**Software Requirements Specification (SRS)**

**Hospital Pharmacy Management System**

# **Introduction**

## ***Purpose***

This document specifies the software requirements for the hospital pharmacy management system. It serves as a guideline for the development, ensuring the system meets the necessary functionality and quality standards , we found that in hospitals it is found that the process of sending the prescription take much time , the work load on the pharmacy may cause overhead for the pharmacist , the conflicts handling that may cause a sever damage to the patient

## ***Document Conventions***

* Should assign the conflicts
* Should decrease workload on the pharmacist
* Should make the process of prospection and medicine taking more efficeint

## ***Intended Audience***

The intended audience includes the development team, project managers, hospital administrators, pharmacists, and stakeholders involved in the hospital pharmacy management system project.

## ***Additional Information***

The system integrates with existing hospital information systems such as electronic health records (EHR) and billing systems to streamline pharmacy operations and ensure data consistency.

## ***Contact Information/SRS Team Members***

* Project Manager: Abdelrahman Osheba , [aas101571@gu.edu.eg](mailto:aas101571@gu.edu.eg) ,01014404503
* Lead Developer: Ahmed Abo-El-Ela , [aam900095@gu.edu.eg](mailto:aam900095@gu.edu.eg) ,01066869194
* QA Lead: Ahmed Alaa-eldin ,ahmed\_alaa@gu.edu.eg,01028115587

# **Overall Description**

## ***Product Perspective***

The hospital pharmacy management system integrates with existing hospital information systems to provide a seamless workflow for medication dispensing and inventory management.

## ***Product Functions***

* **View Patient Information**: Access patient medical history, current medications, and diagnoses.
* **Place Medication Orders**: Process medication orders for patients.
* **Pharmacist Dashboard**: Manage medication orders and inventory.
* **Pharmacy Log-In**: Secure access for pharmacists to the system.

## ***User Classes and Characteristics***

* **Healthcare Providers**: Access patient information and place medication orders.
* **Pharmacists**: Manage medication orders, inventory, and access the pharmacist UI dashboard.

## ***Operating Environment***

* **Hardware**: Local machines running the MySQL database and Python scripts.
* **Software**: System will run on Windows, with components utilizing tkinter, mysql.connector, and subprocess.

## ***User Environment***

* **Healthcare Providers**: Typically use desktop computers or tablets within the hospital network.
* **Pharmacists**: Use desktop computers for detailed inventory management and tablets for quick access on the go.

## ***Design and Implementation Constraints***

* Specific use of MySQL for data storage.
* There can not be more than doctor, patient , drug, diagnosis ,pharmacist with the same id as it is unique identifier

## ***Assumptions and Dependencies***

* System will be primarily used within a hospital environment with stable network connectivity.
* Integration with existing hospital information systems will be possible.

