# Requirements

## Functional requirements

**1** Sales:

**1.1** Selling Medicine:

**1.1.1** The system should allow employees to record sales transactions, including details such as customer   
 information, medicine name, quantity sold, and unit price.

**1.1.2** Integration with a barcode scanner or manual entry should be supported for efficient sales processes.

**1.2** Inventory Management:

**1.2.1** The system should automatically update the available quantity of each medicine after a sale is  
 completed,ensuring real-time inventory tracking.

**1.2.2** Notifications should be generated when the stock of a particular medicine falls below a predefined threshold.

**1.3** Sales Performance Tracking:

**1.3.1** The system should provide tools for tracking and analyzing sales performance, including metrics such as  
 total sales, top-selling medicines, and sales trends over time.

**2** Purchase:

**2.1** Buying Medicine from Distributors:

**2.1.1** Employees should be able to create purchase orders for medicines from authorized distributors, specifying details like quantity, and expiration dates.

**2.2** Inventory increase:

**2.2.1** The system should automatically update the available quantity of each purchased medicine, reflecting the increase in stock.

**3** Reports:

**3.1** Sales Reports: **3.1.1** The system should generate detailed reports on sales activities, including daily, weekly,  
 and monthly summaries, as well as custom date ranges. **3.1.2** Reports should include information on total sales, and top-selling medicines.

**3.2** Purchase Reports: **3.2.1** Detailed reports on purchases should be available, displaying information such as total expenditure,   
 and quantities purchased.

**3.3** Medicine Reports:  
 **3.3.1** Comprehensive reports on medicine inventory, including stock levels, expiration dates,   
 and reorder suggestions, should be generated.

**3.4** Net Profit Reports:

**3.4.1** The system should calculate and provide reports on the net profit, taking into account both sales and   
 purchase transactions.

**4** Alerts:

**4.1** Expiration Alerts:

**4.1.1** The system should generate alerts for employees when the expiration date of a medicine is approaching,  
 allowing timely actions such as marking for discounts or removal from shelves.

**5** Employees:

**5.1** User Authentication:  
 **5.1.1** The system should have a secure login mechanism with role-based access control, ensuring that employees   
 can only access functions relevant to their roles.

**5.2** Activity History:  
 **5.2.1** A detailed activity log should be maintained for each employee, capturing actions such as logins, logouts,   
 sales transactions, purchases, and any system operations.

**6** Return:

**6.1** Return to Distributors: **6.1.1** Employees should be able to process returns of medicines to distributors within the specified time frame, recording relevant details such as reasons for return and quantities returned.

**6.2** Inventory Update: **6.2.1** The system should automatically adjust the available quantity of returned medicines in the inventory.

**6.3** Stock Reduction: **6.3.1** The return process should result in a decrease in the number of available medicines, reflecting accurate stock levels.

Non-functional requirements:

1. \*Performance:\*

1.1 The system should respond to user interactions within two seconds under normal operating conditions.

1.2 The time taken to generate reports, especially large datasets, should not exceed five seconds.

1.3 Concurrent user support: The system should be able to handle at least 100 concurrent users without a significant degradation in performance.

2. \*Reliability:\*

2.1 The system should have a mean time between failures (MTBF) of at least 500 hours.

2.2 In the event of a failure, the system should recover within five minutes with minimal data loss.

2.3 Regular system health checks should be conducted to identify potential issues before they affect operations.

3. \*Security:\*

3.1 All sensitive customer and employee data should be encrypted during transmission and storage.

3.2 Role-based access control (RBAC) should be enforced, ensuring that employees can only access functionalities based on their assigned roles.

3.3 Regular security audits should be conducted to identify and address potential vulnerabilities.

3.4 Passwords should be stored securely using industry-standard hashing algorithms.

4. \*Scalability:\*

4.1 The system should accommodate a 20% annual growth in data and user load.

4.2 It should support the addition of new features and functionalities without requiring a complete system overhaul.

4.3 Scalability tests should be conducted to ensure optimal performance under increased loads.

5. \*Usability:\*

5.1 The user interface should be intuitive and require minimal training for new employees.

5.2 The system should provide clear error messages and guidance to users in case of input errors.

5.3 Accessibility features should be implemented to ensure usability for users with disabilities.

6. \*Availability:\*

6.1 The system should be available 99.9% of the time during regular business hours.

6.2 Scheduled maintenance should be communicated in advance, and efforts should be made to minimize downtime.

6.3 Redundancy measures should be in place to ensure continuous operations in case of server or hardware failures.

7. \*Backup and Recovery:\*

7.1 Regular automated backups of the database and system configuration should be performed.

7.2 The backup data should be stored in a geographically separate location to ensure recovery in case of a catastrophic event.

7.3 A documented and tested recovery plan should be in place to restore the system to its operational state in the event of data loss or system failure.

8. \*Dependability:\*

8.1 The system should be designed with built-in fault tolerance to minimize the impact of hardware or software failures.

8.2 Dependencies on external services or APIs should be identified and managed to prevent disruptions in case of external service outages.

8.3 Continuous monitoring of system dependencies and components should be in place to identify potential issues proactively.