

SRS Documentation

Case study 1 - Time Management Software

Introduction:

Purpose: This document outlines the requirements for the Time Management Software (TMS) for company executives. TMS will help executives manage their daily appointment schedules, meetings, and leave periods, and provide statistics regarding their time management.

Project Scope: TMS helps executives manage appointments, meetings, and tasks, while secretaries schedule group meetings and send email notifications. It provides daily schedule reminders, tracks leave, and generates reports on time spent in meetings and projects.

Environmental Characteristics: TMS will run on the executives' and secretaries' desktop computers, handling multiple simultaneous remote requests. It will communicate with email servers to send daily schedules and meeting notifications. The system must be easy to use for executives who may have limited computer skills.

Overall Description:

Product Perspective: TMS replaces manual scheduling by automating appointment and meeting management for executives. It integrates with desktop environments and email systems for efficient communication and time management.

Product Features:

- Register appointments
- Schedule multi-executive meetings
- Reschedule conflicts
- Email notifications

Manage leave periods
Track tasks
Statistics and reports

User Classes:

Executives: Register their schedules, mark leave periods, log daily tasks, and view statistics.

Secretaries: Manage executive schedules, organise multi-executive meetings, and handle conflicts.

Managers: Review statistics and reports about time spent on meetings.

Operating Environment: TMS will run on desktop computers used by executives and secretaries, with support for multiple simultaneous users. It will interact with email servers to send daily schedules and meeting notifications.

Design Constraints:

Remote access: The software must be capable of handling remote requests simultaneously.

Ease of use: Since many executives are not experienced with computers, the system must have a simple and intuitive user interface.

Email integration: TMS should integrate with the company's email system to send notifications.

User Documentation:

User manuals and training materials will be provided for executives, secretaries, and managers, explaining how to use the system for scheduling, meeting management, and reporting.

Functional Requirements:

Executives:

Register appointments

Log tasks
Mark leave period
View statistics

Secretaries:

Organize meetings
Reschedule conflicts.
Check schedules

Managers:

View reports

External Interface Requirements:

User Interfaces: A simple, intuitive interface for scheduling meetings, registering tasks, and viewing reports. Integration with email servers to send meeting notifications and daily schedules.

Hardware Interfaces: TMS will run on standard desktop computers.

Software Interfaces: Integration to send email notifications.
A database to store executives' schedules, meetings, tasks, and statistics.

Communications Interfaces: Executives will be notified of new meetings, changes, and daily schedules via email.

Other Non-Functional Requirements:

Performance Requirements: The system must support multiple simultaneous remote requests from different users without performance degradation.
Emails containing daily schedules should be sent by 8:00 AM every morning.
The system should generate and display statistics within 10 seconds for any time period.

Safety Requirements: Regular data backups to prevent data loss.
Recovery mechanisms in case of server or system failure.

Security Requirements:

User authentication: Executives, secretaries, and managers must log in using secure credentials.

Access control: Secretaries should not access executives' personal task lists, and vice versa.

Case study 2 - Hotel Automation Software

Introduction:

Purpose: The software would be deployed on the administrator systems of the reception of the hotel and a protected application that can be downloaded and accessed by the appropriate personnel in the IP address of the hotel. The software will be used for assigning rooms to guests, providing frequent guests with special identity numbers for discounts, inputting tariffs based on displayed occupancy rates, calculating food bills of the customer and tallying all payments for final bill generation.

Project Scope: The software is being developed to automate most functions of the receptionist in hotels by keeping clean records of all necessary information with minimal human interference. Assigning rooms to guests based on availability on the spot, storing occupancy rates and bill generation tasks will be automated based on the appropriate details entered by the required personnel. The receptionist must finalise the bookings, the manager must enter the desired tariff change percentage and the hotel catering services manager must enter the quantity of food items consumed by the customer. In further versions, automation of aforementioned manual labour can be explored as per the discretion of the users after receiving their feedback.

Environmental characteristics: The hardware requirements are limited since the software only needs to run an application with network access contained to its own network. Windows 10 operating system is sufficient, there are no major software requirements that cannot be handled by Docker.

Overall Description:

Product Perspective: The product is intended to enhance current hotel management software, which is usually only catered towards the receptionists' work. The manager's job will be made easier with the occupancy rate displayed. The hotel catering services manager would normally have to calculate the customers' bills separately and integration would be taken care of later, which is open to possible faults. Therefore, by building a centralised application, these errors can be minimised.

Product Features: The software will be used for assigning rooms to guests, providing frequent guests with special identity numbers for discounts, inputting tariffs based on displayed occupancy rates, calculating food bills of the customer and tallying all payments for final bill generation.

User Classes: The three major user classes are

1. *Receptionist* - Low skill level needed to enter data pertaining to customers.
2. *Manager* - Low skill level needed to view current occupancy information and tariffs and modify the values as per calculations
3. *Hotel Catering Services Manager* - Low skill level needed to enter details of consumption of food by customer

User Documentation: Demonstration video will be prepared for each user class, further documentation of features will also be provided.

