Introduction and Overview

***Online Pharmacy System***

This document serves as a comprehensive guide outlining the functional and non-functional requirements for the development of an online platform aimed at facilitating the management of medicines and patient interactions within a pharmacy setting.

The "Online Pharmacy" system will provide a user-friendly interface for the user to effectively manage and access pharmaceutical services. Users will have the authority to manage medicines, patient records, and access information about available products.

This document will outline the scope, actors, and general overview of the system, providing a clear understanding of its functionalities and objectives.

Requirements Overview

**- Functional Requirements**

**Authentication For User:**

Upon successful authentication, users shall be forwarded to the main dashboard.

Users shall have the option to register.

Users shall have the option to log in to the system.

Users shall have1 the option to log out of the system.

**User Functionalities:**

**Manage Medicines:**

Admin shall be able to add new medicines to the system, specifying details such as name, description, dosage, price, and availability.

Admin shall have the ability to update existing medicine details, including dosage, price, and availability.

Admin shall be able to delete medicines from the system.

Admin shall be able to view a list of all medicines stored in the system, along with their details.

**Manage Categories of Medicines:**

Admin shall have the capability to add new categories of medicines.

Admin shall be able to update existing categories, including changing category names or descriptions.

Admin shall be able to delete categories from the system.

Admin shall be able to view a list of all medicine categories stored in the system.

**Manage Patients:**

Admin shall be able to add new patients to the system, providing details such as name, contact information, and any relevant medical history.

Admin shall have the ability to update patient details, including contact information and medical history.

Admin shall be able to delete patient records from the system.

Admin shall be able to view a list of all patients registered in the system, along with their details.

**Manage cart:**

Admin shall be able to add new cart to the system.

**- Non-Functional Requirements**

**Security:**

The system shall implement robust authentication and authorization mechanisms to ensure the privacy and security of user data.

Access to sensitive functionalities and data shall be restricted based on user roles and permissions.

**Performance:**

The system shall be responsive and capable of handling concurrent user interactions without significant latency.

Response times for critical operations such as login, search, and request processing shall meet predefined performance targets.

**Reliability:**

The system shall be highly available, with minimal downtime for maintenance or updates.

Data integrity mechanisms shall be implemented to prevent data loss or corruption in the event of system failures.

**Usability:**

The user interface shall be intuitive and user-friendly, catering to users of varying technical proficiency.

Clear and informative error messages shall be provided to guide users in case of invalid inputs or system errors.

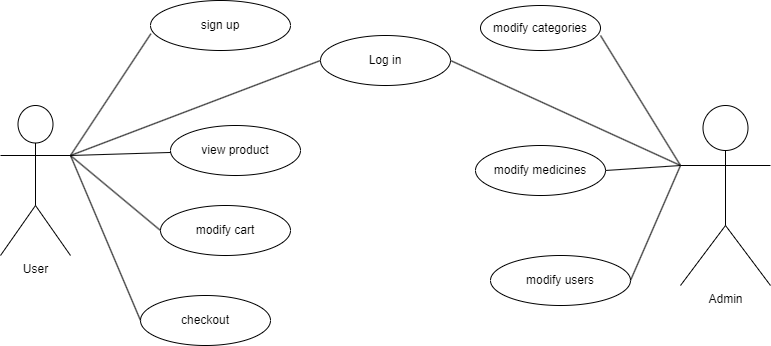
**Scalability:**

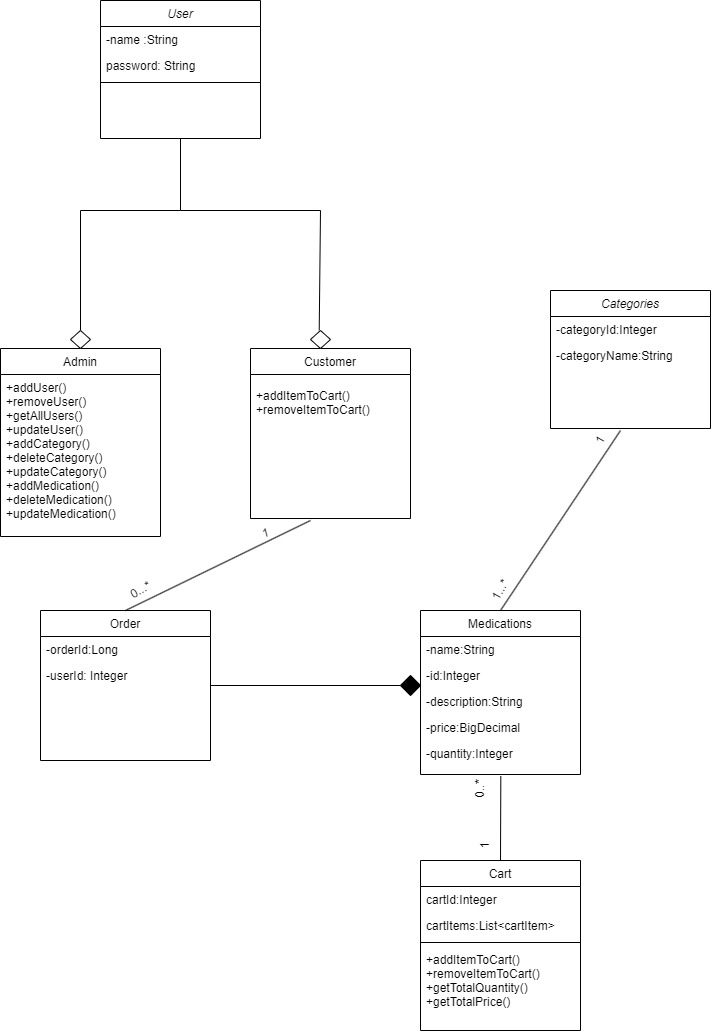
The system architecture shall be designed to accommodate future growth and increased user loads.

