HuaChang Growmax Sales Order Application

TEST PLAN

Musang King

List of your Names:

Name	Position	email	phone
Benjamin Tan Chen Hern	Leader	104477174@student.swin.edu.au	+601110660387
Wallace Iglesias Chandrio	Member	104180579@student.swin.edu.au	+60166082100
Hein Htet Naing	Member	104329055@student.swin.edu.au	+60134942987
Mahanthe Acharige Sachindri Sudeepa Chandrasiri	Member	104338967@student.swin.edu.au	+60108932476

SWE40001 Software Engineering Project A, Year 2 Semester 2, 17 / 11 / 2024

Table 1. Document Change Control

Version	Date	Authors	Summary of Changes
1	17 / 11 / 2024	Benjamin Tan Chen Hern Wallace Iglesias Chandrio Hein Htet Naing Mahanthe Acharige Sachindri Sudeepa Chandrasir	Initial Draft

Table 2. Document Sign Off

[all group members need to sign off]

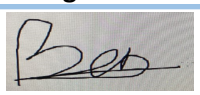
Name	Position	Signature	Date
Benjamin Tan Chen Hern	Leader		17/09/2024
Wallace Iglesias Chandrio	Member		17/09/2024
Hein Htet Naing	Member		17/09/2024
Mahanthe Acharige Sachindri Sudeepa Chandrasiri	Member		18/09/2024

Table of Contents

1. Introduction

1.1. Purpose of the test plan

The test plan outlines the scope, resources, approach, strategy, and criteria for testing activities conducted during and after the development of the Huachang Growmax Sales Order Application. It identifies the features to be tested, features excluded from testing, and the stages at which tests will be conducted. The goal is to ensure that the application is functional, reliable, and meets the client's requirements.

1.2. Scope

Testing will be carried out throughout the software development lifecycle to identify and resolve defects early, ensuring the quality and reliability of the application. This test plan currently covers high-level test cases and assumptions due to the ongoing refinement of the system's architecture and features. As the system evolves, the plan will be updated.
The application to be tested is the **Huachang Growmax Sales Order Application**, which includes features such as sales order creation, confirmation, transaction tracking, data export for integration with the AutoCount system, and basic reporting functionalities. Testing will primarily include:

- **Unit Testing**
- **Integration Testing**
- **System Testing**
- **User Acceptance Testing (UAT)**

Constraints/Assumptions

- Testing will be conducted on the application running on standard desktop and mobile devices.
- A stable internet connection (Wi-Fi or LAN) is required for testing the data export functionality.
- Test data will simulate real-world usage scenarios.

Testing Strategy

The testing process will follow these levels to ensure the application functions effectively:

1. **Unit Testing**: To verify individual components/modules.
2. **Integration Testing**: To ensure modules work together as intended.
3. **System Testing**: To validate the system's functionality as a whole.
4. **User Acceptance Testing (UAT)**: To ensure the application meets client requirements and expectations.

1.3) References Material

This document has been created using the following references:

- Project Plan
- Software Quality Assurance Plan
- Software Requirements Specification (SRS)
- Software Quality Assurance Plan (SQAP)
- IEEE 829 Standard template (Test Plan)

1.4) Objective

- Conduct appropriate tests during various development stages to ensure quality and performance.
- The test plan will serve as a guideline for different test types required to maintain system efficiency and reliability.

1.5) Resources Required

- Desktop/laptop computers (for testing the application's desktop interface).
- Mobile devices (for testing the application's mobile interface).
- Test accounts for the application (Administrator, Manager, and Sales Staff).
- Stable internet connection (Wi-Fi or LAN).
- Data samples for testing (e.g., sales orders, stock details).
- Development tools and testing frameworks (e.g., IDE, automated testing tools).

1.6) Environment Requirements

- Devices with the latest operating systems (Windows/macOS for desktop; Android/iOS for mobile).
- Stable internet connection to simulate live scenarios.
- Access to test environments for staging and production.
- Access to mock databases with sample data for testing.
- Tools for log capturing and debugging (e.g., logging software).

