

# **Supermarket Automation**

## **Software (SAS)**

### **Testing Plan**

- 1. Agniva Saha - 22CS10003**
- 2. Sayandeep Bhowmick - 22CS10069**
- 3. Srinjoy Das - 22CS30054**

**March-April 2024**

**Prepared for Software Engineering Laboratory,  
CS29006**

**Under the guidance of Prof. Sourangshu Bhattacharya  
and Prof. Debashis Samanta**

## **TEST PLAN OUTLINE (IEEE 829 FORMAT)**

1) Test Plan Identifier	2
2) References	2
3) Introduction	2
4) Test Items	2-3
5) Software Risk Issues	3-4
6) Features to be Tested	4-5
7) Features not to be Tested	6
8) Approach	6-7
9) Item Pass/Fail Criteria	7-8
10) Suspension Criteria and Resumption Requirements	8
11) Test Deliverables	8
12) Remaining Test Tasks	8
13) Environmental Needs	9
14) Staffing and Training Needs	9
15) Schedule	9
16) Glossary	9

# 1 Test Plan Identifier

This test plan outlines the testing procedures for the Supermarket Automation Software (SAS), focusing on the functionality of changing item prices. The plan serves as an executive summary, detailing the purpose, scope, and key considerations for testing. Aligned with the Software Project Plan for SAS, the scope includes resource constraints, testing objectives, and coordination with other evaluation activities. Change control procedures will be implemented, and references to relevant documents will be included for clarity and alignment with project requirements.

## 2 References

1. Software Requirements Specification for this project

## 3 Introduction

The purpose of this test plan is to lay out a methodical framework for testing the various functionalities described in the SRS document submitted earlier. This is needed to ensure that the various components of this web application are working as expected in a wide variety of situations. This helps in delivering a smooth, glitch-free user experience to the students, wardens, etc.

## 4 Test Items

### 1. Sales Transaction Processing

- **Functionality:** Automates the process of scanning items, weighing (if necessary), calculating the total cost, and generating a bill for each customer.

- **Key Features to Test:**

- Compatibility with barcode readers and automatic weighing scales.
- Accurate capturing of item ID, name, quantity, unit price, and total item price.
- Generation of bills with a unique serial number for each sales transaction, itemized list of purchases, and the total amount payable.
- Handling of various payment methods (if applicable).

### 2. Inventory Management

- **Functionality:** Keeps track of the stock levels of items in the supermarket, updating inventory as sales are made, and allowing for manual adjustments when new stock arrives.

- **Key Features to Test:**

- Real-time decrement of inventory levels upon sales transaction completion.
- Functionality for employees to update inventory levels with new stock manually.
- Ability to view current inventory levels, including details like item ID, name, and quantity available.
- Alerts or notifications for low stock levels (if applicable).

### 3. Sales Statistics Reporting

- **Functionality:** Provides detailed sales reports for any given day or period, showing the quantity sold, revenue generated, and profits for each item.

- **Key Features to Test:**

- Accuracy in reporting the quantity of each item sold and the corresponding revenue.
- Calculation of net sales and profit for each item.
- Customizability of the reporting period (daily, weekly, monthly, custom dates).
- User-friendly interface for generating and viewing reports.

### 4. Price Management

- **Functionality:** Allows the manager to adjust the selling price of any item in the inventory.

- **Key Features to Test:**

- Flexibility in changing item prices without affecting ongoing transactions.
- Immediate update of new prices in both the sales interface and inventory records.
- Logging or tracking of price changes for audit purposes (if applicable).

### 5. User Interface and Usability

- **Functionality:** Offers a seamless and intuitive interface for sales clerks, inventory managers, and the supermarket manager to interact with SAS.

- **Key Features to Test:**

- Ease of use for conducting sales transactions, including scanning items and finalizing sales.
- User-friendliness of inventory management functions, such as viewing stock levels and updating them.
- Clarity and accessibility of the sales statistics reporting features.
- Simplicity in changing item prices, with safeguards to prevent accidental adjustments.

By addressing these test items, the development team can ensure that the Supermarket Automation Software meets the comprehensive needs of the supermarket's operations, from sales processing and inventory management to reporting and price adjustments.

## 5 SOFTWARE RISK ISSUES

**1. Inconsistent Database Entries:** Inadequate synchronization of CRUD operations on the database can lead to inconsistencies in data entries. These inconsistencies may result in fatal errors during data retrieval or updates, jeopardizing the integrity of the application.

**2. Incorrect User Management and Privilege Separation:** Improper management of user roles and privileges can pose significant risks to the security and accountability of the system. Failure to restrict certain actions or access levels for specific user classes may lead to unauthorized operations and potential breaches of confidentiality.