Functional Requirements:

1. Revising Tariff
 - a. Display average occupancy rate for a month
Input: Month
Output: Average occupancy rate for the given month.
 - b. Revise tariff
Input: Percentage by which to revise tariff, whether tariff is to be increased or decreased, room type for which tariff is to be revised
Output: Revised tariff
2. Reserving room
Input: Guest arrival time, advanced paid, approximate duration of stay, type of room required
Output: Room number and unique token number if the guest can be accommodated. If not available, an apology message is generated.
3. Frequent guest identification
Input: Details of guest
Output: Identification number if applicable
Processing: Calculates how frequently the guest visits and if it is above a threshold value then an identification number is generated.
4. Bill Generation
 - a. Adding catering details

Input: Quantity, type of food, token number of guest, corresponding date and time

Output: Price of food

b. Bill generation

Input: Customer token number, identity number

Output: Entire bill for customer and amount payable by them

Processing: The bill is generated based on tariff of room, discount for frequent customers based on identity number, cost of food based on details given by catering staff.

Non-Functional Requirements:

Safety Requirements: Standard recovery options for database holding information regarding customers' stay and consumption of food. Each transaction will be confirmed on the user's screen only after the database is updated. Every hour, the database will be backed up.

Security Requirements: Each user class will be allowed to use their own class' respective functionalities. The manager can view and edit the receptionist's entries and can view the hotel catering services managers' entries. Only the manager can view their own activities.

User Interface: User-friendly interface for three key user roles: the receptionist, the catering services manager, and the hotel manager. The receptionist interface will allow for room reservations, guest check-ins, and real-time room availability checks, while capturing guest details such as arrival time, advance payments, and stay duration. The catering manager interface will facilitate the logging of consumed food items against each guest's token number. The manager interface will provide access to reports, including the average occupancy rate for a given month, and enable dynamic room tariff adjustments based on demand trends. The system will ensure seamless integration across all modules, following principles of data consistency, security, and scalability to handle various levels of user input, while also offering the personalised service of frequent guest discounts.

Communication Requirements: The protocols to be supported are HTTPS for communication between client interfaces and server, REST for web-based interactions between the software's components, SFTP for file transfer and SMTP for ability to automate mailing guests.

Case study 3- Road Repair and Tracking Software

Introduction:

Purpose: This document outlines the requirements for the development of the Road Repair and Tracking Software (RRTS) for automating the bookkeeping activities associated with road repair tasks for the Public Works Department of the City Corporation.

Project Scope: Automate repair request logging, scheduling, and tracking based on priority , severity, locality, and resource availability.

Environmental Characteristics: RRTS will interact with the existing city network infrastructure, integrating with manpower(the clerks involved) and machinery databases.

Overall Description:

Product Perspective: New software replacing manual systems for repair requests and scheduling, working within the city's administrative network.

Product Features: The RRTS will log repair requests, prioritise and schedule repairs, track resource usage, and generate reports for management. This ensures efficient task handling, resource management, and real-time insights for informed decision-making.

User Classes: Clerks log repair requests, supervisors set priorities and schedule repairs, administrators manage resources, and the mayor views reports.

Operating Environment: Runs on standard city office computers with Windows/Linux OS.

Design Constraints: Must comply with city data security and performance standards, integrate with existing databases and networks.

User Documentation: User manuals and troubleshooting guides.

Functional Requirements:

Clerks: Clerks are responsible for logging new road repair requests into the system and viewing the status of existing repair requests.

Supervisors: Supervisors set the priority for repairs based on road condition and locality type. They schedule repair work according to resource availability and update the status of the work after its completion.

Administrators: Administrators manage the availability of manpower and machinery and modify scheduling in case of any changes to resources.

Mayor: The mayor can generate reports on repairs that are completed or outstanding and view statistics on resource utilisation over any specified time period.

External Interface Requirements:

User Interfaces: Simple UI with screens for repair logging, scheduling, and report generation. Consistent layout and error messages.

Hardware Interfaces: Standard office computers and network devices.

Software Interfaces: Interfaces with city's manpower and machinery databases, and existing administrative systems.

Communications Interfaces: TCP/IP for network communication, with secure data transmission standards.

Other Non-Functional Requirements:

Performance Requirements: The system must handle up to 50 repair requests per minute during peak hours. Scheduling of repairs should be completed within 2 seconds, and report generation must be under 5 seconds.

Safety Requirements:RTS should ensure data integrity during power outages or system crashes. Regular backups should be implemented to avoid data loss.

Security Requirements:RTS must implement user authentication for clerks, supervisors, and administrators. Data encryption is required for all communications, and compliance with city data privacy regulations is mandatory.

Case study 5- Work Processing Software

Introduction:

Purpose: This document outlines the requirements for the development of the Work Processing Software (WPS) to automate the reading, formatting, and editing of text from ASCII or HTML files, enabling users to perform various word-processing tasks.

Project Scope: WPS will automate word processing by reading text files, applying formatting based on user-specified parameters, and offering features like justification, editing, and text styling. Future enhancements may include advanced text styling and collaborative editing.

Environmental Characteristics: WPS will run as standalone software on standard desktop and laptop computers and be compatible with widely used operating systems like Windows and Linux.

Overall Description:

Product Perspective: WPS is a new tool for processing and formatting text, offering features similar to basic word processors, but with custom formatting options like user-defined line lengths and justification.

Product Features:

Read text from ASCII and HTML files.

Format text with left and right justification.

Allow word editing with options to add, modify, or delete words.

Support for text styling such as bold, italic, superscript, and subscript.

Save formatted text as HTML files.

Display document statistics (characters, words, lines, paragraphs).