2. Test Items

Features to Be Tested

1. **Form Functionality**
    - Ensure all input fields accept valid data (e.g., product name, quantity, price).
    - Test validation for invalid data entries (e.g., negative numbers, invalid characters).
2. **Order Submission**
    - Verify the order form submits data correctly to the backend.
    - Confirm proper handling of success and failure scenarios.
3. **UI/UX Consistency**
    - Check that the layout is responsive across devices.
    - Test for accessibility compliance (e.g., WCAG guidelines).
4. **Error Handling**
    - Ensure proper error messages are displayed for incomplete or incorrect submissions.
5. **Data Security**
    - Test for protection against common vulnerabilities like SQL Injection or XSS.
6. **Order Confirmation**
    - Verify users receive confirmation upon successful submission.
    - Validate the backend records the order accurately.
7. **User Role Management**
    - Test if functionalities align with the roles (superadmin, admin, salesperson)

2.2 Test Cases

2.2.1 Functional Test Cases

Test Case 1: Valid Input Data in Form Fields

Steps:

1. Navigate to the sales order form.
2. Enter valid data in all fields:
    - Product Name: "Widget A"
    - Quantity: "10"
    - Price: "20.50"
3. Click the "Submit" button.

Expected Results:

- Form accepts the input without errors.
- Data is correctly formatted and sent to the backend.

Test Case 2: Validation for Invalid Input Data

Steps:

1. Navigate to the sales order form.
2. Enter invalid data:
    - Product Name: Leave blank.
    - Quantity: "-5" or "abc".
    - Price: "-10.00" or "twenty".
3. Click the "Submit" button.

Expected Results:

- Form displays appropriate error messages for each invalid field:
    - Product Name: "This field is required."
    - Quantity: "Enter a positive number."
    - Price: "Enter a valid price."
- Submission is blocked until all errors are corrected.

Test Case 3: Successful Order Submission

Steps:

1. Fill in the form with valid data.
2. Click the "Submit" button.
3. Check the response from the backend.

Expected Results:

- User receives a success message: "Order submitted successfully."
- Order data is saved correctly in the backend database.

Test Case 4: Failed Order Submission

Steps:

1. Simulate a network error or backend unavailability.
2. Fill in the form with valid data.
3. Click the "Submit" button.

Expected Results:

- User receives an error message: "Submission failed. Please try again later."
- Data is not sent multiple times (no duplication).

Test Case 5: Responsive Layout

Steps:

1. Open the sales order form on various devices (e.g., desktop, tablet, mobile).
2. Resize the browser window to different widths.
3. Observe the layout and functionality.

Expected Results:

- Form adjusts layout and remains functional across all screen sizes.
- No overlapping or cutoff elements.

Test Case 6: Accessibility Compliance

Steps:

1. Use a screen reader to navigate the form.
2. Test keyboard navigation (Tab, Enter keys).
3. Check color contrast and text sizes.

Expected Results:

- Screen reader reads all form labels and messages.
- All fields are accessible via keyboard.
- Text is legible, and color contrast meets WCAG guidelines.

Test Case 7: Error Handling for Missing Fields

Steps:

1. Leave one or more required fields blank.
2. Click the "Submit" button.

Expected Results:

- Form highlights the blank fields with a red border or error icon.
- Error message: "Please fill out this field."
- Submission is blocked until all required fields are filled.

Test Case 8: Data Security

Steps:

1. Enter malicious inputs in fields:
    - Product Name: `<script>alert('XSS')</script>`
    - Price: `1; DROP TABLE orders;`
2. Click the "Submit" button.

Expected Results:

- Inputs are sanitized, and no malicious code is executed.
- Submission is rejected for SQL Injection or XSS attempts.
- Security log records the incident (if applicable).

Test Case 9: Order Confirmation

Steps:

1. Submit an order with valid data.
2. Check for confirmation on the UI and in the backend.

Expected Results:

- User receives a confirmation message: "Order #1234 submitted successfully."
- Order details are correctly recorded in the backend.