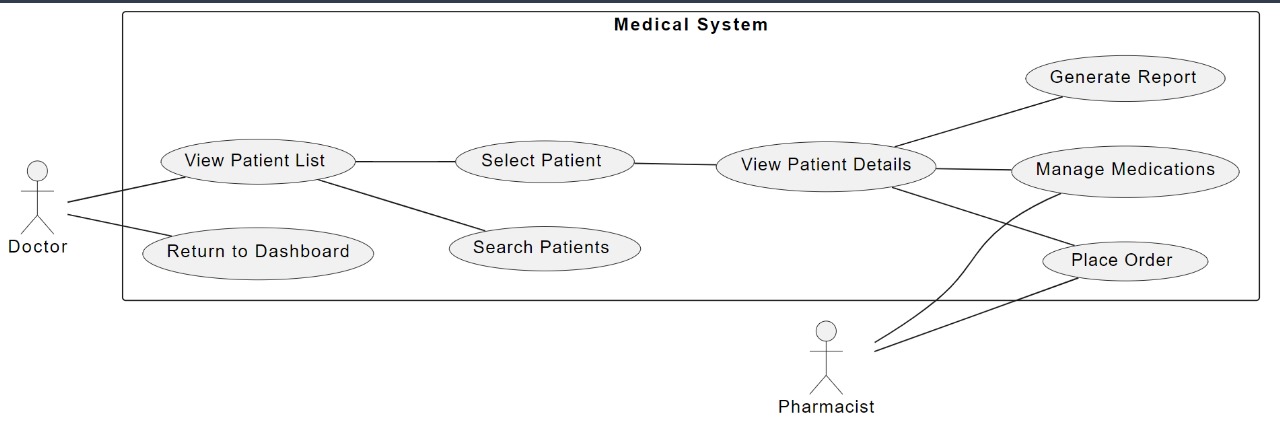


Figure Use Case Diagram

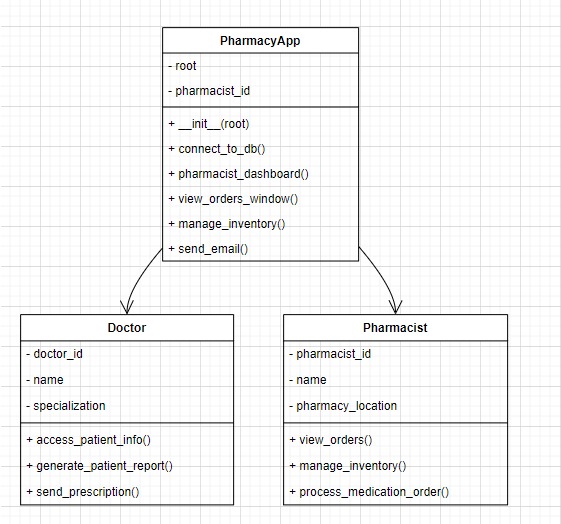


Figure Class Diagram

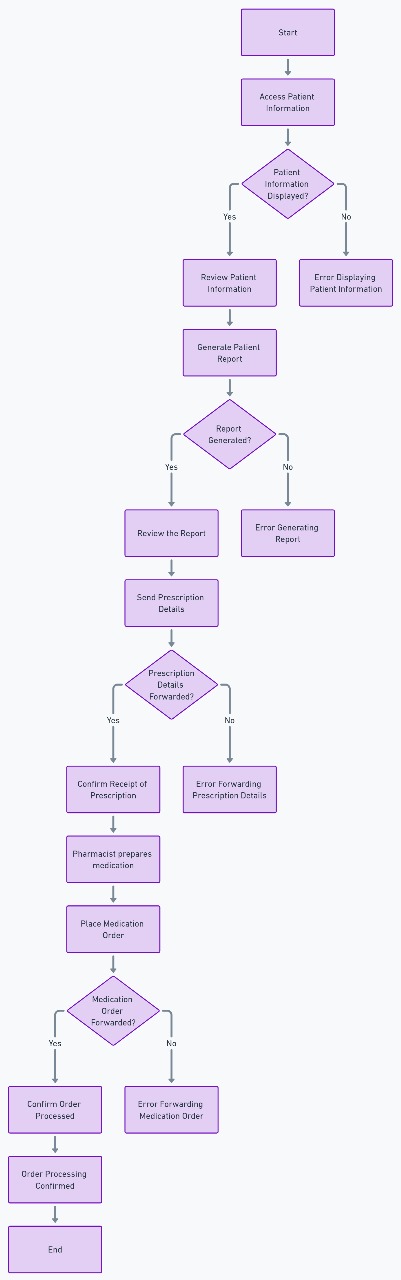


Figure Activity Diagram

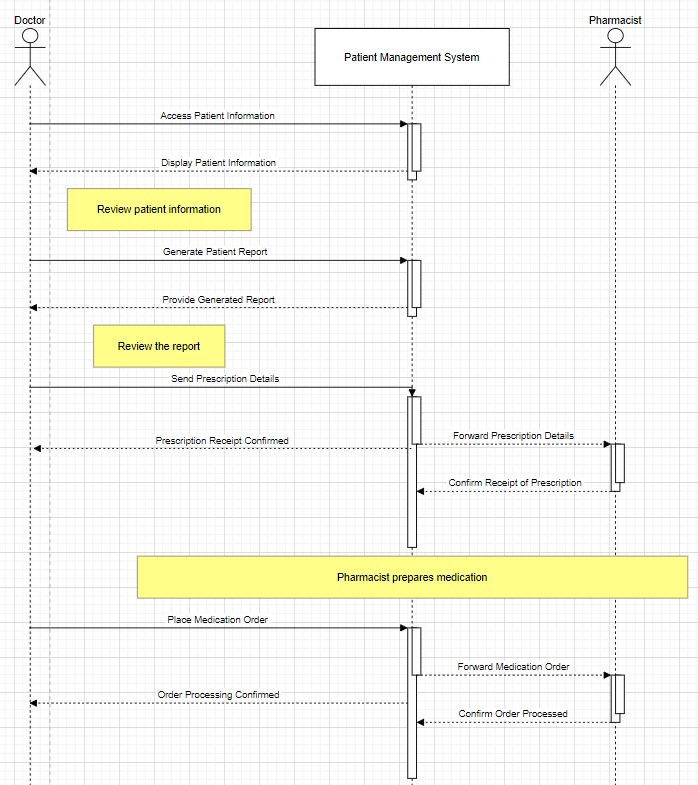


Figure Sequence Diagram

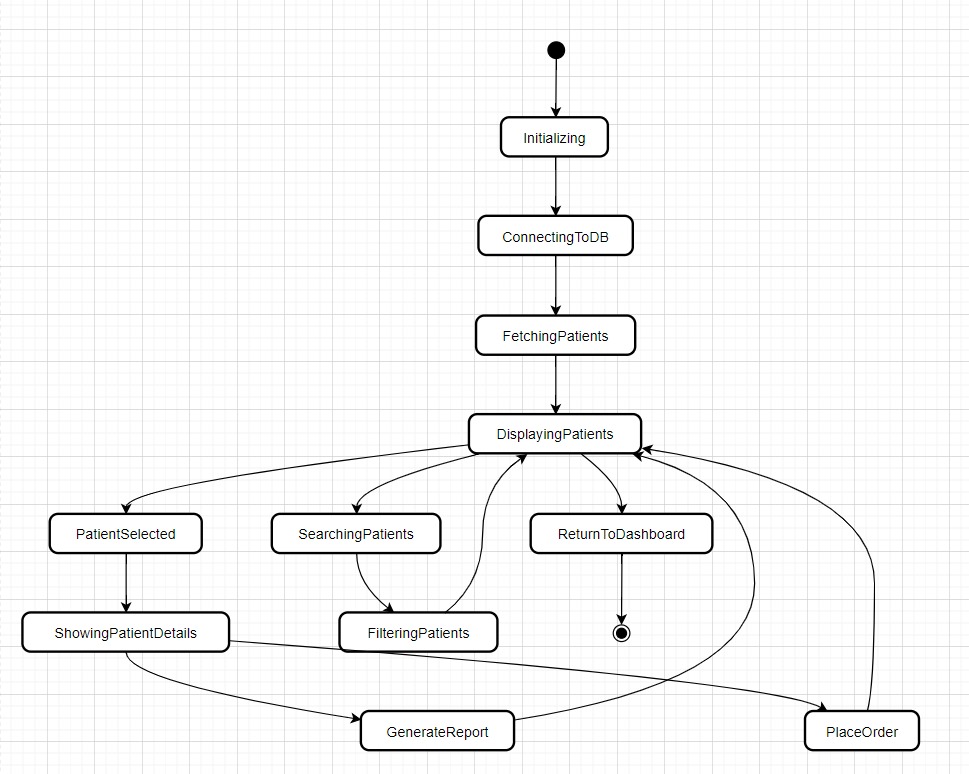


Figure State Machine

# **External Interface Requirements**

### *User Interfaces*

* Consistent layout with standard buttons and error messages via messagebox.
* Screens: Doctor Dashboard, Add Patient, Place Order, Patient Information Editor.
* Font: Helvetica size 12 weight bold for buttons.
* Entry fields: Standard width and height.