User Classes:

General Users: Responsible for reading, editing, formatting, and saving documents.

Operating Environment: WPS will operate on standard desktop or laptop computers with Windows or Linux operating systems.

Design Constraints:The software must adhere to text formatting standards, allow user customization of output line length, and ensure compatibility with HTML file structures.

User Documentation:The software will include a user manual and troubleshooting guide to assist with text formatting, file management, and basic editing tasks.

Functional Requirements:

General Users:

Input text files in ASCII or HTML format.

Define the number of characters per line (between 1 and 132).

Format text with both left and right justification.

Edit text by adding, modifying, or deleting words.

Apply bold, italic, superscript, or subscript styling to words.

Save formatted text as an HTML file.

View statistics, including the number of characters, words, lines, and paragraphs in the document.

External Interface Requirements:

User Interfaces:A simple UI allowing users to upload files, format text, and edit or style words. Consistent error messages and clear navigation.

Hardware Interfaces:Compatible with standard desktop or laptop computers.

Software Interfaces:Integrates with file systems for saving and loading files in ASCII or HTML format.

Communications Interfaces:No special communication interfaces are required as this is standalone software.

Other Non-Functional Requirements:

Performance Requirements:The system should process and format a document with up to 10,000 words within 3 seconds. Editing actions should be near-instantaneous, and document statistics must be available within 1 second.

Safety Requirements:WPS should regularly save backups of edited documents to prevent data loss in case of system crashes

Security Requirements:Basic file access controls should be implemented, allowing users to save and retrieve files securely.

Case study 4 - Judiciary Information System

Introduction:

Purpose: The software is to be deployed on servers in the attorney general's office. The server can only be accessed by systems in the court or the attorney general's office in order to minimise the probability that an unauthorised person gains access. Access is also password protected and only lawyers and judges have valid login credentials.

This software is used by lawyers and judges to browse old court cases. It is also used by the registrar to schedule hearings, update details for ongoing cases, and view details of both past closed cases and upcoming cases. The registrar can also view details of all cases to make their job of scheduling cases and maintaining progress easier.

Project Scope: This software is being developed for the attorney general's office. It makes it easier for the registrar to schedule hearings by automatically finding empty slots during a given time frame. However, the registrar has to manually open cases, schedule hearings and update case details.

Judges and lawyers can both access old cases. This process is automated allowing them to search for keywords and access relevant cases in order to develop lines of arguments or judgements. The lawyer is automatically charged for accessing old cases.

Environmental Characteristics: Requires a server with enough memory to support a large dataset containing past court cases. The server must also be able to handle multiple requests at the same time given the expected volume of people accessing it. However the operations themselves are fairly simple so it does not need a lot of computing power to handle each individual request.

Overall Description:

Product Perspective: This product is a new software. It is not a part of a larger system and focuses primarily on automating tasks that previously had to be accomplished manually.

Product Features:

The software can be used by the court registrar to view vacant slots during the week and schedule hearings during those slots. They can also update case details and schedule new hearings until the case is complete. Once the case is complete they can close the case.

The registrar can also access the database to view upcoming cases on a particular date, pending cases, cases resolved during a particular time and the status of any specific case.

The lawyers and judges have to login to access the system. Lawyers are automatically charged for every court case they access. Both lawyers and judges can browse through old cases by searching keywords. They can also delete their accounts.

User Classes:

1. **Lawyers:** Can browse old cases using keywords. They have login credentials and are charged every time they access a case. Their accounts can also be deleted.
2. **Judges:** Can browse old cases using keywords. They have login credentials and can delete their account whenever they choose. The user interface they interact with has to be simple and rugged because most judges have less experience with technology.
3. **Court Registrar:** The court registrar has basic proficiency in computer systems so the user interface can include some shortcuts to make their work easier. They use the system to schedule hearings, update case details and view details of past and upcoming cases.

Operating Environment: The server used is an encrypted server in the attorney general's office. The only devices which can access this server are those in the office or the court. This is ensured by only allowing devices connected to the court network or the attorney general office network access to the server. These devices accessing the server should have an operating system of Windows 8 or above.

Design and Implementation Constraints: The software must be able to support multiple users accessing the database, especially during peak hours. It must also integrate with pre-existing softwares in the attorney general's office.

User Documentation: Users will be provided with videos tailored for each user class. User manuals and troubleshooting guides will also be provided.

Functional Requirements:

1. Case management
 - a. Case creation
Input: Name of defendant, defendant's address, type of crime, date committed, location committed, name of arresting officer, date of arrest
Output: Unique case identification number
 - b. Assigning Date of Hearing
 1. Slot availability
Input: Press slot availability button
Output: Show vacant slots on any working day when hearing can be scheduled
 2. Slot selection
Input: Selection of slot
Output: Confirms slot selection and updates database accordingly
 - c. Case adjournment
Input: Reason for adjournment by court registrar
Output: Confirmation and updation of database, new hearing date from 2
 - d. Hearing Completion
Input: Summary of court proceedings
Output: Confirmation and updation of database, new hearing date from 2
 - e. Completion of court case
Input: Summary of judgement is recorded, name of preceding judge, public prosecutor, starting date, expected completion date of the trial
Output: Confirmation that case is closed
2. Browsing old cases
 - a. Logging in
Input: Category (Judge, Lawyer or Registrar), Username, Password

