**Practical-1**

**Aim: Develop the Software Requirements Specifications (SRS) document for a specific system.**

**Khatabook (POS)**

**1.1 Introduction:**

**1.1.1 Purpose of the system:**

Khatabook is a crucial tool for retail businesses, serving as the central hub for managing sales, inventory, payment processing, and customer interactions. Here are the key purposes and benefits of the system:

1. Streamlined Operations: It provides checkout efficiency which speeds up the checkout process by centralizing the database and provides real time tracking of inventory levels, reducing the risk of stock-outs and misplaced inventory.
2. Enhanced customer service: It helps to effectively personalize the promotion of products for each customer and supports various payment mediums.
3. Cost savings: By automating the sales tax and discount, it reduces risk of errors and staff time to focus on customer service.
4. Integration and flexibility: It is cloud based and is able to integrate with various tools.

It is better than existing systems in the following ways:

1. Security: All the transactions and transcripts are secured and are at end-to-end device;
2. Cross Platform: Any user can access the application on any device as the technologies used are platform independent.

**1.1.2 Scope of the system:**

Within the scope of the Khatabook, a cashier machine does not stand alone but already includes supporting software and other devices. The system does more than just buying and selling transactions, it can also integrate accounting calculations, goods and stock management, employee payroll modules, accounts payable accounts, and various other functions.

**1.2 General description of the system:**

**1.2.1 Overall description:**

The general functionality of the Khatabook system involves managing sales, inventory, payment processing, and customer interactions. It provides a streamlined interface for cashiers and managers, supports multiple payment methods, and offers detailed sales and inventory reports.

* **User Characteristics:** The users of the system include retail business owners, cashiers, managers, and possibly customers.
* **User Objectives:** Users aim to streamline business operations, enhance customer service, and gain business insights.
* **General Constraints:** The system should be user-friendly, secure, and compatible with various devices and operating systems.

**1.2.2 Feasibility study:**

1. **Technical Feasibility:**
   * The system can be developed using established programming languages such as Java and Python, and can utilize MySQL for database management and Apache server for hosting and likewise to deploy using web technologies.
   * The system can be scalable on demand.
   * As the resourcing is done on the cloud, users can access from anywhere and anytime.
   * The system is designed to be modular, allowing for easy upgrades and the addition of new features as needed.
   * The system will incorporate robust data validation, secure login protocols, encrypted data transmission, and reliable cloud infrastructure to ensure accuracy, reliability, ease of access, and data security.
2. **Operational Feasibility:**

Some of the important issues raised are to test the operational feasibility of Khatabook includes the following: –

* + Initial surveys and stakeholder meetings of the Khatabook will indicate strong support from both management and end-users for the proposed system.
  + Thorough testing and user training, will ensure that the system is user-friendly and functions as intended.
  + Minimal resistance is expected from the user as the system will significantly improve current processes with taking user’s requirement onto the picture during development. Any potential resistance will be addressed through comprehensive user training and support. So there is no question of resistance from the users that can undermine the possible application benefits.

1. **Economic Feasibility:**

Some of the important issues to test economic feasibility of the Khatabook: –

* The financial analysis indicates that the benefits, including increased efficiency, reduced errors, and enhanced customer service, will exceed the development costs. The system is a good investment for the organization.
* The system is economically feasible as it leverages existing hardware and requires minimal additional investment in new software tools, making it a cost-effective solution for the organization.

**1.3 Functional Requirements:**

**1.3.1 Module description:**

1. **Modules**
   1. **Sales Management**

The system will handle all sales-related activities from item scanning to payment processing and receipt generation.

* + 1. **Item Scanning**

Scanning the items using a barcode scanner to add them to the sales list is done here by the cashier.

* + 1. **Price Calculation**

Automatic price calculator calculates the total price including taxes and discounts is done here and at time of payment to cashier window.

* + 1. **Payment Processing**

It processes payments via cash, credit/debit cards, or digital wallets in cashier window.

* + 1. **Receipt Generation**

It generates and print receipts for customers.

* 1. **Inventory Management**

The system will manage the stock levels, updates inventory after each sale, and alerts for

low stock items.

* + 1. **Stock Entry**

It inputs new stock into the system.

* + 1. **Stock Adjustment**

It updates stock levels after sales or returns.

* + 1. **Inventory Tracking**

It tracks inventory levels in real-time.

* + 1. **Low Stock Alerts**

System will generate alerts when stock levels fall below the reorder threshold.

* 1. **Customer Management**
     1. **Customer Data Entry**

Customer will register themselves through the bar code provided and can shop and scan without the help of the cashier or user themselves.

* + 1. **Purchase History Tracking**

They can track and analyze customer purchase history and can learn on the purchasing patterns of the customer.