### *Hardware Interfaces*

* Local machine where the MySQL database and Python scripts are running.
* Interaction with local MySQL database for fetching, inserting, and updating patient data.

### *Software Interfaces*

* Integration with MySQL database using SQL queries.
* Use of tkinter for GUI components, mysql.connector for database connections.
* Streamlit web interface

### *Communication Protocols and Interfaces*

* Use of SQL for database operations.
* No specific requirements for email, web browser, or electronic forms.

# **System Features**

## ***System Feature: View Patient Information***

#### **Description and Priority**

This feature allows healthcare providers to access detailed information about a patient, including their medical history, current medication, diagnoses, and other relevant data. It is of high priority as it provides crucial information for making informed medical decisions.

## ***Action/Result***

* The user selects the "View Patient Information" option from the dashboard.
* The system prompts the user to enter the patient's ID or Name to search bar.
* Upon entering the ID and confirming, the system retrieves the patient's information from the database.
* The system displays the patient's information in a clear and organized manner, including medical history, current medication, diagnoses, etc.

## ***Functional Requirements***

* **Input**: User selects the "View Patient Information" option and enters the patient's ID.
* **Validation**: The system verifies that the entered ID is valid and exists in the database.
* **Database Query**: The system retrieves the patient's information from the database based on the entered ID.
* **Display**: The system presents the patient's information in a structured format.
* **Error Handling**: If the entered patient ID does not exist or if there is an error in retrieving the data, the system provides appropriate error messages.

## ***System Feature: Place Medication Order***

## ***Description and Priority***

This feature enables healthcare providers to place medication orders for patients, ensuring timely access to required medications. It is of high priority as it directly impacts patient care and treatment.

## ***Action/Result***

* The user selects the "Place Medication Order" option from the dashboard.
* The system prompts the user to enter the patient's ID and the number of drugs to order.
* After entering the required information, the system displays entry fields for each drug, including drug ID and quantity.
* The user enters the details for each drug and submits the order.
* The system validates the order, checks for drug conflicts, and updates the inventory accordingly.

## ***Functional Requirements***

* **Input**: User selects the "Place Medication Order" option and enters patient ID and number of drugs.
* **Validation**: Validate input fields for patient ID and number of drugs.
* **Drug Entry**: Display entry fields for each drug including drug ID and quantity.
* **Order Submission**: Allow the user to submit the medication order after entering drug details.
* **Order Validation**: Validate the order, check for drug conflicts, and ensure availability of stock.
* **Inventory Update**: Update medication inventory based on the placed order.

## ***System Feature: Pharmacist UI Dashboard***

#### **Description and Priority**

This feature provides pharmacists with a user interface dashboard to manage medication orders and inventory efficiently. It is of high priority as it enables streamlined order processing and ensures adequate stock management.

#### **Action/Result**

* The pharmacist logs in to the system using their credentials.
* Upon successful login, the system displays the pharmacist UI dashboard.
* The dashboard presents options to view and manage medication orders and inventory.
* Pharmacists can view incoming orders, process them, and update the order status accordingly.
* They can also monitor medication inventory, check stock levels, and update inventory as needed.

#### **Functional Requirements**

* **Authentication**: Require pharmacists and doctor to log in with valid credentials.
* **Dashboard Display**: Present a user-friendly dashboard interface upon successful login.
* **Order Management**: Provide options to view incoming orders, process them, and update order status.
* **Inventory Management**: Enable pharmacists to monitor medication inventory and manage stock levels.
* **Error Handling**: Display appropriate error messages for invalid login attempts or system errors.
* **Navigation**: Allow pharmacists to navigate between different sections of the dashboard and return to the main menu.

## ***System Feature: Pharmacy Sign-In***

#### **Description and Priority**

This feature allows pharmacists to sign in to the system using their credentials to access the pharmacist UI dashboard. It is of high priority as it provides secure access to essential pharmacy management functions.