Output: If password is correct, access provided to cases else revert to login page

b. Search cases

Input: Keywords regarding the case

Output: Past court cases containing those keywords. If lawyer is accessing, a charge should be displayed to them

3. Account deletion

Input: Pressing account deletion button after login

Output: Confirmation of account deletion and according updation of database

4. Registrar querying currently pending court cases

Input: Currently pending cases button

Output: Currently pending cases sorted by CIN with each displaying date in which case started, defendant's name, address, crime details, lawyer's name, public prosecutor's name, judge's name

Processing: Sort the CIN

5. Registrar querying cases resolved over any given period

Input: Beginning and ending date

Output: Display starting date of the case, the CI, the date on which the judgement was delivered, name of the attending judge, judgement summary

Processing: Arrange in chronological order of date

6. Registrar querying cases coming up for hearing on a particular day

Input: Desired day

Output: Display the cases coming up on the inputted day with their respective details

7. Registrar querying the status of any particular case

Input: CIN

Output: Status of the case identified by the inputted CIN

External Interface Requirements:

User Interfaces: The UI lawyers and judges interact with is simple with a search bar for keywords and an area below it displaying result. Back buttons are available on the screen to refer to a previously searched query. The court registrar has more buttons on their screen and more shortcuts as well to make their work easier.

Hardware Interfaces: Standard server and computer hardware can be used. Network devices required are standard speed but a high level of security.

Software Interfaces: The software interfaces with the dataset containing all cases and their dates. This database is updated in real time.

Communications Interfaces: TCP/IP connection are used to secure communication between clients and the server. The server network can only access requests made from systems in the attorney general's office or the court in order to prevent unauthorised personnel from accessing the server.

Other Non-Functional Requirements:

Performance Requirements: Given the fact that multiple users may be accessing the server, the software must be able to complete up to 50 transactions per second. The response time for a request must be under 3 seconds and modifications made should update in real time.

Safety Requirements: The database should be recoverable even in case of power outage. There should be a secondary server used as a backup in a separate secondary location (like the court) in case the primary server is destroyed due to fire, flood, insect or any other physical issue. Data integrity should be maintained through power outages and system crashes.

Security Requirements: Users have to be verified using their account information before they get access to the operating system. The software must also only be functional when connected to either the court's network or the attorney general office's network in order to prevent unauthorised access. Data encryption for all communications over these networks is mandatory.

Case study 6 - Restaurant Automation System

Introduction:

Purpose:

The Restaurant Automation System (RAS) will be deployed within the restaurant's operational environment to automate order processing, billing, and accounting activities. The software will be used by restaurant staff to enter item codes and quantities sold, generate customer bills, track ingredient usage, and manage inventory levels. It will also facilitate accurate supply ordering by calculating inventory thresholds, generating purchase orders, and processing invoices for received ingredients. Additionally, RAS will support price updates and generate monthly sales and expense reports on request.

Project Scope:

The scope of the software includes automating core functions such as order processing, billing, and inventory management to enhance operational efficiency. The software will handle tasks like tracking sales, managing ingredient stock, generating purchase orders based on inventory levels, and processing invoices.

Future developments may include advanced analytics for sales trends and integration predictive models to determine future ingredient usage.

Environmental Characteristics:

The software will operate on standard desktop or tablet computers within the restaurant, using common operating systems such as Windows or Android. It will interact with hardware components including barcode scanners for item entry and receipt printers for bill generation.

Overall Description:

Product Perspective:

The Restaurant Automation System (RAS) is a new software designed to enhance and automate various operational processes within a restaurant. It is intended to replace manual methods of order processing, billing, and inventory management, improving accuracy and efficiency. RAS will function independently. It accesses financial and inventory databases and makes modifications.

Product Features:

The Restaurant Automation System will automate key functions including order processing, billing, and inventory management. It will handle sales transactions by

generating bills, track ingredient usage, and calculate inventory thresholds to manage stock levels effectively. The software will also generate daily purchase orders when stock levels fall below predefined limits, process invoices for received ingredients, and facilitate immediate cheque printing if cash balance allows. Additionally, RAS will provide monthly reports on sales and expenses to support financial oversight.

User Classes:

Restaurant Manager: Oversees overall system operation, manages item prices, reviews monthly sales and expense reports, and authorises cheque printing for invoices. Expected to have a moderate level of computer expertise to navigate the software's features and generate reports.

Sales Clerks: Use the software to enter item codes and quantities during transactions, and generate customer bills. Basic computer skills are required to operate the system efficiently.

Kitchen Staff: Enter data regarding ingredient usage into the system. They need to have basic familiarity with the software to update ingredient usage accurately.

Accounts Personnel: Handle invoice data, process payments, and ensure sufficient cash balance for cheque printing. They will interact with financial components of the software and need a good understanding of accounting practices and software functionality.

Operating Environment:

It will run on standard desktop or tablet computers within the restaurant, compatible with operating systems such as Windows 10 and above, or Android. The software will interact with hardware components like barcode scanners for entering item codes and receipt printers for generating customer bills. It will work with the inventory and finance databases present for the restaurant.

Design and Implementation Constraints:

The software must comply with data protection regulations for financial and inventory information. It will be designed to run on standard restaurant hardware. It does not have to integrate with pre existing applications but has to integrate with the restaurant inventory and financial database which may be stored on servers with poor response time.