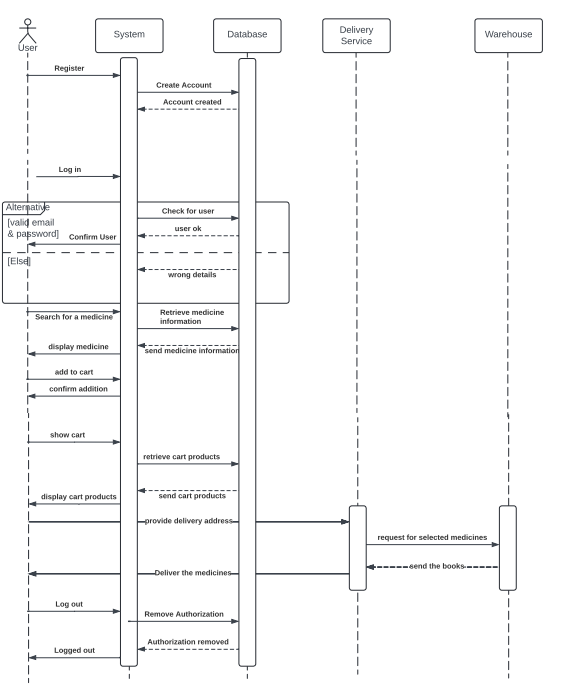
Scalability mechanisms such as load balancing and horizontal scaling shall be considered to ensure optimal performance under increased demand.

