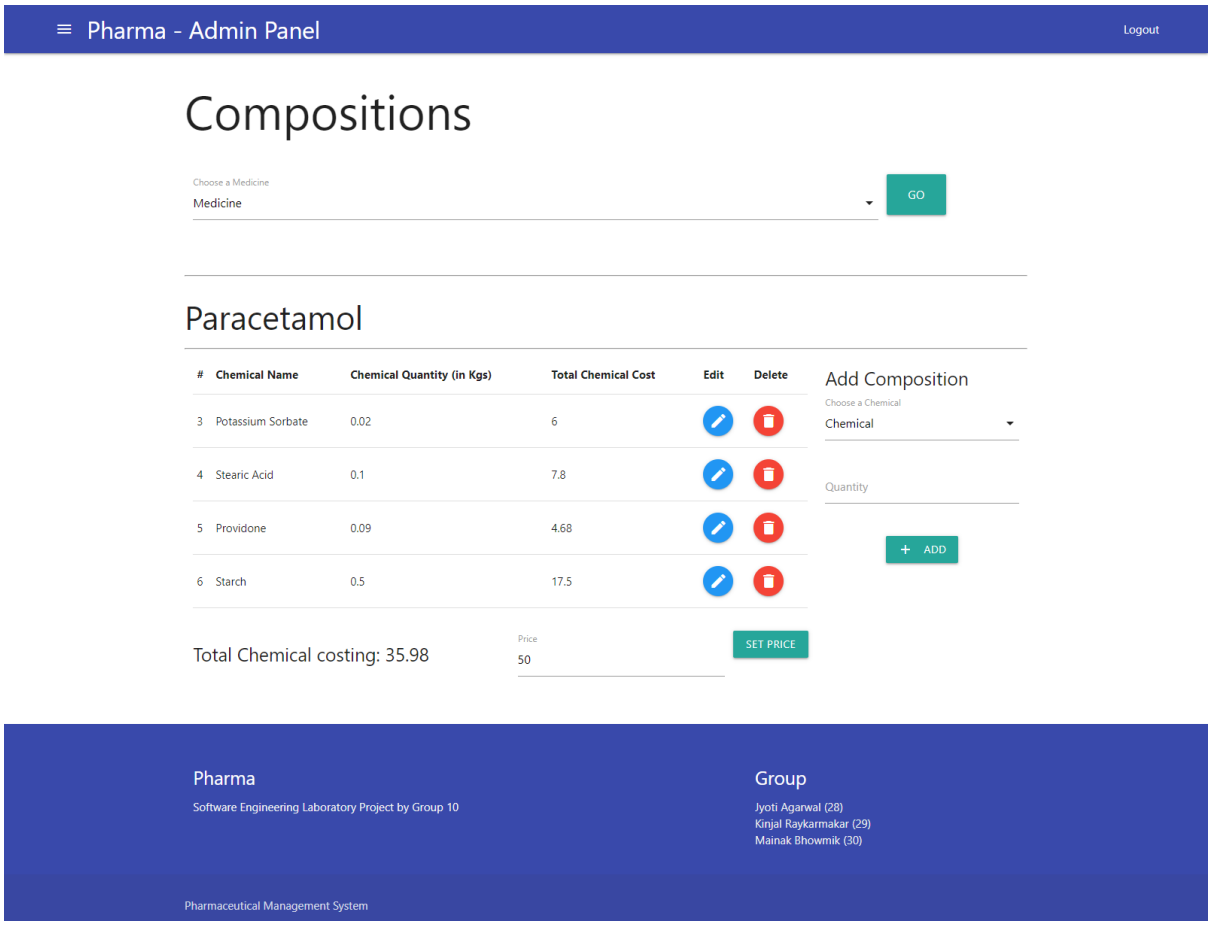

Figure 22 Medicine Composition Page – Including Add/Modify Composition and Medicine price setting