User Documentation:

Includes comprehensive user manuals that cover system features and operational procedures. Additionally, troubleshooting manuals will be provided to help users identify and resolve common issues and errors, along with training videos for employees to learn to use the software.

Functional Requirements:

1. Changing price of item

Input: Item code and new price

Output: Change made in database and confirmation message regarding the same

2. Sale of food item

Input: Item code and quantity sold

Output: Computer generated bill using the price stored in the database

3. Ingredient handling

- a. Ingredient ordering

Input: Ingredient code issued for preparation of food item and quantity of item being used

Output: If the ingredient quantity falls below the threshold value then a purchase order for the ingredient is generated

- b. Ingredient payment

Input: Invoice data regarding price and quantity of ingredient

Output: Store the data in the database. If insufficient cash balance is available print a cheque against the invoice

4. Statistical analysis

- a. Display monthly sales receipt:

Input: Press the button labelled monthly sales receipt

Output: Display the sales from the last month

- b. Display monthly expenses data

Input: Press the button labelled monthly expenses data

Output: Display expenses from the last month.

External Interface Requirements:

User Interfaces:

This software will feature a user-friendly graphical interface designed for ease of use. The interface will include clearly organised screens for order processing, billing, and inventory management. Each screen will have standard push buttons for essential functions like "Save," "Cancel," and "Help," positioned for easy access. Keyboard shortcuts will be provided for common actions, such as "Ctrl + S" for saving and "Ctrl + P" for printing. Error messages will be displayed in standardised dialog boxes with clear instructions to guide users in resolving issues. Detailed user interface design, including sample screens and specific design guidelines, will be documented separately.

Hardware Interfaces:

The RAS will interface with several hardware components. It will support standard barcode scanners connected via USB or wired connections for item code entry. Receipt printers will be used for generating customer bills, connected through wired network interfaces.

Software Interfaces:

The Restaurant Automation System (RAS) will interface with several software components to ensure its functionality. It will connect with a local database to store and retrieve data related to orders, inventory, and financial transactions. The database will handle inputs such as item codes, quantities sold, ingredient usage, and invoice details, and output data such as sales reports, inventory levels, and purchase orders. The software will run on operating systems like Windows or Android. Integrated commercial components, such as barcode scanning libraries or receipt printer drivers, will be used to support hardware interactions.

Communications Interfaces:

The software will require communication interfaces to support its operations. It will use HTTP or HTTPS protocols for secure web access, ensuring encrypted data transmission.

Other Non-Functional Requirements:

Performance Requirements:

The software must be capable of processing up to 15 transactions per second to handle peak sales periods effectively. Inventory updates should occur in real time, with changes reflected within 3 seconds of data entry. The system must generate sales and purchase reports within 10 seconds.

Safety Requirements:

The software will include automatic data backup and recovery features to protect against power failures and minimise data loss. It will have error-handling mechanisms and alerts for software and hardware issues, with logs to aid in troubleshooting.

Security Requirements:

The software accesses the financial database of the restaurant thus must be highly secure. The channels of communication to the database are encrypted as are the cheque printing mechanisms. The other components are secured but the inventory logs have less layers of encryption.

Case study 10-Supermarket Automation Software

Introduction:

Purpose: This document outlines the requirements for the Supermarket Automation Software (SAS) to automate sales transactions, inventory management, and sales statistics reporting for the supermarket.

Project Scope: SAS will automate item scanning, billing, inventory updates, and sales statistics reporting. Future enhancements may include more advanced inventory management and customer loyalty features.

Environmental Characteristics: SAS will interact with barcode readers, weighing scales, and the supermarket's inventory database to register sales and manage stock.

Overall Description:

Product Perspective: SAS is a new system replacing manual billing and inventory tracking, fully integrated with supermarket hardware like barcode readers and scales.

Product Features:

Generate itemised bills for customers

Update inventory in real-time after each sale

Allow manual inventory updates when new stock arrives

Generate sales statistics for any specified period

User Classes:

Sales Clerks: Process transactions

Employees: Update inventory

Manager: View inventory, update prices, and review sales reports

Operating Environment: Runs on supermarket POS systems connected to barcode readers, scales, and printers, using Windows/Linux OS.

Design Constraints: Must integrate with existing supermarket hardware, support real-time updates, and ensure fast processing during peak hours.

User Documentation: User manuals and training materials for sales clerks, employees, and managers.

Functional Requirements:

Sales Clerks: Sales clerks will scan items, register quantities, and generate bills with total payable amounts for customers.

Employees: Employees will update inventory records when new stock arrives.

Manager: The manager can view inventory details, modify item prices, and generate sales statistics for specified time periods.

External Interface Requirements:

User Interfaces: Simple UI for sales, inventory management, and reporting. Consistent design for ease of use, with error handling and print options.

Hardware Interfaces: Integrates with barcode readers, weighing scales, and receipt printers.

Software Interfaces: Interfaces with inventory and sales databases to record transactions and update stock in real time.

Communications Interfaces: Interfaces with inventory and sales databases to record transactions and update stock in real time.

Other Non-Functional Requirements:

Performance Requirements: The system must handle up to 100 transactions per minute, update inventory in real time, and generate sales reports in under 5 seconds.

Safety Requirements: Data backup should occur regularly, and the system should recover from power or hardware failures without data loss.