**3. Performance Under Heavy Load:** The web application must be capable of handling a substantial load, especially in environments with a large user base like IIT Kharagpur. Failure to ensure responsiveness and stability under high traffic conditions may result in denial of service and user dissatisfaction.

**4. Security Vulnerabilities:** Insufficient security measures can expose the application to various threats, including unauthorized access, data breaches, and malicious attacks. Vulnerabilities in authentication mechanisms, data encryption, and input validation can compromise the confidentiality and integrity of user data.

**5. Compatibility and Integration Challenges:** Integration with external systems, third-party components, or legacy software versions may introduce compatibility issues and functionality gaps. Failure to address these challenges can disrupt system interoperability and hinder the seamless exchange of data between different modules or platforms.

By identifying and addressing these software risk issues proactively, we aim to minimize the potential impact on project timelines, budget, and overall success. Our approach will involve thorough risk analysis, comprehensive testing, and robust mitigation strategies to safeguard the integrity and reliability of the SAS project.

## 6 Features to be tested:

Sure, here are the features to be tested for the URLs configuration, models, views, and forms across the dashboard, user, and staff applications of the SAS project:

### 1. URLs Configuration (urls.py):

- We will ensure that all URLs are correctly configured and mapped to the appropriate views.
- Our goal is to verify that URL patterns adhere to project requirements and follow best practices.
- We'll test URL patterns for consistency and accuracy across all applications (dashboard, user, staff).

### 2. Model Classes (models.py):

- We'll validate the integrity and correctness of model fields, relationships, and constraints.
- Testing model methods and properties will help ensure expected behavior and functionality.
- My focus is on ensuring that models accurately represent the data schema and business logic of the application.
- We'll verify proper handling of edge cases and boundary conditions in model definitions.

### 3. Views and Business Logic (views.py):

- We will test view functions to ensure proper handling of HTTP requests and responses.
- Verifying that views interact correctly with models to perform CRUD operations is essential.
- We will test view logic for authorization, authentication, and permission handling.
- Our aim is to validate data processing and manipulation within views for accuracy and security.
- I will ensure proper error handling and exception management in view functions.

#### **4. Forms and Input Validation (forms.py):**

- Testing form classes for accurate representation of input fields and validation rules is my priority.
- We'll validate form behavior for handling user input, data validation, and error messages.
- Verifying that forms correctly bind and save data to associated model instances is crucial.
- We'll test form submissions with valid and invalid input to ensure proper validation and error handling.
- Our focus will also be on validating form security measures, such as CSRF protection and input sanitization.

#### **5. Dashboard Application:**

- We'll test dashboard-specific URLs, models, views, and forms for functionality and correctness.
- Our goal is to validate dashboard features such as data visualization, reporting, and administrative tasks.
- We'll ensure seamless integration with other components of the SAS project and adherence to project requirements.

#### **6. User Application:**

- Testing user-related URLs, models, views, and forms for registration, authentication, and profile management is crucial.
- We'll validate user authentication workflows, password management, and access control features.
- Our aim is to test user interfaces for usability, accessibility, and responsiveness across devices and browsers.

#### **7. Staff Application:**

- Testing staff-specific functionalities such as order processing, inventory management, and sales reporting is important.
- We'll validate staff-facing URLs, models, views, and forms for accuracy and reliability.
- Ensuring that staff members can perform their assigned tasks efficiently and securely within the application is my focus.

By thoroughly testing these features, We aim to ensure the robustness, functionality, and quality of the software system.

## 7. Features Not to be Tested

There are no such features since the web application won't have future revisions. Therefore, no hidden disabled features will be present. Since this is a one-time project with no further maintenance, all features need to be tested.

## 8. Approach (Strategy)

### 1. Functional and Interface Testing

**(a) Cross-Browser Compatibility:** The tester will launch the web application on a variety of popular browsers (e.g., Chrome, Firefox, Safari, Edge) to identify any compatibility issues. This ensures a consistent user experience across different browsing platforms.

**(b) Link Validation:** All links on the web pages will be thoroughly tested to ensure they function properly and lead to the intended destinations. This eliminates "broken links" that can frustrate users.

**(c) Security Testing:** The tester will attempt to access web pages that should require authentication (login). This verifies the effectiveness of the application's security measures in preventing unauthorized access.

### 2. Subsystem-Specific Testing

**(a) Test Case Execution:** The tester will navigate to relevant web pages for each subsystem and follow detailed test cases. These cases will outline specific actions (e.g., button clicks) and expected system responses.

**(b) Defect Reporting and Resolution:** Any discrepancies between the actual system behavior and the anticipated outcome will be documented in a bug report. Developers will then investigate the issue and implement a fix.

### 3. Client Compatibility Testing

**(a) Multi-Device Testing:** In addition to browser compatibility, the application will be tested on various devices (e.g., Android phones, iPhones, desktops, laptops) to guarantee smooth operation across different operating systems and hardware configurations.

