1. **Project Objective**

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

1. **Problem Statement**

Medicare is a company that supplies medicines and a couple of other healthcare essentials at an affordable price. It was established in 2012 in Delhi, India. It had been serving fine all these years, however, the business analysts noticed a decline in sales since 2017. They found out that online ordering of medicines with companies, such as 100mg and mfine are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.  
You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

**Features of the application:**

1. Registration
2. Login
3. Payment gateway
4. Searching
5. Filtering
6. Sorting
7. Dynamic data
8. Responsive and compatible with different devices

**Recommended technologies:**

1. Database management: MySQL and Oracle
2. Backend logic: Java programming, NodeJS
3. Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript
4. Automation and testing technologies: Selenium, Jasmine (frontend testing), and TestNG
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

**Project development guidelines:**

* The project will be delivered within four sprints with every sprint delivering a minimal viable product.
* It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
* The learner can use any technology from the above-mentioned technologies for different layers of the project.
* The web application should be responsive and should fetch or send data dynamically without hardcoded values.
* The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
* The learner must implement a CI/CD pipeline using Jenkins.
* The learner should also deploy and host the application on an AWS EC2 instance.
* The learner should also implement automation testing before the application enters the CI/CD pipeline.
* The learner should use Git branching to do basic automation testing of the application in it separately.
* The learner should make a rich frontend of the application, which is user-friendly and easy for the user to navigate through the application.
* There will be two portals in the application, namely admin and user portal.

**Admin Portal:**  
The admin portal deals with all the backend data generation and product information. The admin user should be able to:

* Add or remove medicine details from the application to build a rich product line
* Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices
* Enable or disable a medicine product

**User Portal:**  
It deals with the user activities. The end-user should be able to:

* Sign-in to the application to maintain a record of activities
* Search for products based on the search keyword
* Apply filters and sort results based on different cuisines to get the best deals
* Add all the selected food items to the cart and customize the purchase at the end
* Perform a seamless payment gateway
* Get an order summary details page once the payment is complete

1. **Application Name**

Medicare (Java e-healthcare web application)

1. **Developer Details**

Anurag Pal, 216 Java SG (Evening Batch), Phase-6 Capstone Project.

1. **Sprint Planning**

To complete this application, four sprints were planned to develop different aspects of the Java e-healthcare web-based application.

**Sprint 1 –** Designed a RESTful web service to act as the backend. Three entities were created namely – Admin, User and Medicines. User and Medicines have a Many-To-Many relationship in the form of another table Orders. MySQL is used as a database. The CRUD operations are carried out using JPA Repository.

There are three controllers namely – Admin Controller for handling all the admin related features using Get Mapping, Post Mapping and Put Mapping. These features are executed using the Admin Service.

User Controller for handling all the user related features using Get Mapping, Post Mapping and Put Mapping. These features are executed using the User Service.

Medicines Controller for handling all the medicines related features using Get Mapping, Post Mapping and Put Mapping. These features are executed using the Medicines Service.

**Sprint 2 –** Designed the index page which displays two links using a nav bar, admin login and user login.

After choosing admin login, a form is displayed asking the admin to enter username and password. A link is also given for new admin registration. This data will be stored into the database. After logging in, the admin dashboard is displayed with a nav bar having options – show list of admins, show list of users, show list of all medicines, show list of enabled medicines, change password and logout.

List of admins shows a list of admins registered for the application in a tabular format. List of users shows a list of users registered for the application in a tabular format.

**Sprint 3 –** List of medicines shows a list of all medicines from the database. Three buttons are also displayed – add a medicine, edit a medicine and remove a medicine. Add a medicine displays a form asking the admin for medicine details which are then stored into the database. Edit a medicine also returns a form for all the medicine details, which are then edited and stored into the database. Remove a medicine asks for the medicine ID of the medicine, which is then removed from the database. The list of medicines can also be sorted by category, price and date. Finally, an enable button is displayed beside every medicine clicking on which enables the medicine. All medicines are disabled by default.

List of enabled medicines shows a list of all enabled medicines from the database.

Change password provides a form asking the admin for username and new password. If the username entered does not match any username from the database, then an error message is displayed, otherwise a success message is displayed.