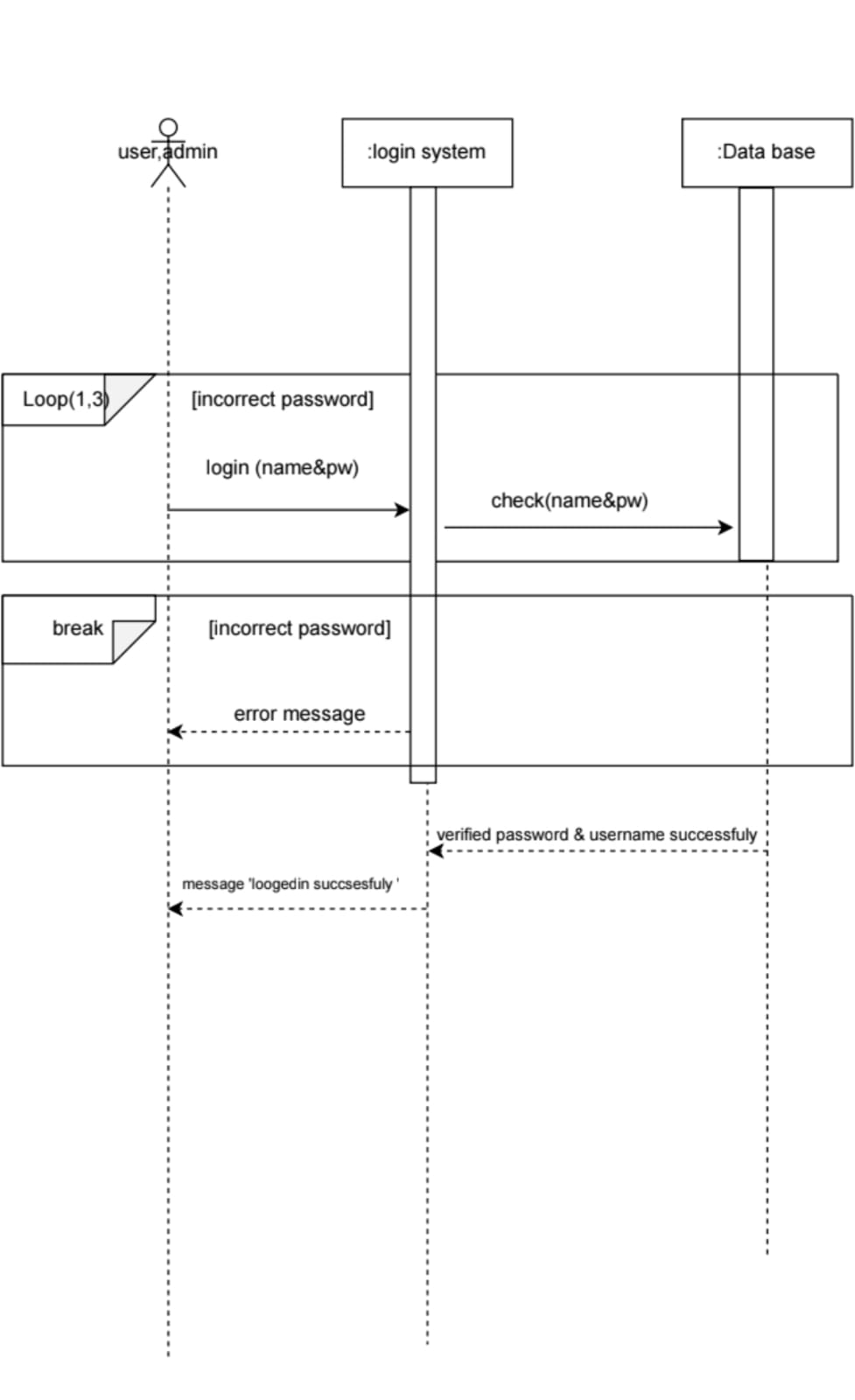
Diagrams

***USECASE:***

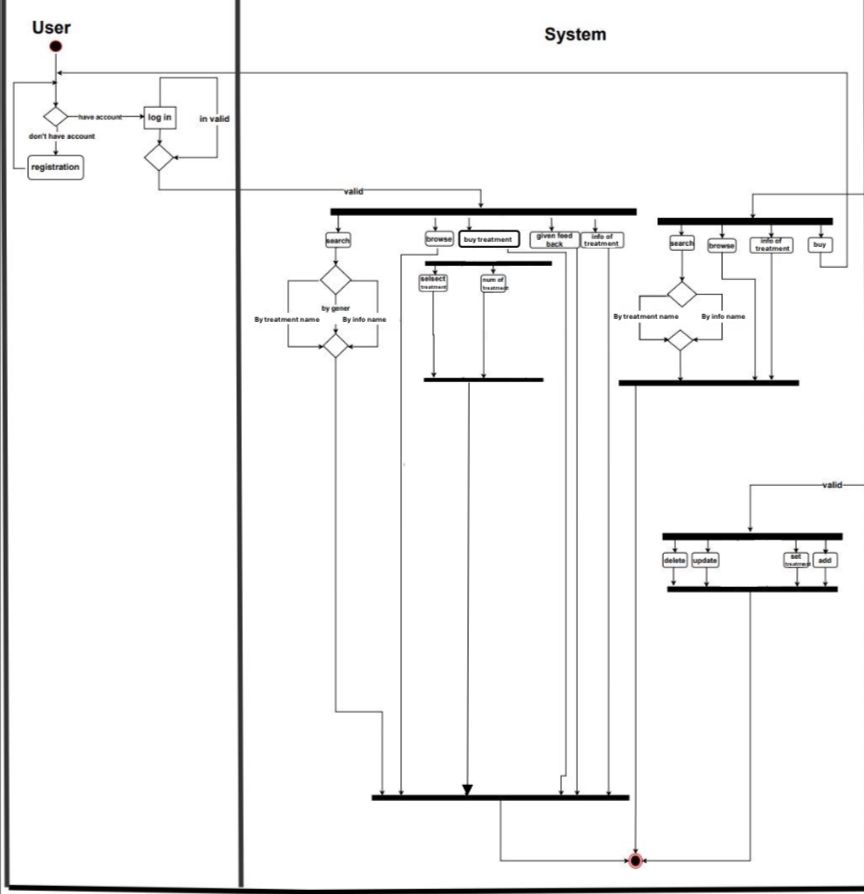


***CLASS DIAGRAM:***

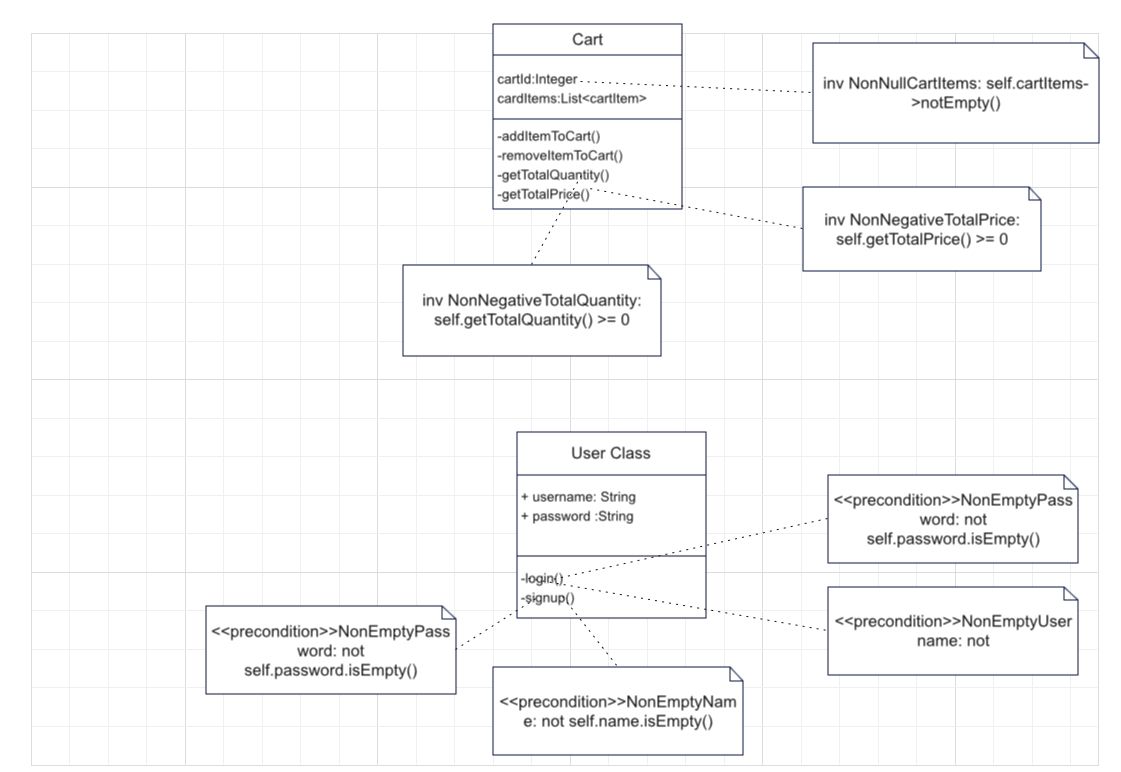
***Sequence Diagram:  
***

******

***Activity Diagram:***

******



***OCL(Object Constraint Language):***

-context Cart:

Inv: NonNullCartItems: self.cartItems->notEmpty()

Inv: NonNegativeTotalQuantity: self.getTotalQuantity() >= 0

Inv: NonNegativeTotalPrice: self.getTotalPrice() >= 0

-context CartItem:

Inv: NonNullCart: self.cart <> null

Inv: NonNullMedication: self.medication <> null

Inv: NonNegativeQuantity: self.quantity >= 0

- context Categories:

Inv: NonNullCategoryName: self.categoryName <> null

-context Login:

pre NonEmptyUsernameOrEmail: not self.usernameOrEmail.isEmpty()

pre NonEmptyPassword: not self.password.isEmpty()

-context Medication:

Inv: NonNullMedName: self.medName <> null

Inv: NonNullDescription: self.description <> null

Inv: NonNullPrice: self.price <> null

Inv: NonNegativeQuantity: self.quantity >= 0

-context Order:

Inv: NonNullUser: self.user <> null

Inv: NonNullOrderDate: self.orderDate <> null

Inv: NonNegativeTotalAmount: self.totalAmount >= 0

-context OrderDetails:

Inv: NonNullOrder: self.order <> null

Inv: NonNullMedication: self.medication <> null

Inv: NonNegativeQuantity: self.quantity >= 0

-context SignUp:

pre NonEmptyName: not self.name.isEmpty()

pre NonEmptyUsername: not self.username.isEmpty()

pre NonEmptyEmail: not self.email.isEmpty()

pre NonEmptyPassword: not self.password.isEmpty()

pre NonEmptyAddress: not self.address.isEmpty()

-context User:

Inv: NonNullUsername: self.username <> null

Inv: NonNullPassword: self.password <> null

Inv: NonNullEmail: self.email <> null

Inv: NonNullName: self.name <> null