Security Requirements: User authentication for sales clerks, employees, and managers. Data encryption for transaction and inventory records.

Case study 11-Book Shop Automation Software

Introduction:

Purpose: This document outlines the requirements for the Book-shop Automation Software (BAS), designed to automate various activities in a small book shop, including sales transactions, inventory management, book procurement requests, and sales statistics reporting.

Project Scope: BAS will allow customers to query book availability, assist in stock management, track sales transactions, generate sales statistics, and facilitate reordering of books. Future enhancements may include more advanced analytics and customer management features.

Environmental Characteristics: BAS will interact with the book shop's inventory system, providing real-time updates to stock levels, sales transactions, and book procurement requests. It will be used by sales clerks, employees, and the manager on existing computer systems running Windows/Linux OS.

Overall Description:

Product Perspective:: BAS is a new system replacing manual processes for querying book availability, updating stock, generating receipts, and maintaining sales statistics. It is designed to integrate smoothly with the existing operations of a small book shop.

Product Features:

Query book availability by title or author.

Record requests for books not in stock.

Update stock and generate sales receipts for purchases.

Maintain book pricing and sales data.

Generate sales statistics (book name, publisher, ISBN number, number of copies sold, sales revenue) for any period.

Calculate inventory levels required based on recent sales and procurement lead times.

Generate reorder lists for books that fall below a specified threshold.

Include the publisher's details in reordering reports.

User Classes:

Customers: Query book availability.

Sales Clerks: Process transactions and update stock.

Employees: Record book procurement details and update inventory.

Manager: Monitor sales statistics, set inventory thresholds, and review book requests for procurement.

Operating Environment: BAS will run on standard desktop systems (Windows/Linux), interfacing with inventory databases and receipt printers. It will work in an online or offline environment and support database synchronization when reconnected.

Design Constraints: BAS must support real-time inventory updates, handle multiple simultaneous user queries, and generate reports quickly. It should ensure minimal downtime and provide data integrity even during unexpected power or system failures.

User Documentation: User manuals and training materials will be provided for sales clerks, employees, and the manager, covering tasks like book queries, inventory updates, and report generation.

Functional Requirements:

Book Query: Users (customers or sales clerks) can search for books by title or author. If a book is not found in stock, the user will be prompted to enter the book's details (title, author, publisher, and ISBN) for future procurement.

Book Requests: The system will track requests for books not in stock and increment a request counter for each book. Managers can view the request field to estimate demand.

Sales Transaction: Sales clerks will enter the ISBN of the selected book for purchase. The system will update the stock, generate a sales receipt, and store the transaction details (ISBN, title, price, and number of copies sold).

Sales Statistics: The system will generate sales statistics, including book name, publisher, ISBN, number of copies sold, and sales revenue for any specified period. The manager can view these statistics to assess business performance.

Inventory Management: BAS will maintain stock levels for each book. The system will calculate the required inventory level for a book based on recent sales over two

weeks and the number of days needed to procure the book. The system will print daily reports of books that have fallen below the threshold, along with the publisher's details for reordering.

External Interface Requirements:

User Interfaces: Simple, user-friendly interface for sales clerks and employees to process transactions and manage inventory.

Reports interface for the manager to view sales statistics and book requests.

Hardware Interfaces: Receipt printer for printing sales receipts.

Computer systems to interface with the inventory database.

Software Interfaces: Interface with the book shop's inventory and sales databases to record transactions and update stock in real time.

Communications Interfaces: The system must be able to operate offline with local databases and synchronize when reconnected to the central inventory system.

Other Non-Functional Requirements:

Performance Requirements: The system must handle up to 50 transactions per minute. Inventory and sales updates should be reflected in the system in real time.

Reports must be generated within 3 seconds.

Safety Requirements: The system should include a backup process that saves data every hour. It should recover from unexpected failures without losing data.

Security Requirements: Authentication required for sales clerks, employees, and managers. Sensitive data, such as book pricing and sales statistics, should be encrypted.

Case study 15 - University Department Information System

Introduction:

Purpose: The software would be deployed on the administrator systems of the University's Administrator block to be used by each department secretary to perform their duties more efficiently.

Project Scope: The software is being developed to automate the calculations and storage of details regarding students taking admission and already present in the University. These are calculating CGPA from grades, marking subjects as backlogs or

pass and printing the grade sheet for the student based on their roll number. The activities that will be stored and allowed to efficiently be entered are entering student details at time of admission, keeping track of inventory, storing details of University grants and storing information regarding ongoing research projects and faculty publications. For all the information stored, there will be an appropriate querying facility.

Environmental characteristics: The hardware requirements are limited since the software only needs to run an application with network access contained to its own network. Windows 10 operating system is sufficient, there are no major software requirements that cannot be handled by Docker.

Overall Description:

Product Perspective: The product is intended to aid the department secretaries to efficiently document the various activities relating to their department, its faculty and its students. It is meant to replace the exhausting book-keeping activities of the department secretaries that is tedious to enter, hard to search through and difficult to send information.

Product Features: The software will be used by the department secretary for student management for admission, course registration and grade entry, inventory management, financial management for departmental expenses, report generation, administrative functions of course tracking and inventory auditing and keeping track of academic information. For all the above features, there will be an interactive querying feature and a means of sending information directly from the software to the concerned personnel and authorized personnel if need be.