#### **Action/Result**

* The pharmacist accesses the pharmacy sign-in page.
* The system prompts the pharmacist to enter their credentials (username and password).
* Upon entering valid credentials and submitting, the system grants access to the pharmacist UI dashboard.
* In case of invalid credentials, appropriate error messages are displayed, and access is denied.

#### **Functional Requirements**

* **Authentication**: Prompt pharmacists to enter their username and password.
* **Validation**: Verify entered credentials against the database.
* **Access Granting**: Grant access to the pharmacist UI dashboard upon successful authentication.
* **Error Handling**: Display error messages for invalid login attempts.
* **Secure Access**: Ensure secure access to the pharmacist UI dashboard.

# **Nonfunctional Requirements**

### *Performance Requirements*

* The system should handle concurrent access by multiple users without significant performance degradation.

### *Safety Requirements*

* The system should ensure data integrity and prevent unauthorized access to patient information and medication orders.

### *Security Requirements*

* Ensure secure access to the system and protection of patient data through Password of Doctor.

***Software Quality Attributes***

* **Adaptability**: The system should be adaptable to changes in hospital pharmacy processes without extensive reprogramming.
* **Availability**: The system should be operational with minimal downtime.
* **Correctness**: The system must accurately record medication orders and inventory levels.
* **Flexibility**: The system should handle various medication types and inventory management scenarios.
* **Interoperability**: The system should integrate with existing hospital information systems.
* **Maintainability**: The system should be designed for easy maintenance.
* **Portability**: The system should be accessible across different devices and platforms.
* **Reliability**: The system should perform reliably under varying workload conditions.
* **Reusability**: Components should be reusable in other hospital pharmacy applications.
* **Robustness**: The system should handle errors gracefully.
* **Testability**: The system should be easily testable.
* **Usability**: The system should feature an intuitive user interface.

### *Project Documentation*

* **Requirements Specification Document**: Detailed functional and non-functional requirements.
* **Design Document**: System architecture and data flow diagrams.
* **Implementation Plan**: Development steps, timelines, and resource allocations.
* **Testing Plan**: Testing strategies, test cases, and criteria for test success.
* **Maintenance Plan**: Procedures for handling bug fixes, updates, and enhancements.

### *User Documentation*

* **User Manuals**: Detailed guides on using the system.
* **Online Help**: Integrated context-sensitive assistance.
* **Tutorials**: Interactive sessions guiding users through common workflows.
* **Quick Reference Guides**: Concise materials for frequently performed tasks.
* **Training Materials**: Slides, videos, and hands-on exercises.

## **Other Requirements**

## Appendix A: Terminology/Glossary/Definitions List

* **Adaptability**: The capability of the system to adjust to changes in hospital pharmacy processes and configurations without requiring extensive reprogramming. This allows for easy customization to accommodate evolving workflows.
* **Availability**: The degree to which the system is operational and accessible when required for use, characterized by minimal downtime and ensuring continuous service to pharmacy staff.
* **EHR (Electronic Health Records)**: A digital version of a patient's paper chart, which contains comprehensive patient history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory test results.
* **SQL (Structured Query Language)**: A standardized programming language used for managing and manipulating relational databases. SQL is essential for performing tasks such as querying data, updating records, and managing database structures.

## Appendix B: To be Determined

Details for this appendix will be defined as the project progresses and additional requirements or terms need to be clarified.

### Risk Management Plan for Hospital Pharmacy Management System

#### **1. Introduction**

This Risk Management Plan outlines the process for identifying, assessing, and mitigating risks associated with the development and implementation of the hospital pharmacy management system as defined in the Software Requirements Specification (SRS). The goal is to ensure that the project meets its objectives while minimizing potential negative impacts on operations, patient care, and data integrity.

#### **2. Risk Identification**

The first step in the risk management process involves identifying potential risks that could affect the project. These risks are categorized into technical, operational, and compliance-related risks.