Test Case 10: Role-Based Functionality

Steps:

1. Log in as each role (superadmin, admin, salesperson).
2. Attempt to access and perform role-specific tasks:
    - Superadmin: Access all orders, manage users.
    - Admin: Access assigned orders, update orders.
    - Salesperson: Create and view their own orders.

Expected Results:

- Each role sees only the functionalities assigned to them.
- Unauthorized actions display an error message: "You do not have permission to access this feature."

2.2.1. Non-functional Testing

Testing Goal

The purpose of the following test cases is to ensure that the Hua Chang Growmax sales order website is both **functional** and **user-friendly**, with a minimal learning curve. The aim is for the average user, regardless of technical expertise, to efficiently navigate and perform essential tasks such as creating, submitting, and managing sales orders.

Testing Procedure

1. The test will involve **real participants** acting as sales team members (salesperson, admin, and superadmin).
2. Each participant will be tasked with performing basic actions on the sales order website.
3. Participants will provide **feedback** on task difficulty and offer suggestions for improving the user interface.
4. Observers will monitor user interactions and record time taken for task completion.

Pass/Fail Criteria

- **Time to Complete**: Each task should be completed within **30 seconds**.
- **Difficulty Goal**: Participants will rate task difficulty on a scale of 1 to 10. The average score for each task must be **≤ 5** (1 = very easy, 10 = very hard).
- **Task Success**: All participants should successfully complete assigned tasks without requiring external help or instructions.

Testing Constants

1. Tests will be performed using the **Hua Chang Growmax sales order website** in its latest deployed version.
2. All tasks will involve **sample sales order data** provided to the participants.
3. Participants will be pre-briefed on their roles:
    - **Salesperson**: Creating and viewing their own orders.
    - **Admin**: Managing orders assigned to them.
    - **Superadmin**: Managing all orders and overseeing the system.
4. All tests will be conducted in a **controlled environment** using standard devices (laptops/desktops with a browser).

Testing Assumptions

- Participants are **new to the application** and have minimal prior exposure.
- Tasks represent typical actions users would perform during daily operations.
- Feedback from participants reflects the usability experience for a broader audience.

Sample Test Cases

Task 1: Create a New Sales Order

1. Participant Role: **Salesperson**
2. **Steps**:
    - Log in using salesperson credentials.
    - Navigate to "Create Order" page.
    - Fill in order details (e.g., product name, quantity, price).
    - Submit the order.
3. **Pass/Fail Criteria**:
    - Time: Completed within 30 seconds.
    - Feedback: Difficulty rating ≤ 5.
    - Result: Order appears in the salesperson's order list upon submission.

Task 2: Approve an Assigned Order

1. Participant Role: **Admin**
2. **Steps**:
    - Log in using admin credentials.
    - Navigate to the "Pending Orders" section.
    - Review the order details.
    - Click "Approve."
3. **Pass/Fail Criteria**:
    - Time: Completed within 30 seconds.
    - Feedback: Difficulty rating ≤ 5.
    - Result: Order status changes to "Approved," and a confirmation message is displayed.

Task 3: Generate a Sales Report

1. Participant Role: **Superadmin**
2. **Steps**:
    - Log in using superadmin credentials.
    - Navigate to the "Reports" section.
    - Select the date range for the report.
    - Generate and download the report.
3. **Pass/Fail Criteria**:
    - Time: Completed within 30 seconds.
    - Feedback: Difficulty rating ≤ 5.
    - Result: Report is generated and successfully downloaded in the specified format.

Task 4: Edit a Sales Order

1. Participant Role: **Salesperson**
2. **Steps**:
    - Log in using salesperson credentials.
    - Navigate to the "My Orders" section.
    - Select an order to edit.
    - Update the quantity and price fields.
    - Save changes.
3. **Pass/Fail Criteria**:
    - Time: Completed within 30 seconds.
    - Feedback: Difficulty rating ≤ 5.
    - Result: Updated order is displayed with the new details.

