EGERTON UNIVERSITY

**FACULTY OF SCIENCE**

**DEPARTMENT OF COMPUTER SCIENCE**

**SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT**

**FOR**

**PHARMACY MANAGEMENT AND INVENTORY SYSTEM**

**PREPARED BY: KIPKOSGEI COLLINS**

**REG. NO: S13/03582/21**

**PROGRAMME: BSc. COMPUTER SCIENCE**

**UNIT: COMP 493**

**PROJECT SUPERVISOR: MR. KEMEI**

**PROJECT PERIOD: SEPTEMBER 2024 - APRIL 2025**

# INTRODUCTION

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the Pharmacy Management and Inventory System. This system is designed to streamline operations within pharmacies by providing a robust platform for inventory management, sales processing, and administrative tasks. It aims to enhance efficiency, reduce human error, and ensure regulatory compliance in pharmacy operations.

The SRS document is vital as it provides a clear and structured framework for the project. His ensures that all stakeholders have a shared and deep understanding of its objectives, scope and requirements. By outlining these down, the document serves as a reference throughout the development lifecycle.

# Overall Description

## System Perspective

The Pharmacy Management and Inventory System introduces a collaborative digital solution to bridge gaps in inventory management, sales tracking, and operational efficiency. It eliminates manual processes by automating inventory updates and sales records, enabling staff to access real-time information from anywhere. Designed with inclusivity and scalability in mind, the system emphasizes accountability, transparency, and real-time data insights to foster seamless operations.

## User Objectives

* **Efficient Inventory Management:-** Maintain an up-to-date inventory of medicines and other products, with automated stock updates and alerts for low-stock or expired items.
* **Seamless Sales Processing:-** Simplify billing and sales transactions with an intuitive interface, ensuring smooth operations at the point of sale.
* **Real-Time Notifications:-** Provide real-time updates for stock levels, order statuses, and system alerts, ensuring staff remain informed.
* **Comprehensive Reporting:-** Generate detailed reports on sales, inventory, and operational performance to support decision-making.
* **Enhanced Security:-** Implement role-based access control and encrypted data storage to safeguard sensitive information.

## Operating Environment

* **Compatibility:-**The application supports all major operating systems, including Windows, macOS, and Linux. It is optimized for performance across various web browsers such as Chrome, Brave, Safari, Firefox, and Microsoft Edge.
* **Technology Stack:**
  + **Frontend:** Angular and Tailwind CSS for a responsive and visually appealing user interface.
  + **Backend:** Spring Boot for robust and scalable business logic.
  + **Databases:** PostgreSQL for structured data and MongoDB for logs and notifications.
* **Connectivity:** Requires stable internet connectivity for real-time data synchronization and communication.
* **Security:**
  + **HTTPS:** Ensures encrypted communication between client and server to safeguard data during transmission.
  + **Encryption:** Sensitive information is encrypted at rest and in transit to prevent unauthorized access.
  + **Authentication and Authorization:** Incorporates secure authentication mechanisms (e.g., JWT, Keycloak) and role-based access control.

# Scope

## In Scope

* **Inventory Management:** Automate stock monitoring with real-time updates and notifications for low-stock or expired items.
* **Sales Processing:** Streamline billing and customer transactions.
* **Notification System:** Real-time notifications for stock alerts and order updates.
* **Reporting Tools:** Generate detailed reports for inventory and sales analytics.
* **User Management:** Role-based access control for administrators and staff.
* **Secure Data Handling:** Encrypt sensitive information and implement robust security protocols.

## Out of Scope

* **Direct Integration with Suppliers:** The system will not handle direct procurement or supplier management at this stage.
* **Mobile Application Development:** The project will focus solely on a web-based platform, with no mobile application included at this stage.
* **Pharmacy Staff Training:** While user guides may be provided, staff training is not within the project scope.

## Out of Scope

* **Direct Integration with Suppliers:** The system will not handle direct procurement or supplier management at this stage.
* **Mobile Application Development:** The project will focus solely on a web-based platform, with no mobile application included at this stage.
* **Pharmacy Staff Training:** While user guides may be provided, staff training is not within the project scope.

# FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

## Functional Requirements

### User Management

* The system shall allow admins to create, edit, and delete user accounts.
* The system shall provide secure authentication using JWT.
* The system shall implement role-based access control using Keycloak.

### Inventory Management

* The system shall allow users to add, update, and delete medicines.
* The system shall track stock levels and send alerts for low stock.
* The system shall maintain a database of suppliers and purchase orders.

### Sales and Billing

* The system shall generate invoices for customer purchases.
* The system shall store sales records for reporting purposes.

