**Java Pharmacy Management System**

Welcome to the Java Pharmacy Management System! This web-based application is designed to streamline pharmacy operations, offer online medicine ordering, and facilitate remote doctor consultations. With an intuitive user interface and automated features, managing healthcare needs has never been more convenient.

Functional and non-functional requirements

Functional Requirements:

1. User Registration and Authentication:

* Users should be able to register with the system using their email and password.
* Authentication mechanisms should ensure secure access to the system.

2. Medicine Catalog:

* The system should maintain a comprehensive catalog of medicines, including details like name, dosage, usage instructions, and price.

3. Online Medicine Ordering:

* Users should be able to browse the medicine catalog and place orders for medicines online.
* Users can add medicines to their cart, specify quantities, and proceed to checkout.

4. Doctor Consultation:

* Users should have the option to consult with doctors online.
* The system should facilitate scheduling appointments, conducting video calls, and receiving medical advice.

5.Order Management:

* Users should be able to view their order history, including details of ordered medicines, quantities, and statuses.
* The system should provide real-time updates on order status (processing, dispatched, delivered).

6.Inventory Management:

* Pharmacy staff should be able to manage the stock of medicines.
* Automatic alerts should notify staff when stock levels are low.

7.User Profiles:

* Users should have profiles where they can manage personal information, addresses, and communication preferences.

Non-Functional Requirements:

1.Security:

* Data encryption should be employed for sensitive information.
* Secure authentication and authorization mechanisms are essential to prevent unauthorized access.

2.Performance:

* The system should handle a significant number of concurrent users without significant slowdowns.
* Response times for actions like placing orders and loading medicine details should be quick.

3.Scalability:

* The system should be designed to scale horizontally to accommodate increasing user loads.

4.Reliability:

* The system should be available and functional 24/7 to accommodate users from different time zones.
* Backup and disaster recovery mechanisms should be in place to prevent data loss.

5.Usability:

* The user interface should be intuitive and user-friendly, catering to users of varying technical backgrounds.
* Navigation and interactions should be smooth and efficient.

6.Compliance and Regulation:

* The system should adhere to relevant medical and pharmaceutical regulations, ensuring accurate medication information and secure handling of medical records.

7.Accessibility:

* The system should be designed to accommodate users with disabilities, conforming to accessibility guidelines.

8.Integration:

* The system should support integration with external services for payment processing, video conferencing (doctor consultations), and more.

Use Cases

1.User Registration and Authentication:

Use Case: User Registration and Login

Actors: User

Description: A user registers an account and logs in to the system.

Flow:

1.User navigates to the registration page.

2.User provides email and password.

3.System validates the email format and password strength.

4.System securely stores the hashed password and other user details.

5.User logs in with the registered email and password.

6.System validates credentials and grants access if valid.

2. Medicine Ordering:

Use Case: Place Medicine Order

Actors: User, Pharmacy Staff

Description: A user places an order for medicines.

Flow:

1.User logs in to the system.

2.User browses the medicine catalog.

3.User adds desired medicines to the cart.

4.User specifies quantities and proceeds to checkout.

5.System calculates the total cost.

6.User confirms the order.

7.System generates an order ID and updates inventory.

8.Pharmacy staff processes the order and marks it as "Processing."

9.User receives an order confirmation with details.

3.Doctor Consultation:

Use Case: Request Doctor Consultation

Actors: User, Doctor

Description: A user requests an online consultation with a doctor.

Flow:

1.User logs in to the system.

2.User navigates to the doctor consultation section.

3.User views available doctors and their profiles.

4.User schedules an appointment with a chosen doctor.

5.System sends a notification to the doctor.

6.Doctor confirms the appointment or suggests an alternate time.

7.At the scheduled time, the user and doctor conduct a video consultation.

4. Order Tracking:

Use Case: Track Order Status

Actors: User, Pharmacy Staff

Description: A user tracks the status of a placed order.

Flow:

1.User logs in to the system.

2.User navigates to the order tracking section.

3.User enters the order ID or selects a recent order.

4.System displays the current status of the order (e.g., processing, dispatched, delivered).

5.Pharmacy staff updates the order status as it progresses.

6. Inventory Management:

Use Case: Manage Medicine Inventory

Actors: Pharmacy Staff

Description: Pharmacy staff manages the stock of medicines.

Flow:

1.Pharmacy staff logs in to the system.

2.Pharmacy staff navigates to the inventory management section.

3.Pharmacy staff views the current stock of medicines.

4.Pharmacy staff receives automated alerts for low stock items.

5.Pharmacy staff updates stock levels when new medicines arrive.

Objects, classes, and relationships

Here you can find the CRC cards for my project.

https://echeung.me/crcmaker/?share=

Design class diagrams picturing classes, their attributes, and relations in the system(Optional)