**(b) Network Performance Testing:** The tester will simulate low-bandwidth conditions to assess the application's functionality. This ensures the app remains usable even in areas with poor internet connectivity.

**(c) Client Configuration Testing:** The application will be evaluated under diverse client configurations, including font variations, browser settings, JavaScript and cookie permissions. This identifies potential rendering issues caused by user preferences.

#### 4. Test Metrics Collection

**(a) Performance Metrics:** The tester will collect data on web page loading times to pinpoint performance bottlenecks.

**(b) Load Testing:** The application's response under simulated heavy user traffic will be measured. This helps identify scalability issues and ensures the app can handle peak loads.

**(c) Error Logging:** All errors encountered during testing will be logged and categorized to track bug prevalence and prioritize fixes.

**(d) Client-Side Variations:** Data will be collected on how the application behaves across different devices and configurations to identify inconsistencies.

**(e) Usability Evaluation:** Testers will assess the user interface for potential complexities like convoluted menus or submenus that might hinder user experience.

## 9 ITEM PASS/FAIL CRITERIA

The pass/fail criteria for each item in the testing plan should be based on predefined conditions or expectations that determine whether the item meets the desired quality standards. These criteria are typically established during the planning phase and are used to assess the success or failure of the testing process. Here are some general guidelines for setting pass/fail criteria:

### 1. Functional Requirements:

If the item is related to a specific functionality or feature of the software, the pass criteria may involve verifying whether the functionality behaves as expected, meets the specified requirements, and produces the intended results. The item would pass if it fulfills all functional requirements without errors or deviations.

### 2. Proper Login Functionality:

- Pass Criteria: Users can successfully log in using a valid username and password combination.
- Fail Criteria: Users encounter errors or are unable to log in despite providing correct credentials.



### **3. Password Security Standards:**

- Pass Criteria: Passwords meet security requirements, including at least one capital letter, one special symbol, and a minimum length.
- Fail Criteria: Passwords do not meet the specified security criteria, such as lacking required characters or not meeting the minimum length.

### **3. Username-Password Similarity Check:**

- Pass Criteria: During registration, the system correctly identifies and alerts users if their username and password are similar.
- Fail Criteria: The system allows users to register with a username and password that are too similar, potentially compromising security.

By incorporating these additional criteria, we ensure that the login functionality not only meets basic requirements but also adheres to security standards and integrates smoothly with external authentication methods like Gmail. This comprehensive approach enhances the overall quality and security of the software.

## **10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS**

The tests associated with a subsystem should be suspended in only one case, i.e. when the main web page associated with the specific subsystem does not load in a client. This means other test cases related to that subsystem cannot even proceed. This is the only case where suspension should occur. The tests need not be suspended otherwise, since the isolation subsystems is a core principle. Test failures in specific subsystems can be used to fix errors pertaining to only those subsystems, thus guaranteeing that this failure is completely contained within a specific unit. This also tells us how various components interact with each other, and suspending the tests means we won't be able to gather sufficient information. Resumption occurs once we fix the errors associated with the test failures that prevent the main pages from loading.

## **11 Test Deliverables**

1. Test plan
2. Test cases
3. Bug reports
4. Error logs

## **12 Remaining Test Tasks**

There are no remaining test tasks, We have completed all the test tasks for the current version of our software.

## 13 Environmental Needs

These are the required python libraries that should be installed to run this software

Django  
 opencv-python  
 django-crispy  
 crispy-bootstrap  
 python-barcode

## 14 Staffing and Training Needs

We don't need any kind of staffing and training needs.

## 15 Schedule

The testing will occur simultaneously with the development. The final date is 9 April 2024.

## 16 Glossary

**Bill** - A bill is a commercial document issued by a seller to the buyer indicating the products, quantities and agreed prices for products or services the seller has provided the buyer. It can indicate a sales transaction only.

**Inventory** - It describes the goods and materials that a business holds for the ultimate purpose of sale.

**Bar code** - A barcode is an optical machine-readable representation of data related to the object to which it is attached.

**Automatic weighing machine** - An electronic device which can measure the weight of an object kept on it, and the weight is displayed on an LED display with a high level of accuracy.

**Employee** - people working for the company.

**Sales clerk** - A Sales clerk is an employee whose special activity includes being responsible for carrying out transactions with the customers for the different items in the supermarket.

**Manager** - A Manager is the person who is responsible for supervising the employees and analyzing the sales statistics in a given period of time.

**Loose item** - Items not packed e.g. fruits and vegetables.

**Packaged item** - Packed items e.g. biscuit packets.

**BCR** - barcode reader

**AWM/C** - automatic weighing machine

**SAS** - Supermarket automation software