### Reporting and Notifications

* The system shall generate inventory and sales performance reports.
* The system shall send notifications for:
  + Low stock.
  + Medicines nearing expiration.

### Security

* The system shall use JWT for secure authentication.
* The system shall integrate Keycloak for user management and authorization.
* Sensitive data shall be encrypted.

## 2.5.2 Non-Functional Requirements

Non-Functional Requirements

### Performance

* The system shall handle up to 500 concurrent users without performance degradation.
* API response time shall not exceed 1 second under normal conditions.

### Scalability

* The system shall support horizontal scaling for increased user demand.

### Usability

* The UI shall be intuitive and responsive.
* Tailwind CSS shall be used for a consistent and modern design.

### Reliability

* The system shall maintain 99.9% uptime.

### Maintainability

* The codebase shall follow industry standards and include comprehensive documentation.

### Security

* The system shall comply with OWASP guidelines to prevent common vulnerabilities.

# PROJECT ARCHITECTURE

## Architecture Overview

The Pharmacy Management and Inventory System follows a three-tier architecture model, comprising the frontend, backend, and database layers. This architecture is designed to ensure modularity, scalability, and maintainability of the system. Each tier has distinct responsibilities, and their seamless interaction ensures the platform delivers a responsive, secure, and user-friendly experience.

### 1. Frontend

The Frontend represents the user interface and is the system's client-facing component. It is responsible for providing a visually appealing, intuitive, and interactive experience for the users.

* **Technologies:**
  + Angular application with modular design.
  + Tailwind CSS for responsive and modern styling.
* **Key Features:**
  + Dynamic dashboards tailored for different user roles (admin, staff).
  + User-friendly navigation for inventory management, sales processing, and report generation.
  + Real-time updates via API calls for notifications, stock alerts, and system updates.
  + Responsive design to ensure compatibility with desktops, tablets, and mobile devices.

### 2. Backend

The backend layer handles the business logic and serves as the intermediary between the frontend and the database. It processes requests from the frontend, performs necessary computations or data manipulations, and returns the appropriate responses.

* **Technologies:**
  + REST APIs built using Spring Boot.
  + Business logic separated into service layers.
* **Responsibilities:**
  + **Authentication and Authorization:** Securely manage user accounts, roles, and permissions using JWT and Keycloak.
  + **Business Logic:** Implement inventory management, sales processing, and notification handling.
  + **API Integration:** Serve as the communication bridge between the frontend and database.
  + **Data Validation:** Validate and sanitize user inputs to ensure system integrity and prevent security vulnerabilities.

### 3. Database

The database layer manages all persistent data for the system. It securely stores user profiles, inventory details, sales records, and notification logs, ensuring the platform remains reliable and efficient.

* **Technology:**
  + PostgreSQL for structured data.
  + MongoDB for logs and notifications.
* **Responsibilities:**
  + Store and retrieve data such as medicine inventory, sales records, and user details.
  + Maintain logs for notifications and audit purposes.
  + Ensure data integrity and security through schema validation and encrypted storage.
  + Optimize performance through indexing and query optimization to support large-scale usage.

### Integration and Interaction

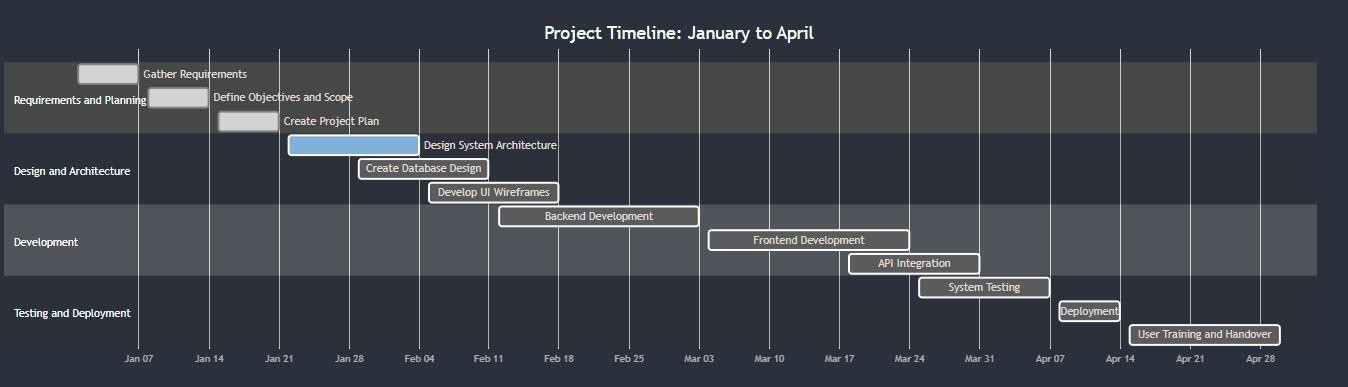
The three layers interact seamlessly to ensure a consistent and reliable user experience:

* The frontend sends user actions (e.g., adding a medicine, processing a sale) to the backend via REST APIs.
* The backend validates the requests, performs necessary computations or updates, and interacts with the databases to store or retrieve data.
* The frontend receives responses from the backend and dynamically updates the user interface, ensuring real-time interaction and responsiveness.