User Classes: The only user class is the *Department Secretary* of each Department. Low skill level is sufficient since major task is only entering information in the appropriate table using a simple form type user interface. To query, they can choose what details they wish to specify in a drop down menu and provide the information about that to view the corresponding rows of data following the constraints

User Documentation: Demonstration video will be prepared for each user class, further documentation of features will also be provided.

Functional Requirements:

1. Student Management
 - a. Student admission
Input: Student's name, address, course registered, roll number, date of admission.
Output: Confirmation of successful admission, creation of a student profile in the system.
 - b. Course registration
Input: Roll number, courses to be registered for the semester.
Output: Student's current registration status, tracking of completed courses and backlog, confirmation of course registration, updated student profile.
 - c. Grading information entry
Input: Roll number, grades submitted by instructors for registered courses.
Output: Computation of semester grade point average (GPA), cumulative grade point average (CGPA), and updated student profile. Grade sheet is generated and printed.
2. Inventory management/tracking
Input: Item name, item type (equipment, furniture, etc.), location, purchase date, purchase price.
Output: Confirmation of item entry into the system, updated inventory records, equipment details including location.
3. Financial management
 - a. Departmental expenses

Input: Item name, item type, purchase price, date of purchase, funding source (annual grant or consultancy fund).

Output: Confirmation of expense, updated financial records, summary of total expenditure from grants and consultancy services.

b. Annual grant management

Input: Annual grant amount, date of grant receipt.

Output: Confirmation of grant receipt, updated financial records showing available balance, summary of grant usage.

c. Consultancy fund management

Input: Consultancy service details, amount received, date of receipt.

Output: Confirmation of fund receipt, updated financial records showing available consultancy funds.

4. Reporting

a. Grade sheet generation

Input: Roll number, grades for current semester.

Output: Generation and printing of student's grade sheet, updated GPA and CGPA.

b. Financial reports

Input: Date range, category (annual grant, consultancy services).

Output: Detailed financial report showing expenditures and remaining balances for the selected category.

5. Administrative functions

a. Course completion tracking

Input: Roll number, courses completed, backlogs.

Output: Updated student academic history, flagging of incomplete or backlog courses.

b. Inventory Auditing

Input: Date range, item type (equipment, furniture, etc.).

Output: Report on inventory usage and location, confirmation of equipment status.

6. User access

a. Login for department secretary

Input: Username, password.

Output: Access granted to department management functions or reverted to the login page upon failure.

b. Login for instructors

Input: Username, password.

Output: Access to grading input functions or reverted to the login page upon failure.

7. Delete student profile

Input: Roll number, confirmation of deletion by department secretary.

Output: Confirmation of account deletion and updating of the database to reflect the student's removal.

8. Communication

Input: Selecting the row from any of the tables that are accessible and not private

Output: Opening Outlook and automatically attaching the appropriate information that can only be sent to the concerned personnel and authorized administrators if needed.

9. Academic information

a. On-going Research Tracking

Input: Affiliated Professor names, affiliated student names, branch, year, research topic, funds

Output: Stored in database as on-going research

b. Past publications

Input: Professor names, branch, year of publishing, research topic, publication name / conference name

Output: Stored in database as past publications

Non-Functional Requirements:

Safety Requirements: Standard recovery options for database holding information regarding all applicable tables. Each transaction will be confirmed on the user's screen only after the database is updated. Every 15 minutes, the database will be backed up. More importance will be given to preserving financial transactions.

Security Requirements: Only department secretaries and their superiors will be able to log in to the software. Incorrect attempts will be recorded and sent to the Outlook mail of the department secretary and their superior.

User Interface: On opening the software, the department secretary will be presented with a login page for their registered username and password. After logging in, they will be shown a welcome page and widgets will be present named after the various classes of activities they can perform, which are named after the headings of the functional requirements. After clicking on a widget, they will be permitted to choose from the functions they can perform under that option. For all entry related functions, an appropriate form like page will be provided, required fields will be marked and constraints will be mentioned. For all viewing functions, a facility to send a mail to the appropriate personnel will be present with a button. For all querying functions, a drop down menu to select features and then enter their values will be provided.

Communication Requirements: The protocols to be supported are HTTPS for communication between client interfaces and server, REST for web-based interactions

between the software's components and SMTP for ability to automate mailing desired personnel.

Case study 16-Medicine Shop Software

Introduction:

Purpose: The purpose of the Medicine Shop Software (MSS) is to automate and streamline inventory management and sales operations for a retail medicine shop. Deployed on the shop's computer system, MSS will facilitate real-time tracking of medicine inventory, automatically generating reorder alerts when stock levels fall below predefined thresholds and identifying expired items for vendor replacement. The software will also handle sales transactions by producing cash receipts, calculating daily revenues, and providing detailed monthly sales reports and graphs.

Project Scope:

The project scope for the product encompasses the automation of key inventory management and sales processes within a retail medicine shop. Initially, MSS will address the automation of inventory tracking, including generating reorder alerts based on sales thresholds and managing expired medicines. It will also streamline sales transactions by producing cash receipts and generating daily and monthly revenue reports.

Future evolution of the software may include the integration of advanced features such as real-time inventory updates across multiple locations, predictive analytics for stock management, and enhanced vendor management capabilities.