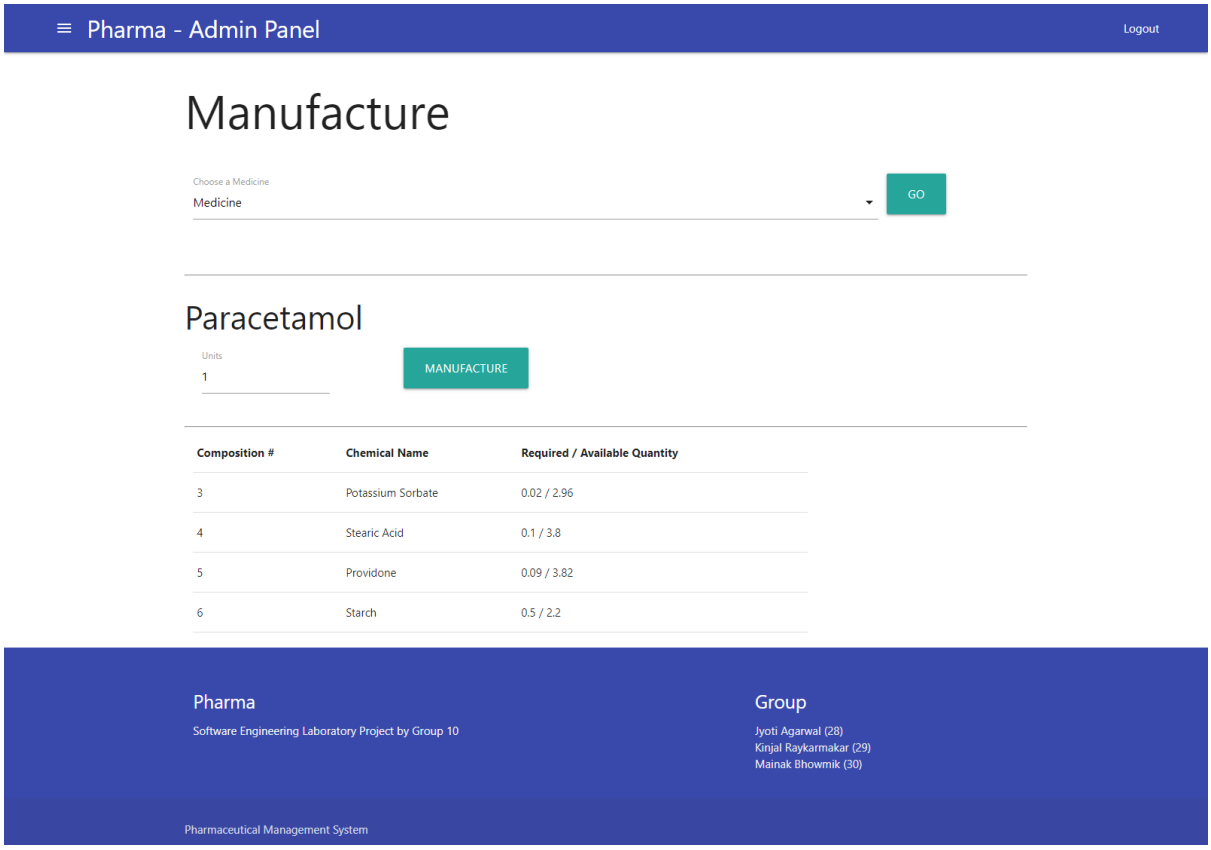

Figure 23 Medicine Manufacture page

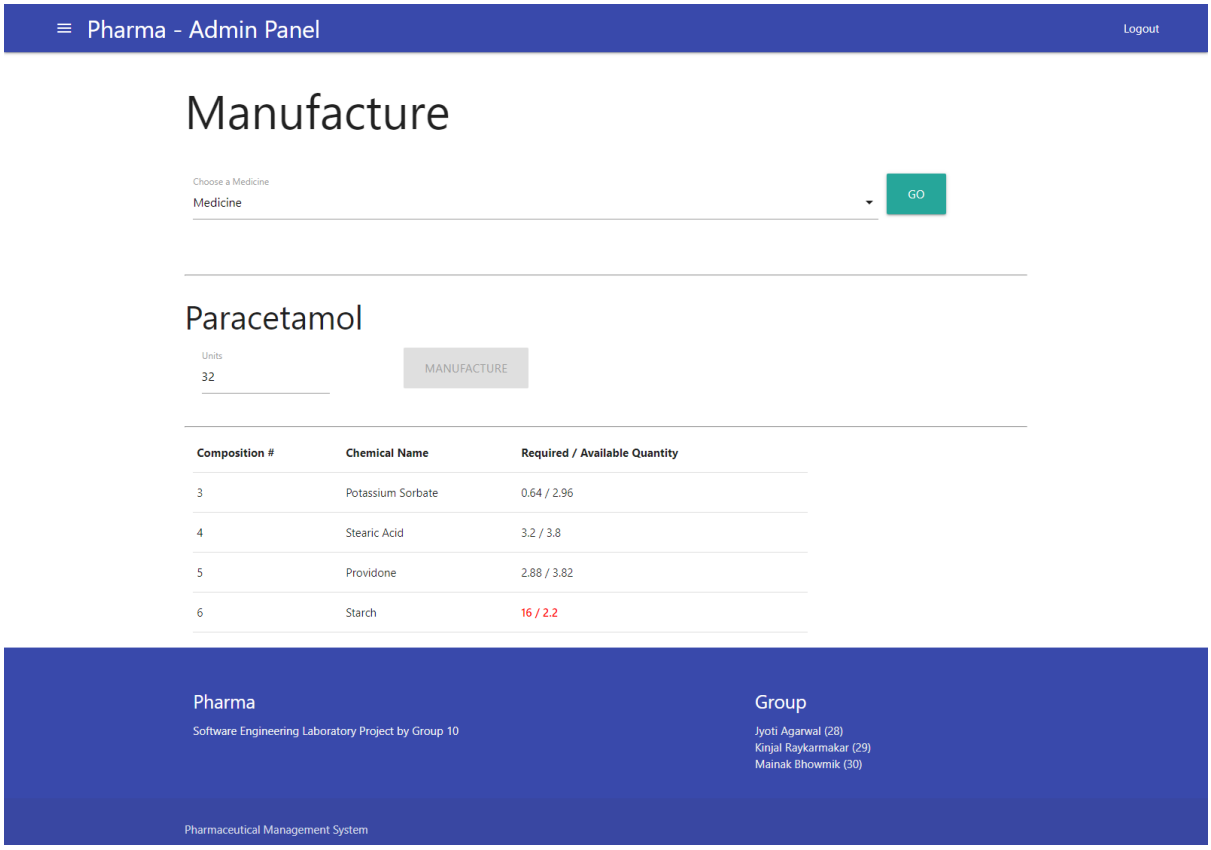

Figure 24 Trying to manufacture medicine with insufficient chemical quantity

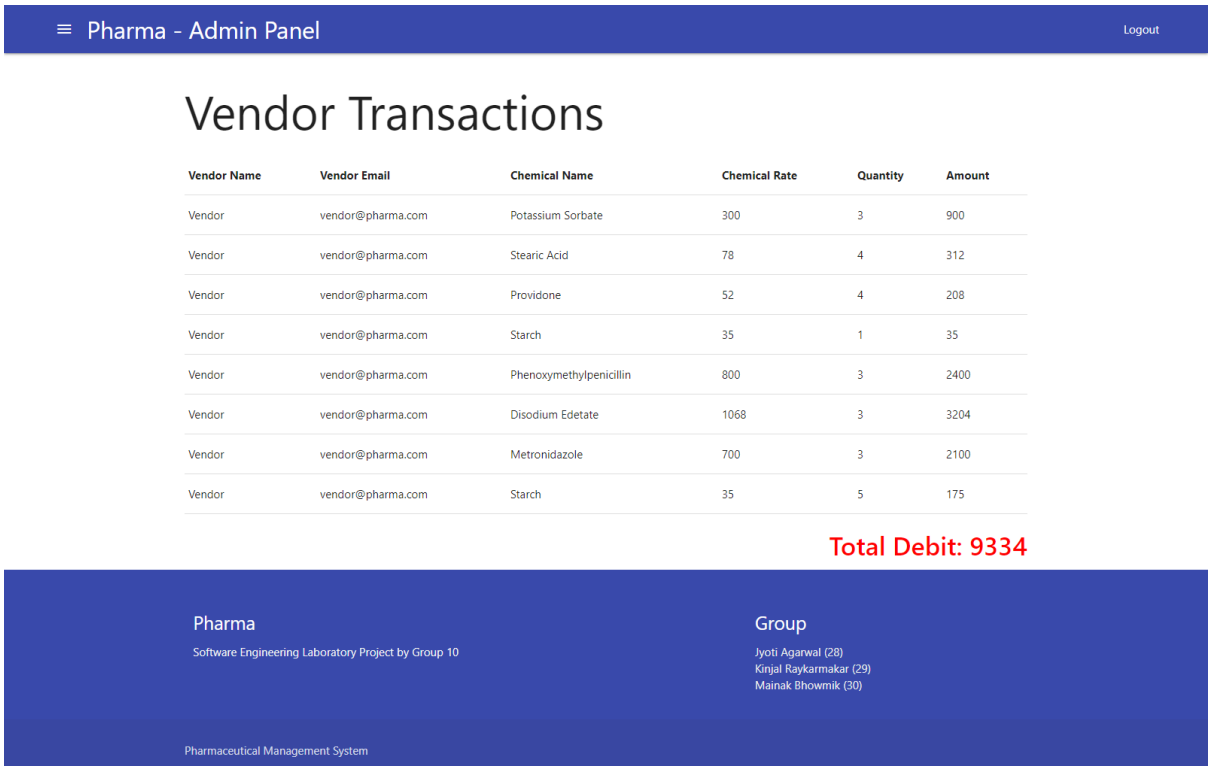

Figure 25 Listing Vendor Transactions

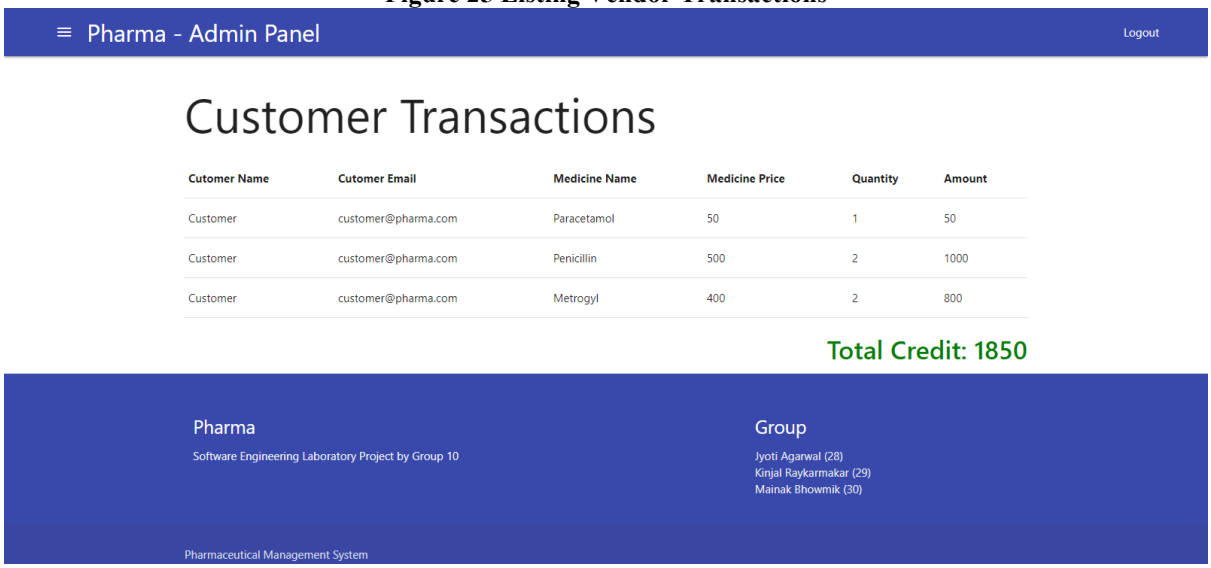

Figure 26 Listing Customer Transactions

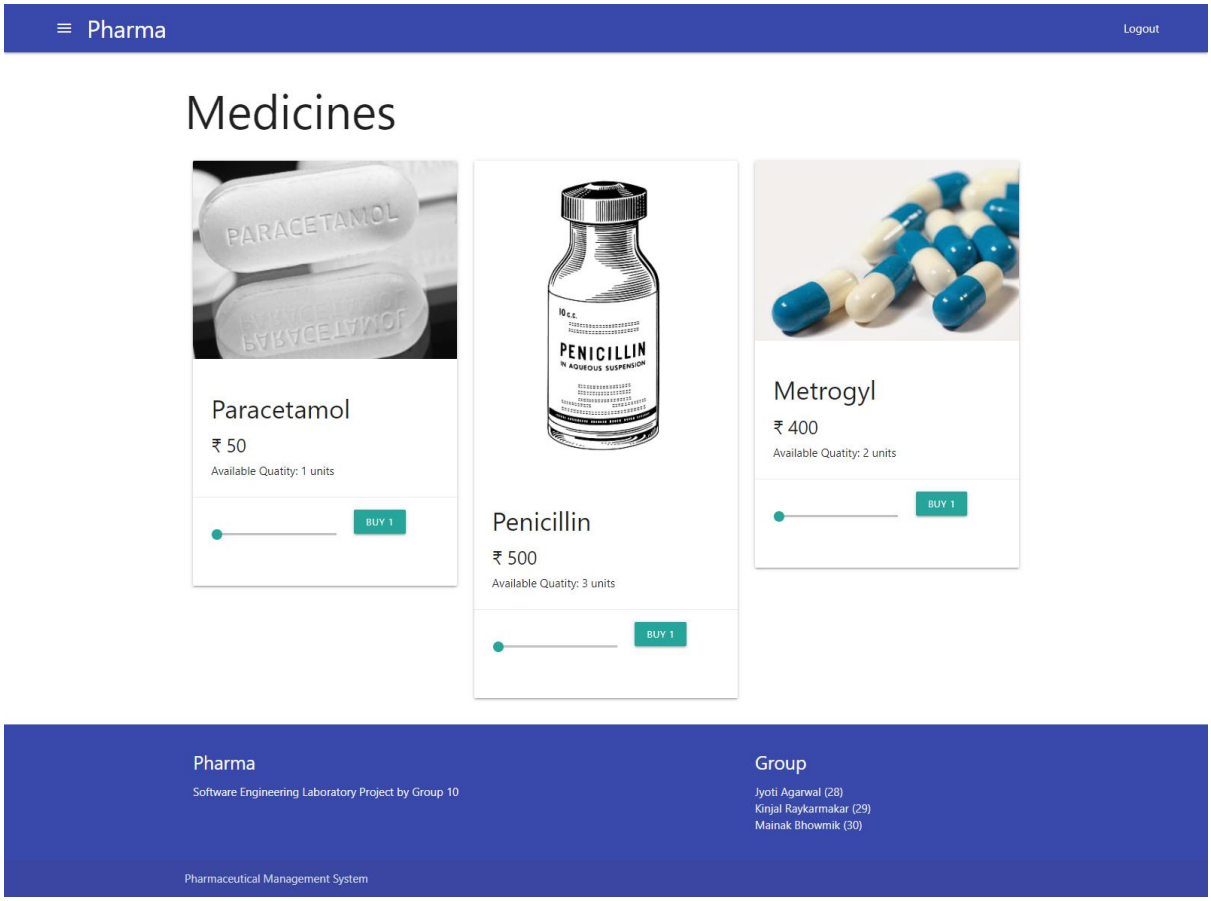

Figure 27 Index Page - Logged in as customer

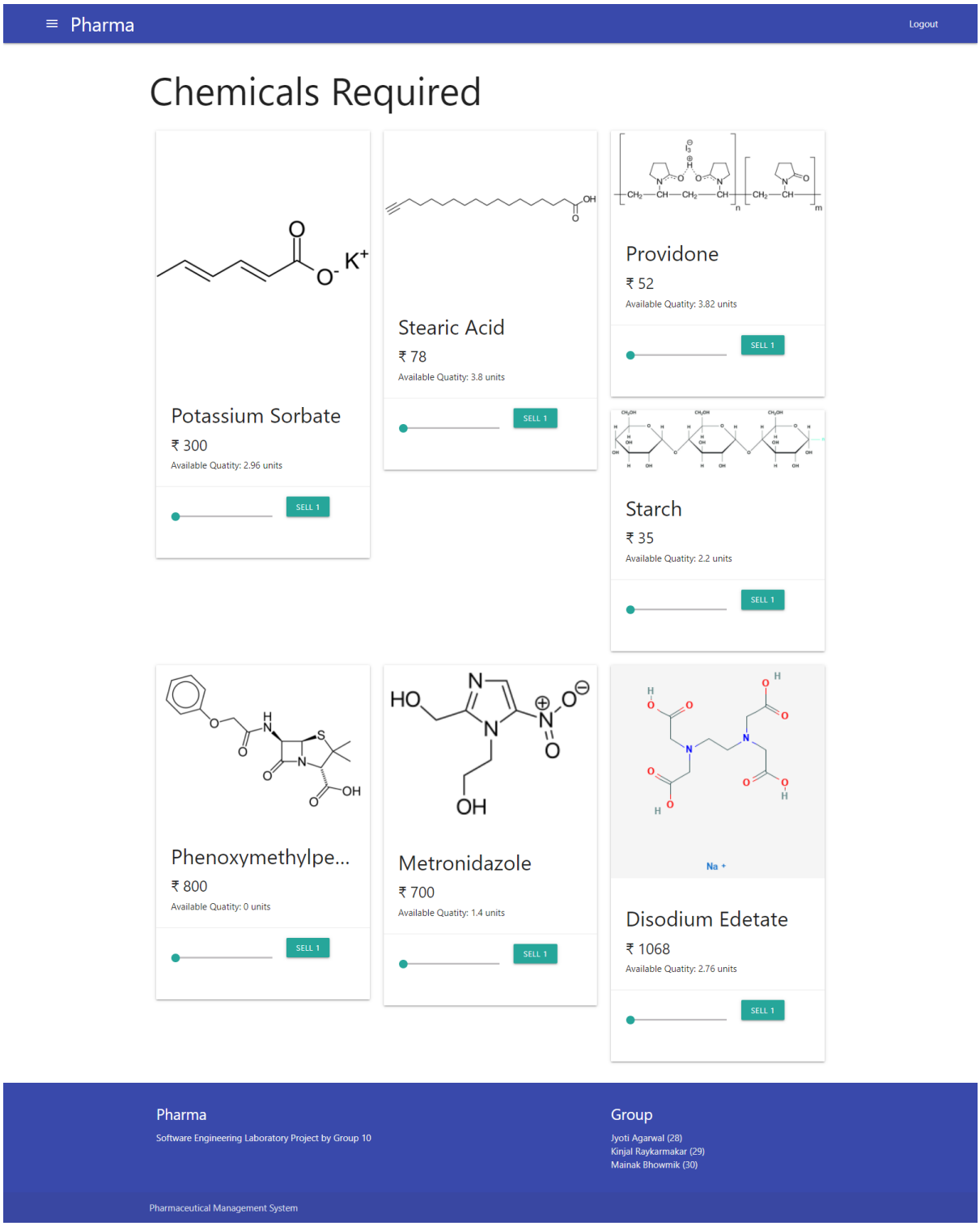

Figure 28 Vendor's page

Appendix D: Testing

Throughout the development process, each module was unit tested as well as integration tested to make remove any bugs or design issues. The following sections were overall tested