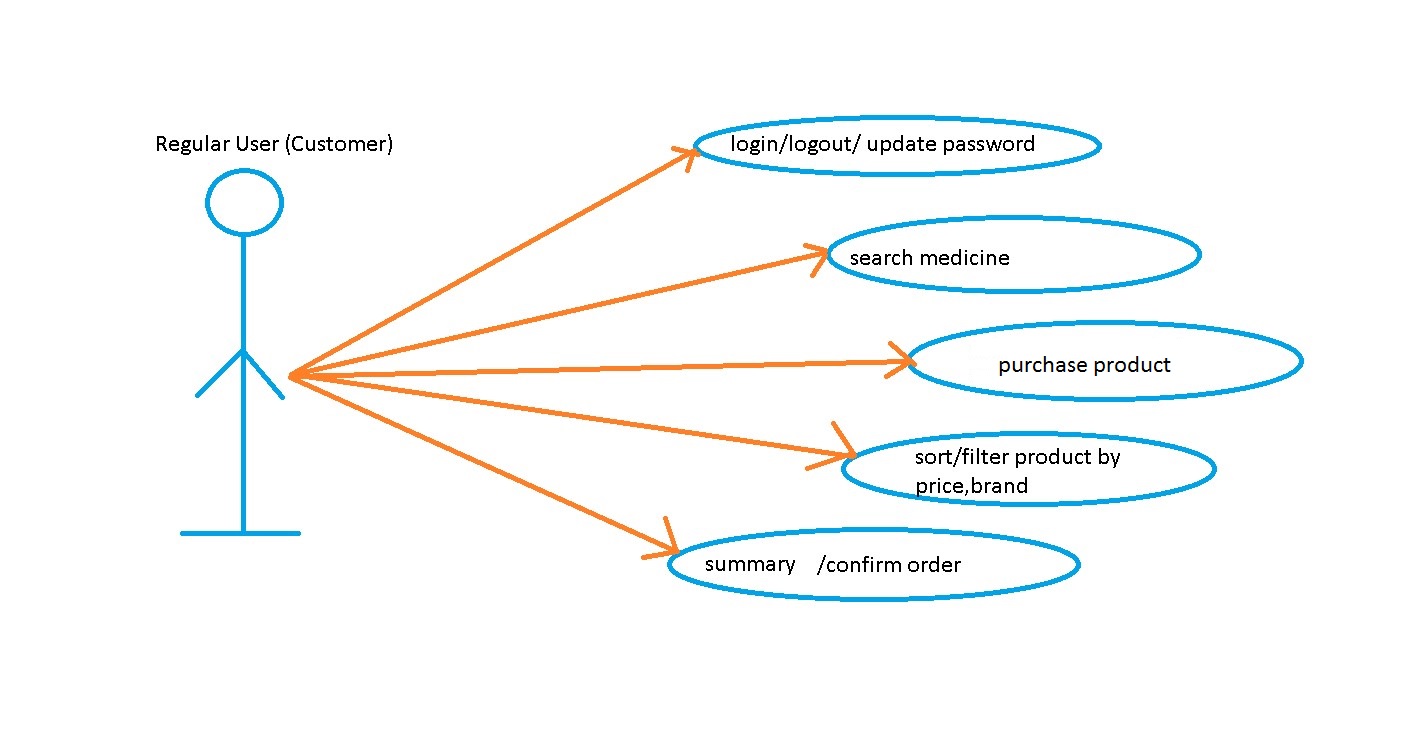
Finally, logout shows a success message and the admin can then login again.