Environmental Characteristics:

The software will run on a standard desktop or laptop computer with common operating systems like Windows. It will interact with barcode scanners for entering medicine codes and printers for generating receipts and reports. It will also interact with the software used to store the database of medicines.

Overall Description:

Product Perspective:

The Medicine Shop Software (MSS) is a new application designed to improve inventory and sales processes, rather than replacing an existing system. It will operate independently but is built to integrate with barcode scanners, printers, and possibly future inventory or vendor management systems. The software will streamline current manual tasks and may later connect with larger systems as the shop's needs grow.

Product Features:

- Tracks inventory levels and triggers reorder alerts.
- Lists expired medicines and generates replacement reports.
- Records sales transactions and prints cash receipts.
- Calculates daily revenue and provides daily revenue reports.
- Generates monthly sales graphs and trends.

User Classes:

Shop Owner: Manages inventory, monitors stock levels, handles reorder requests, reviews daily revenue reports, oversees expired medicines and vendor replacements, and processes sales transactions.

Shop Staff: Enters sales transactions, updates stock levels, and generates cash receipts.

Operating Environment:

The software will run on standard desktop or laptop computers equipped with common operating systems such as Windows. It will interact with hardware components including barcode scanners for entering medicine codes and printers for generating receipts and reports. It should work with the database which stores the medicine inventory. To ensure it has the memory and computing power it will only work with operating systems Windows 10 and higher versions.

Design and Implementation Constraints:

The design and implementation of the software will adhere to several constraints. It must comply with relevant regulatory requirements for handling and managing pharmaceutical data, including data protection and privacy regulations. Hardware

limitations include ensuring compatibility with standard desktop or laptop configurations and minimal system resource requirements to maintain performance. The operating system it works on is Windows 10 or higher. The software can only be accessed from the shop, so communication protocols are not necessary.

User Documentation:

User manuals will be provided to the shop owner along with videos on how to use the system. Additional training videos will also be sent to aid in teaching the shop staff.

Functional Requirements:

1. Sale of medicines

Input: Code of medicine, corresponding quantity sold

Output: Add this value to the database and print cash receipt for the purchase

2. Ordering medicines

a. Purchasing medicines when quantity is less

Input: Sale of a medicine

Output: If the quantity of medicine is less than threshold value place an order to the vendor to purchase the medicine.

b. Listing the ordered items

Input: Press the button saying list items ordered

Output: Generate a list of all items ordered with the corresponding medical description, quantity and address of vendor supplying the medicine

c. Listing expired items

Input: Press the button saying list expired items

Output: Display a vendor wise list of all the medicines that have expired by comparing the expiry date in the database with the date of the next day.

3. Statistical analysis

a. Generate daily Revenue

Input: Button labelled daily revenue is clicked. An option is selected from a drop down menu which includes a list of all medical items and an option labelled total.

Output: Revenue for that day generated by the medicine selected from the drop down is displayed. If total is selected then total revenue generated for the day is displayed.

- b. Generate monthly graph

Input: Button labelled monthly is clicked. An option is selected from a drop down menu which includes a list of all medical items and an option labelled total.

Output: Graph displaying the sales of the item selected per day for the month is generated. If total is selected then a graph displaying total sales per day for the month.

External Interface Requirements:

User Interfaces:

The Medicine Shop Software (MSS) will feature an intuitive graphical user interface (GUI) designed for ease of use. The interface will be organised with clear navigation for key functions such as inventory management, sales processing, and report generation. Essential buttons like "Help," "Save," "Cancel," and "Submit" will be consistently placed on each screen. Keyboard shortcuts will be provided for common actions, such as "Ctrl + S" for saving and "Ctrl + P" for printing. Error messages will be clearly displayed in standard dialog boxes with helpful instructions.

Hardware Interfaces:

The product will interface with several hardware components to facilitate its operations. It will support standard desktop and laptop computers running Windows. The software will interact with barcode scanners to capture medicine codes during sales transactions, using USB or wired connections for data input. Additionally, MSS will interface with printers for generating cash receipts and reports, supporting common printer types via standard communication protocols such as wired network connections.

Software Interfaces:

The software is connected to a database via standard internet protocols. Data inputs to the database include changes in quantity every time it is stored. The output is data from the database presented based on the specification given by the shop owner.

Communications Interfaces:

Standard TCP/IP and HTTPS protocols are used to connect to the database via the internet. This ensures security of the communication.

Other Non-Functional Requirements:

Performance Requirements:

The Medicine Shop Software (MSS) is designed to meet several performance requirements to ensure efficient operation. It should handle up to 10 transactions per second, accommodating peak sales periods effectively. Inventory updates must be reflected in real-time, with changes visible within 2 seconds of data entry. Report generation, including daily revenue and monthly sales trends, should be completed within 5 and 15 seconds respectively. Additionally, the system must retrieve and display database information within 3 seconds to ensure quick access to inventory and sales data.

Safety Requirements:

The product will include automatic data backup and recovery features to protect against power failures and minimise data loss. It will have error-handling mechanisms and alerts for software and hardware issues, with logs to aid in troubleshooting.

Security Requirements:

The software will ensure robust security for data and communications. All data transmitted between the software and external servers will be encrypted using HTTPS to protect against interception and unauthorised access. The software will implement secure server configurations, including firewalls and intrusion detection systems, to safeguard against external attacks and unauthorised access. The software will also follow the data protection regulations for medical information.