This architecture provides a robust foundation for the Pharmacy Management and Inventory System, ensuring scalability, security, and user satisfaction.

# TIMELINES

**Gantt Chart**

****

# PROJECT TEAM

## Developer:

* **Responsibilities:**
  1. Handles the design, development, and testing of the Pharmacy Management and Inventory System.
  2. Ensures the system is functional, scalable, and user-friendly.
  3. Implements security measures, resolves bugs, and adds new features as required.

## Stakeholders:

### 1. End Users:

* **Pharmacy Admins:**
  + Manage the overall operations of the system, including inventory and sales.
  + Generate reports and handle user accounts.
* **Pharmacy Staff:**
  + Use the system to process sales, track inventory, and access notifications.
  + Update stock levels and notify admins about low-stock or expired items.

### 2. IT Administrators:

* **Responsibilities:**
  + Oversee system maintenance and ensure seamless operation of the platform.
  + Perform regular updates, monitor uptime, and address technical issues.
  + Manage user accounts, verify data integrity, and ensure the security of sensitive information.

# RESOURCES

## ​Technical Resources

### Hardware

* + **Development Machines**: Laptops/desktops (Intel i5, 8GB RAM).
  + **Servers**: Cloud hosting (AWS).
  + **Backup Storage**: External drives/cloud storage.

### Software

* + **Development Tools**: Intellij,Docker,Jenkins.
  + **Frontend**: Angular, Tailwind CSS.
  + **Backend**: Spring Boot,Postgres,MongoDB
  + **Hosting**: AWS

## ​Human Resources

* + **Developer (Me)**: Full-stack development (frontend, backend, database, integration, deployment).
  + **Supervisor**: Project oversight and guidance.
  + **QA Testers**(my friends): Unit, integration, and user acceptance testing.

# ASSUMPTIONS AND DEPENDENCIES

## Assumptions

1. Users will actively engage with the platform.
2. Internet access will be available for all users.
3. Users are assumed to have basic knowledge of navigating web platforms.
4. Alumni, students and admin will actively contribute resources, mentorship opportunities, scholarships and event information.

## Dependencies

1. Access to hosting platforms and development tools.
2. Stakeholder cooperation to refine requirements.
3. Integration with third party services e.g. email providers
4. Adequate hardware and software resources for development and testing are necessary to meet performance and quality standards.

# RISK MANAGEMENT

To mitigate potential risks that may arise during the project, the following strategies will be implemented:

## Risk Identification

Conduct regular risk assessments throughout the project lifecycle to identify new risks or changes to existing risks.

## Risk Analysis

Evaluate the impact and likelihood of identified risks to prioritize them effectively.

## Risk Management

* **Technical Risks:**
  + **Mitigation:** Regular code reviews and pair programming to address bugs early.

## Resource Availability:

* + **Mitigation:** Identify backup resources (e.g., additional developers or tools) to avoid delays.

## User Engagement:

* + **Mitigation:** Regular feedback sessions during the development phase to ensure the platform meets user needs.

## Timeline Delays:

* + **Mitigation:** Implement Agile methodology to allow for flexibility in task management and prioritize critical features.

## Budget Overruns:

* + **Mitigation:** Keep track of all expenses and establish a contingency budget of 10% of total costs.

# IMPLEMENTATION PLAN

The system will adopt the Agile software development framework, which emphasizes flexibility, iterative development, and close collaboration with stakeholders.

## ​Rollout Strategy

The system will be deployed in a phased approach to ensure smooth integration with existing workflows and systems while minimizing disruption for users.

## Phase 1: Pilot Testing

* + - Roll out the system to a small group of users, including selected students, alumni, and academic staff.
    - Gather initial feedback on usability, functionality, and performance.
    - Identify and resolve any critical issues before full deployment.

## Phase 2: Limited Deployment

* + - Expand access to a broader group within the university, focusing on one department or constituency at a time.
    - Monitor usage and ensure the system integrates seamlessly with existing workflows.

## Phase 3: Full Deployment

* + - Launch the system for all intended users, ensuring all features are operational and tested.
    - Provide ongoing support and collect feedback for continuous improvement.