##### **2.1 Technical Risks**

* **Integration Issues**: Challenges in integrating with existing hospital information systems (EHR, billing).
* **System Performance**: Potential slowdowns or crashes under high user load.
* **Data Security**: Risks of unauthorized access to sensitive patient information.

##### **2.2 Operational Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk ID | Risk Description | Likelihood (1-5) | Impact (1-5) | Risk Score (L\*I) | Mitigation Strategy |
| R1 | Integration Issues | 3 | 4 | 12 | Conduct thorough integration testing and develop contingency plans. |
| R2 | System Performance | 2 | 5 | 10 | Optimize system performance and conduct stress testing. |
| R3 | Data Security | 4 | 5 | 20 | Implement robust security protocols and conduct regular audits. |
| R4 | User Training | 3 | 3 | 9 | Develop comprehensive training programs and materials. |
| R5 | Resource Availability | 2 | 4 | 8 | Ensure timely procurement and availability of resources. |
| R6 | Project Timeline | 3 | 4 | 12 | Use agile project management techniques and regular status updates. |
| R7 | Regulatory Compliance | 2 | 5 | 10 | Regularly review compliance requirements and conduct audits. |
| R8 | Data Integrity | 3 | 4 | 12 | Implement data validation rules and perform regular data quality checks. |

* **User Training**: Insufficient training leading to improper use of the system.
* **Resource Availability**: Shortages of necessary hardware or software components.
* **Project Timeline**: Delays in achieving project milestones and final delivery.

##### **2.3 Compliance Risks**

* **Regulatory Compliance**: Non-compliance with healthcare regulations (e.g., HIPAA).
* **Data Integrity**: Risks related to inaccurate or incomplete data entry.

#### **3. Risk Assessment**

Each identified risk is evaluated based on its likelihood and impact, using a risk matrix.

#### **4. Mitigation Strategies**

For each identified risk, specific mitigation strategies are implemented to reduce its likelihood and impact.

##### **4.1 Integration Issues (R1)**

* **Mitigation**: Conduct comprehensive testing of all integration points with EHR and billing systems before deployment.
* **Contingency**: Develop fallback procedures and ensure technical support availability during the integration phase.

##### **4.2 System Performance (R2)**

* **Mitigation**: Optimize system performance and conduct thorough stress testing to ensure the system can handle peak loads.
* **Contingency**: Implement system monitoring and scalable infrastructure to manage performance issues dynamically.

##### **4.3 Data Security (R3)**

* **Mitigation**: Implement encryption, access controls, and conduct regular security audits to protect sensitive data.
* **Contingency**: Develop incident response plans and conduct regular security drills.

##### **4.4 User Training (R4)**

* **Mitigation**: Develop detailed training manuals, conduct hands-on training sessions, and provide ongoing support.
* **Contingency**: Implement a helpdesk or support hotline to assist users with immediate issues.

##### **4.5 Resource Availability (R5)**

* **Mitigation**: Plan for resource needs well in advance and maintain an inventory of critical components.
* **Contingency**: Establish relationships with multiple suppliers to mitigate supply chain issues.

##### **4.6 Project Timeline (R6)**

* **Mitigation**: Use agile project management methods to keep the project on track and adaptable to changes.
* **Contingency**: Conduct regular progress reviews and adjust the project plan as needed.

##### **4.7 Regulatory Compliance (R7)**

* **Mitigation**: Stay updated on regulatory changes and incorporate compliance checks into the project workflow.
* **Contingency**: Engage with legal and compliance experts to ensure adherence to regulations.

##### **4.8 Data Integrity (R8)**

* **Mitigation**: Implement data validation checks at the point of entry and conduct regular audits of data quality.
* **Contingency**: Develop procedures for data correction and conduct regular training on data entry best practices.

#### **5. Monitoring and Review**

The risk management plan is a living document that will be reviewed and updated regularly throughout the project lifecycle. Regular risk assessment sessions will be conducted to identify new risks and assess the effectiveness of mitigation strategies.