Task 5: Handle Invalid Data Entry

1. Participant Role: **Any**
2. **Steps**:
    - Attempt to create an order with invalid data (e.g., negative quantity or price).
    - Observe the error message.
3. **Pass/Fail Criteria**:
    - Time: Error message displayed within 5 seconds.
    - Feedback: Difficulty rating ≤ 5 for understanding and resolving the issue.
    - Result: System prevents invalid data submission and highlights affected fields.

Task 6: Navigate Through the Website

1. Participant Role: **Any**
2. **Steps**:
    - Log in with assigned credentials.
    - Browse through various sections (Dashboard, Orders, Reports, etc.).
    - Return to the homepage.
3. **Pass/Fail Criteria**:
    - Time: Navigation through all sections completed within 1 minute.
    - Feedback: Difficulty rating ≤ 5 for navigation clarity.
    - Result: No confusion or dead-ends during navigation.

Feedback Collection

At the end of the testing session, participants will provide:

1. Difficulty ratings for each task.
2. Suggestions to improve the user interface.
3. General comments about their experience.

2.3. Features not to be Tested

Since all the features are being implemented for the first time and are all capable of being tested. There are currently no features that will not be tested.

3.Strategy

To guarantee a dependable, effective, and user-friendly system, the HuaChang Growmax sales order application will go through an extensive testing process using a four-level approach. The following steps are part of the testing strategy:

- **Unit Testing –** As various modules (such as order management, stock availability, and customer management) are created by team members, this step will be completed within the first coding phase. To ensure that errors are found early in the development process, developers will write unit tests to validate individual components.
- **Integration Testing –** This test verifies that all modules (such as the database, front-end, and back-end) work together when the system reaches a point where several modules are integrated. To ensure smooth data flow and operation, developers and the team leader will test how various components interact with one another.
- **System Testing –** This phase will be carried out once the system is functional enough to carry out essential business tasks such processing payments, managing customer data, creating sales orders, and determining stock availability. System testing, which focuses on both functional and non-functional needs including usability and performance, will confirm that all requirements are met.
- **Acceptance Testing –** Acceptance Testing: This last stage of testing makes sure the customer is happy with the system. To ensure that all requirements are met and the system is prepared for deployment, the client will have to try it first hand to ensure that both parties are satisfied with the end result.

3.1. Roles and Responsibilities

- **Unit Testing –** Carried out constantly when working on various system parts by individual developers. Unit tests must be written by each developer in order to verify their code.
- **Integration Testing –** Under the direction of the team leader, the development team conducts integration testing. To ensure that modules interact properly, the team will employ test cases.
- **System Testing –** System testing is carried out by a specialized testing team under the team leader's direction. To evaluate usability and functionality, the testing team will include those who have experience in web-based applications.

- **Acceptance Testing –** Conducted with the client by a different testing team. To make sure the system satisfies practical requirements, a select group of end users will participate to ensure all the requirements are met. The project manager will oversee testing to ensure that all customer criteria are met.

3.2. Test Deliverables

- Test Plan: Document outlining the testing strategy, scope, and schedule.
- Test Cases: Detailed scenarios and test cases covering all system functionalities.
- Defect/Enhancement logs: Logs to track and report bugs or suggested improvements.
- Test Reports: Summaries of testing results, including issues found and bugs resolved.
- Usability Test Guide: Instructions for conducting usability tests, ensuring user satisfaction with the interface and workflow.

3.3. Schedule

Table 5. Semester 2 Schedule

Features to test	**Estimated Testing**
User Authentication & Role-based Access	End of June
Order Management (Create, Update, Track)	End of June
Stock Availability Checks	End of June
Customer Management	End of June

3.4. Risk and Contingency

Table 7. Risks associated with testing

Risks	**Contingency**
Limited time available for testing	Perform testing iteratively as modules are developed to avoid last-minute rush.
Changes in client requirements during development	Regularly update and confirm requirements with the client to prevent delays.
Design limitations preventing easy modifications	Design the system with flexibility to accommodate potential changes.