## ​Training Plan

Comprehensive training will be conducted to familiarize users with the platform and ensure they can utilize its features effectively.

## Workshops and Training Sessions

* + - Organize separate sessions tailored to different user groups (students, alumni, academic staff, and administrators).
    - Provide hands-on demonstrations covering essential tasks, such as mentorship matching, accessing resources, and managing user profiles.

## Training Materials

* + - Develop user manuals, video tutorials, and step-by-step guides for onboarding.
    - Provide FAQs and quick-reference sheets for common tasks.

## Support Channels

* + - Offer live support through a helpdesk during the initial rollout phase.
    - Set up email and chatbot support for resolving user queries post-launch.

## ​Maintenance Plan

To ensure the platform remains functional, secure, and up-to-date, a structured maintenance plan will be implemented.

## Corrective Maintenance

* + - Identify and resolve bugs or issues reported by users promptly.
    - Monitor system logs to detect and address errors, crashes, or downtime.

## Perfective Maintenance

* + - Introduce new features and improvements based on user feedback.
    - Optimize existing features for better performance and usability (e.g., improving loading times or refining the user interface).

## Adaptive Maintenance

* + - Ensure compatibility with emerging technologies, such as new browsers, operating systems, or device types.
    - Regularly update software libraries and dependencies to maintain security and compliance.
    - Prepare the platform for scalability to handle an increasing user base or additional functionalities in the future.

## Implementation Timeline

The implementation plan aligns with the project timeline and is distributed as follows:

1. **Weeks 13–14**: Pilot Testing
   * Deploy the system to a small group of users.
   * Collect and analyze feedback to refine the system.
2. **Weeks 15–16**: Full Deployment
   * Launch the system for all users.
   * Provide training sessions and ongoing support during the initial adoption period.
3. **Post-Launch (Week 17 Onwards)**: Monitoring and Maintenance
   * Regularly monitor the system for performance and user feedback.
   * Schedule periodic updates and patches to maintain system integrity.

## Key Considerations for Implementation

* **User Engagement**: Foster excitement and participation through promotional efforts, such as email campaigns or in-person announcements.
* **Change Management**: Ensure users are prepared for the transition by addressing their concerns and highlighting system benefits.
* **Feedback Mechanism**: Establish a feedback loop to continuously collect user insights for future improvements.
* **Scalability**: Design the system with flexibility to accommodate future growth, such as expanding to other departments or adding new features.

# CONCLUSION

The Pharmacy Management and Inventory System is designed to bridge critical gaps in inventory management, sales tracking, and efficient pharmacy operations. By streamlining processes and providing real-time insights, the system creates a dynamic environment where pharmacy staff and administrators can manage resources and serve customers seamlessly. It fosters a secure, organized, and efficient workflow that empowers pharmacies to deliver exceptional service while maintaining operational excellence.

Through features like real-time stock monitoring, automated notifications, sales processing, and detailed reporting, the system transforms the way pharmacies operate. Administrators gain powerful tools to manage inventory and analyze performance, while staff benefit from user-friendly interfaces that simplify their daily tasks.

Ultimately, this system aspires to not only enhance operational efficiency but also elevate the quality of service provided to customers. By leveraging technology to optimize pharmacy operations, the platform paves the way for a new standard of excellence in pharmacy management, ensuring customer satisfaction and long-term success for businesses.

.

## References

* Baran, P., Baum, R., Bieńkowski, J., Golec, A., Korzeniewski, J., Kuźmar, S., et al. (2021). Pharmacy Automation and Inventory Management: A Solution for Efficient Pharmacy Operations. Creative Commons Attribution-Non-Commercial 3.0 IGO.
* Rahman, M. M., & Rahman, M. H. (2017). An Overview of Pharmacy Management Systems: The Role of Automation in Healthcare. Journal of Health Informatics, 13(2), 45-58. https://doi.org/10.1016/j.jhin.2017.02.005
* Kumar, R., & Rathi, P. (2019). Pharmacy Inventory Management: Challenges and Solutions. International Journal of Advanced Research in Computer Science and Engineering, 6(8), 110-120.
* Ahmed, F., & Hasan, S. (2020). Building Efficient Pharmacy Management Systems with Automation: Case Studies and Applications. Springer Nature. https://doi.org/10.1007/978-3-030-11885-1
* World Health Organization (WHO). (2020). Guidelines for Pharmacy Practice in Healthcare Systems. World Health Organization. <https://www.who.int/pharmacy>
* Ali, S., & Rahman, A. (2018). E-Pharmacy: An Online Pharmacy System for Management and Prescription Fulfillment. Journal of Pharmaceutical Technology, 32(3), 89-94.