* **Frequency**: Monthly risk review meetings.
* **Participants**: Project Manager, Lead Developer, QA Lead, and key stakeholders.
* **Reporting**: Risks and their statuses will be documented and reported to senior management.

#### **6. Roles and Responsibilities**

* **Project Manager**: Overall responsibility for risk management and ensuring that mitigation strategies are implemented.
* **Lead Developer**: Address technical risks related to system performance and integration.
* **QA Lead**: Ensure that all testing and quality assurance processes mitigate identified risks.
* **Compliance Officer**: Monitor regulatory compliance and data integrity.

#### 

#### **Conclusion**

In conclusion, the Software Requirements Specification (SRS) for the hospital pharmacy management system provides a comprehensive framework for developing a robust and efficient solution that integrates seamlessly with existing hospital information systems. This document outlines essential functional and non-functional requirements, ensuring the system supports key operations such as patient information management, medication ordering, and inventory control. By addressing the needs of healthcare providers and pharmacists, the system aims to enhance patient care and streamline pharmacy workflows.

The risk management plan further underscores the commitment to delivering a secure, reliable, and compliant system by identifying potential risks and establishing effective mitigation strategies. Regular monitoring and adaptive project management approaches will ensure the project remains on track, delivering a high-quality solution that meets the stringent standards of the healthcare industry.

Ultimately, this SRS serves as a vital guide for all stakeholders, fostering a shared understanding and clear expectations, which are crucial for the successful development and implementation of the hospital pharmacy management system.

**Content**

[**Introduction** 1](#_Toc167688118)

[***Purpose*** 1](#_Toc167688119)

[***Document Conventions*** 1](#_Toc167688120)

[***Intended Audience*** 1](#_Toc167688121)

[***Additional Information*** 1](#_Toc167688122)

[***Contact Information/SRS Team Members*** 1](#_Toc167688123)

[**Overall Description** 2](#_Toc167688124)

[***Product Perspective*** 2](#_Toc167688125)

[***Product Functions*** 2](#_Toc167688126)

[***User Classes and Characteristics*** 2](#_Toc167688127)

[***Operating Environment*** 2](#_Toc167688128)

[***User Environment*** 2](#_Toc167688129)

[***Design and Implementation Constraints*** 3](#_Toc167688130)

[***Assumptions and Dependencies*** 3](#_Toc167688131)

[**External Interface Requirements** 8](#_Toc167688132)

[*User Interfaces* 8](#_Toc167688133)

[*Hardware Interfaces* 8](#_Toc167688134)

[*Software Interfaces* 8](#_Toc167688135)

[*Communication Protocols and Interfaces* 8](#_Toc167688136)

[**System Features** 8](#_Toc167688137)

[***System Feature: View Patient Information*** 8](#_Toc167688138)

[***Action/Result*** 8](#_Toc167688139)

[***Functional Requirements*** 9](#_Toc167688140)

[***System Feature: Place Medication Order*** 9](#_Toc167688141)

[***Description and Priority*** 9](#_Toc167688142)

[***Action/Result*** 9](#_Toc167688143)

[***Functional Requirements*** 9](#_Toc167688144)

[***System Feature: Pharmacist UI Dashboard*** 10](#_Toc167688145)

[***System Feature: Pharmacy Sign-In*** 11](#_Toc167688146)

[**Nonfunctional Requirements** 11](#_Toc167688147)

[*Performance Requirements* 11](#_Toc167688148)

[*Safety Requirements* 11](#_Toc167688149)

[*Security Requirements* 12](#_Toc167688150)

[*Project Documentation* 12](#_Toc167688151)

[*User Documentation* 13](#_Toc167688152)

[**Other Requirements** 13](#_Toc167688153)

[Appendix A: Terminology/Glossary/Definitions List 13](#_Toc167688154)

[Appendix B: To be Determined 13](#_Toc167688155)

[Risk Management Plan for Hospital Pharmacy Management System 13](#_Toc167688156)