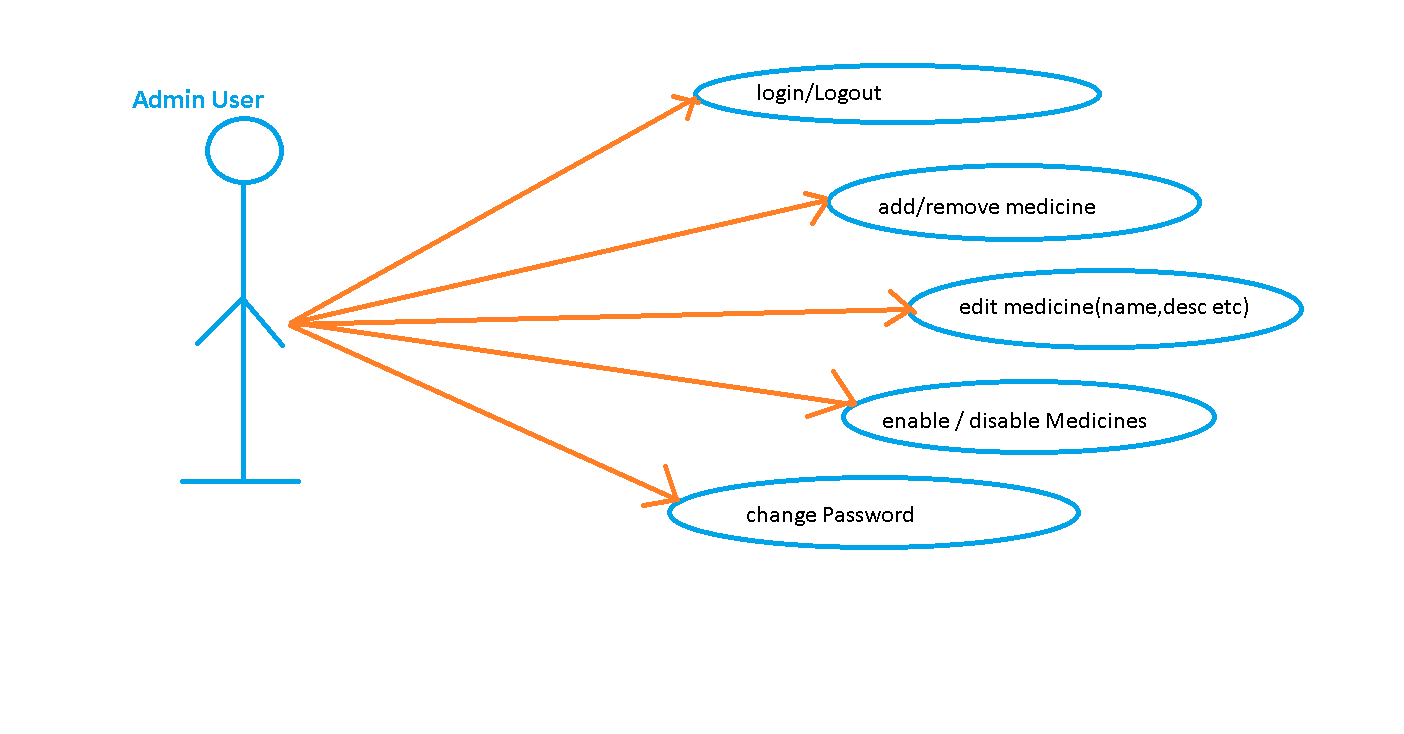
**Sprint 4 –** After choosing user login, a login form is displayed asking the user for username and password. A link is also given for new user registration. This data will be stored into the database. After logging in, the user dashboard is displayed with a nav bar having three options – search for a medicine, change password and logout.

Search for a medicine displays a form asking the user for the medicine name. All the medicines matching that name are then displayed below in a table. Beside every medicine, a purchase button is provided clicking on which the user will be redirected to the payment gateway displaying the order details and asking the user for the quantity. Once user clicks on the book button, a successful message is provided.

Change password provides a form asking the user for username and new password. If the username entered does not match any username from the database, then an error message is displayed, otherwise a success message is displayed.

Finally, logout shows a success message and the user can then login again.

1. **Use-Case Diagrams**

****

1. **Core Concepts Used**
2. Database and Table creation done in MySQL.
3. Front end created using Angular and JavaScript, and back end is created using RESTful web services.
4. Total four tables were created namely – Admin, User, Medicines and Orders. Those four tables were mapped with servlets using Entity classes.
5. CRUD operations are used using JPA Repository and the features are executed using controllers and services.
6. The front end is designed using angular components, each feature for each component.
7. Three classes have been created for Admin, Medicines and User.
8. Three services have also been created, admin service, user service and medicine service.
9. The designing is done using html, bootstrap and CSS.
10. **Conclusion**

This application is mainly designed using RESTful web services, Angular and JavaScript.

The RESTful web services is used to create the backend of the application using MySQL database to store details.

The concept of angular components is used to create different aspects of the application while JavaScript, HTML, Bootstrap and CSS are used together to create the front-end of the application.

**Unique Selling Points (USPs)**

1. User friendly interface helps admins and users to guide through different options without any hassle.
2. Displaying list of admins, users and medicines can be done using just the click of a button. Adding, editing and removing medicines are also simple and easy to navigate.
3. Users can easily search for a medicine and then buy it with just a click of a button.