* 1. **Automated Sales Promotion**

Automated message alerts on offers and discounts on special events like birth-weeks, festivals etc., prior to one-two weeks of the event.

**1.4 Non- Functional Requirements:**

1.4.1 **Security:**

* **Password Protection**: The system should enforce strong password policies, including requirements for length, complexity, and periodic updates.
* **Data Encryption**: All sensitive data must be encrypted both in transit and at rest using industry-standard encryption protocols.
* **Access Control**: Implement role-based access control (RBAC) to ensure users can only access data and functions necessary for their roles.
* **Audit Logs**: Maintain comprehensive audit logs of all user activities and system events to detect and investigate security incidents.

1.4.2 **Reliability:**

* **Uptime**: The system must ensure 99.9% uptime, excluding scheduled maintenance periods.
* **Redundancy**: Implement redundant systems and failover mechanisms to ensure continuous operation in the event of hardware or software failures.
* **Error Handling**: Include robust error handling and logging to capture and address errors efficiently, minimizing impact on users.

1.4.3 **Availability:**

* + - **Quick Access**: Customer information and critical business functions should be quickly accessible from any authorized device.
    - **Disaster Recovery**: Establish a comprehensive disaster recovery plan, including regular backups and offsite storage to ensure data can be restored quickly in case of data loss.
    - **Load Balancing**: Use load balancing to distribute workloads evenly across servers, ensuring high availability even during peak usage times.

1.4.4 **Maintainability:**

* **Modular Design**: The application should be designed in a modular fashion to facilitate easy updates, enhancements, and bug fixes.
* **Documentation**: Maintain thorough documentation for system architecture, code, and user operations to assist in maintenance and onboarding of new developers.

1.4.5 **Portability:**

* **Cross-Platform Compatibility**: The application should be operable on multiple operating systems, including Windows, macOS, and Linux.
* **Device Independence**: Ensure the application functions seamlessly across various devices, including desktops, tablets, and smartphones.
* **Minimal Dependencies**: Reduce reliance on platform-specific features or proprietary technologies to enhance portability.

1.4.6 **Reusability:**

* **Modular Components**: Design system components to be reusable in different contexts, allowing for easy adaptation to new retail locations or business units.
* **Template-Based Setup**: Use templates for common business functions to streamline the setup process for new deployments.
  1. **Hardware and Software Requirements**

**1.5.1 Hardware**

* Any devices which can support any browser, with an internet connection.
* (Can be Optional, depending upon the medium which the client chooses) Barcode scanner and receipt printer
  + 1. **Software**

**Backend**

* Node.js
* Express.js

**Frontend**

* React.js, UI extensions

**Database**

* MongoDB

**Other tools used in development**

* Git for version control
* VS Code for development

**1.6 Data Dictionary:**

**1.6.1** login\_table

To store admin, customer, and cashier login details.

**Primary Key:** user\_name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name** | **Datatype** | **Constraint** | **Description** |
| 1 | user\_name | Varchar(10) | Primary key | To store the user name |
| 2 | user\_type | Char(1) | Not null | To store user type |
| 3 | password | nvarchar(10) | Not null | To store the password |
| 4 | user\_id | Varchar(10) | Not null | Unique id of the user |

**1.6.2 sales\_table**

* To store sales transaction details.
* **Primary Key:** transaction\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name** | **Datatype** | **Constraint** | **Description** |
| 1 | transaction\_id | varchar(20) | Primary key | To store the transaction ID |
| 2 | item\_id | varchar(10) | Not null  Foreign key reference | To store item ID, |
| 3 | quantity | int | Not null | To store the quantity sold |
| 4 | customer\_id | Varchar(10) | Foreign key reference | To refer Customer table |
| 5 | employee\_id | Varchar(10) | Foreign key reference | To refer employee table |

**1.6.3 inventory\_table**

* To manage inventory details..
* **Primary Key:** item\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Name | Datatype | Constraint | Description |
| 1 | item\_id | Varchar(10) | Primary key | To store the item ID |
| 2 | item\_name | Varchar(50) | Not null | To store the item name |
| 3 | stock\_level | Int | Not null | To store the current stock level |
| 4 | reorder\_level | Int | Not null | To store the reorder threshold |
| 5 | price | Decimal(10,2) | Not null | To store the item price |

**1.6.4 customer\_table**

To store customer information.

* **Primary Key:** customer\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Name | Datatype | Constraint | Description |
| 1 | customer\_id | Varchar(10) | Primary key | To store the customer ID |
| 2 | first\_name | Varchar(50) | Not null | To store the customer's first name |
| 3 | last\_name | Varchar(50) | Not null | To store the customer's last name |
| 4 | email | Varchar(50) | Not null | To store the customer's email |
| 5 | phone\_number | Varchar(15) | Not null | To store the customer's phone number |