1. Links: All the internal links and anchor links were tested to work as expected.
2. Forms: All the forms were tested with various test cases to check the working of front-end form validations.
3. Session: Setting and resting session variables at times of login/register and logout respectively were thoroughly tested.
4. UI/UX: All buttons' clicks were tested to have the ripple effect in-line with the Material Design.
5. Database: Multiple test queries were run to make sure the database maintains data integrity and optimal response times.

7. Future Scope

This pharmacy management system can be efficiently used to automate the process of data management and maintenance of records related to stock, liquid flows, staffs, customers, and suppliers. New features, modules, and other components can be incorporated into the system as per user requirements in future.

- Efficient organization of all knowledge is the analysis company and easy analysis access and retrieval of information can be included.
- In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related medicines.
- Company using this software will always be able to plan in future and always be aware of their financial position in the market.
- The project can be made more robust by including biometric verification.
- There is also a scope to expand by implementing newer technologies like cloud etc.
- IoT can be incorporated to sync this application directly with medicine production.

In the coming days, a major chunk of business, trade and services are going to get digitalized on a greater scale. So, to keep up with the pace of digital progress and development, this product is a great option. Customers, users, patients, traders, shopkeepers, manufacturers, sellers, buyers, vendors, etc. are going to be immensely benefitted from this product. At times when going out in person for purchasing medicines or drugs, and also raw materials becomes extremely difficult due to any reason, this product comes into play. With a single click from the user-end device, they can access the product and its corresponding website, and thus can easily fulfil their requirements. In due course of time, we also intend to develop and upgrade the product, and enrich it with some additional features. This will make the product more sophisticated and provide a greater user experience with ease of usage. In this way, our product can also compete well in the digital market.

8. Bibliography

- https://en.wikipedia.org/wiki/Pharmacy_management_system
- https://www.researchgate.net/publication/305426561_Online_Pharmaceutical_Management_System
- https://www.msh.org/sites/msh.org/files/mds3-ch49-infosystems-mar2012.pdf