3.5. Testing Tasks

- Test scenarios will be developed based on client requirements and documented for reference.
- Test cases will be created to validate individual features such as order management, stock availability, and payment processing.
- Necessary resources (e.g., testing environments, tools, and personnel) will be allocated before conducting tests.
- Bugs will be logged systematically and tracked for resolution using a defect management tool.

4. Pass/Fail Criteria

4.1) Product Level

The Pass/Fail criteria ensure the system meets functional and non-functional requirements as specified. The table below outlines specific test cases and their criteria.

Testing Criteria	**Pass Criteria**	**Fail Criteria**
Login and Authentication	Users can log in securely using valid credentials and access role-specific features.	Users cannot log in or access restricted features not relevant to their role.
Order Creation	Sales representatives can create orders, save them, and view order summaries accurately.	Errors in saving orders, missing information, or inaccurate summaries.
Stock Availability Check	Inventory data is displayed in real-time when queried, with updates reflecting current stock.	Stock information is inaccurate or fails to display.

Order Status Tracking	Administrators and sales reps can view order statuses and update progress.	Status updates fail to save or are inaccurately reflected.
Price History Retrieval	Customer-specific pricing history is displayed correctly within seconds of query.	Incorrect or missing pricing history information for selected customers.
Notifications	System sends accurate notifications about order status changes or issues.	Notifications are delayed, incorrect, or not sent at all.

4.2. Testing Stages

Testing is performed during each development stage and is evaluated based on the following criteria:

Testing Level	**Pass criteria**	**Fail criteria**
Unit Testing	• Individual components, such as order creation, inventory updates, and user authentication, function correctly.	• Components fail to perform expected tasks, or unit tests reveal critical bugs.
Integration Testing	• Data flows smoothly between modules (e.g., Order Management and Inventory) without errors or inconsistencies.	• Communication between modules is incomplete or incorrect, resulting in failed transactions or data loss.

System Testing	● The full system meets functional and non-functional requirements, with all workflows operating seamlessly.	● Significant defects prevent the system from fulfilling core functionalities such as order processing or access control.
Acceptance Testing	● End users (sales representatives, administrators, and warehouse staff) can complete key tasks, and the system receives client approval.	● Users encounter critical issues preventing them from completing tasks, or the client rejects the deliverables.

4.3. Suspension criteria and resumption requirements

4.3.1. Suspension Criteria

Testing activities will be temporarily halted under the following conditions:

- **Critical Defects**: Severe bugs or errors that prevent the system from performing essential functions, such as order creation or inventory updates.
- **System Instability**: Frequent crashes, inability to log in, or unexpected behavior affecting multiple modules during testing.
- **Unavailability of Test Environment**: Key testing resources, such as the database, server, or payment gateway integration, become inaccessible or non-functional.
- **Requirement Changes**: Significant alterations to system requirements that necessitate design or functionality updates before further testing can proceed.

4.3.2. Resumption Requirements

Testing will resume only after the following conditions are met:

- **Critical Issues Resolved**: All identified high-priority defects are fixed, and affected modules pass retesting.
- **System Stability Verified**: The system runs without frequent crashes, and all critical functionalities are operational.

- **Environment Restored**: All required testing resources, such as databases or third-party integrations, are fully operational.
- **Requirement Updates Implemented**: Changes to system requirements are addressed, and the updated modules meet the defined specifications.

4.4. Approvals

- **Team Leader Approval**:
    - Confirms that all critical functionalities in the current testing phase meet the defined Pass Criteria.
    - Signs off on the successful resolution of critical defects and retesting results.

- **Testing Manager Approval**:
    - Verifies that the system meets all testing goals, including functional and non-functional requirements.
    - Reviews and approves test case results, ensuring no unresolved critical issues remain.
- **Client Approval**:
    - During User Acceptance Testing, the client reviews and approves the system's performance, functionality, and usability.
    - Final approval is required before deployment to production.