Test Plan

for

Online Sales Portal

Version 1.0 approved

Prepared by Group-14

Hrushikesh Reddy-21CS30028

Shivang Agrawal-21CS30048

Hari Krishna-21CS30025

Aseem Anand-19CS10013

CSE, IIT Kharagpur

Table of Contents ii

1.Test Plan Identifier :Online Sales Portal Test Plan

2. Introduction

2.1 Purpose

The purpose of this test plan is to define the testing strategy and procedures for the Online Sales Portal.The testing will be performed to

verify that the system meets the functional and non-functional requirements as specified in the project document. The testing will also be performed to ensure that the system is reliable,efficient,and user-friendly.

2.2 Testing overview

The pattern for testing the system is to identify the various major functions that may cause bugs.

The basic philosophy to design the test design suite is to maximize discovery of bugs so that they can be debugged before release.

First, review the test data and test cases so that they are exhaustive for each unique unit's verification.

Identify the expected results.

The expected results are documented.

The tests are performed and the results are compared with the expected results.

Any specifications/details to be reviewed are reported

3 Test Items

3.1 Unit testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness. For the testing purposes, JUnit library is used. The following functions are some of the functions tested.

- register(), registermanager()
- login(),loginmanager()
- home_user()
- logout()
- home_manager()

- manager_details()
- manage_customer()
- deleteseller(),activateseller(),deleteuser(),activateuser()
- manage_product()
- deleteproduct(),activateproduct()
- logoutmanager()
- sellproduct()
- cart()
- orderstatus()
- userdetails()
- search()
- viewproduct()
- addtocart()
- buycart()
- buyproduct()
- salesstatus()
- viewbalance()

The way we have implemented unit testing is in a white box testing fashion as we are testing directly within the application at the source code level.

4.Features to be Tested

- Sign-up as a manager
- Sign-up as an user, Login as an user
- Upload products for sale
- Search,View products,View Cart
- Go to Home
- Explore different categories of products
- Buy Products, Add to cart,Buy from cart
- Check User details
- Check order status
- Check sales status
- View Balance and add balance

- Logout from the user,login from a different user and test buying products uploaded by previous user.
- Login as a Manager
- View Active customers,sellers,products,order details.
- View manager account details
- Manage customer,seller,products
- Logout

5. Features not to be tested

- Add multiple instances of same product to cart individually and buy them.

6. Approach

6.1 Unit Testing

Each unit is tested thoroughly and independently to make sure it has no error . These tests have been performed and the screenshots are shown in the test report.

6.2 Integration Testing

Testing of the integrated system to ensure that the components work together correctly.

6.3 Interface Testing

The software has been tested on machines with different configurations(for example computers with different operating systems like Windows or Linux) and the functionality of the interface has been verified.

6.4 Use Case Testing:

This is the testing for bad inputs. - if during login or signup details were left blank or wrongly entered, then appropriate error message is shown. - if details entered were wrong or in the wrong format, error message is shown.

6.5 Alpha Testing

The software was tested by 2 students other than the creators to ensure it was working correctly.

:

6.6 Black box testing

Black box testing is the type of testing where the entire application's internal working(and source code) is abstracted and all possible inputs are tested on. In this testing phase, every major use case is given both right inputs and erroneous inputs. On giving the right inputs, the program should give the expected result and on giving erroneous inputs, the program should give an error message. The below are some of the major features tested.

Login and register

- Case when user enters valid and invalid credentials
- Case where email is already registered
- Case where OTP is not generated
- Case where OTP entered does not match

Search Product

- Case when no Product exists with the given query.
- Case when search button is pressed but no query is given.
- When the buyer status is inactive i.e he cannot buy as decided by the manager
- Search product by category when the buyer status is inactive

Display Product

- Display product only when product is active and has more than zero copies for sale.

Buy Product

- Buying the product when there is not enough balance
- Buying all the Items from a cart when there is not enough balance
- Viewing cart to buy when buyer status is inactive

Here, the application is tested on the interface level and appropriate screenshots are given wherever required.

6.7 White box testing

White box testing is the testing at the application's source code level. Major features and control flows are tested in this process. Unit testing falls under this category. Apart from that, major control deciding elements like buttons were tested whether the right action was taken by writing print statements in the listeners.

7. Pass/Fail Criteria

Failing Criteria: The test is considered failed if any of the following is encountered:
• The software crashes
• It results in wrong output
Passing Criteria :The test is considered passed if any of the following is encountered:
• The correct results are obtained

8. Test Deliverables

The following test deliverables will be provided for the Online Sales Portal:

•Test plan
•Test cases and scripts
•Test results and defect reports
•Test summary report

9. Remaining Test tasks

The remaining test tasks for the Online Sales Portal include:

• Execution of test cases
• Defect tracking and resolution
• Retesting of fixed defects
• Test summary report preparation

9.1 Integration testing

The primary objective is to test whether different modules of the software (which have been tested before through white and black box testing) work together in tandem when integrated. The two main modules of the software are the DB and GUI modules. A submodule is the PDF generation and viewing module. But as both use functions from two external libraries which have already been tried and tested by a huge user community, the tests have been avoided for that.

9.2 GUI Module

Each button and component is checked for correct behaviour.

Buttons should either perform a specific task assigned (like generate receipt button) or help navigate to different panels (Logout, Back).

Checks need to be done whether textfields with compulsory values accept null values.

Situations where the user is ”stuck” in a particular panel i.e the user cannot go to another panel as no navigational component is present are enlisted

9.3 DB module

Check whether the data received from database is correctly populated on the interface.

Check conditions when either no data is received or too much of data is received

9.4 System Testing

The main goal of system testing is to validate the software as a whole i.e its performance aspects, security, portability and configuration aspects.
The performance aspects and the function validation are done as part of this test plan.
For function validation, the behaviour of functions is cross checked with the expected outputs of the corresponding use cases in the SRS document.
As expected, the above system tests have been implemented in a black box testing fashion.

10.Environmental needs

The following test deliverables will be provided for the Online Sales Portal:

Hardware:

Modern Operating System:

Windows 7 and higher

Mac OS X 10.11 or higher, 64-bit

Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)

4 GB RAM

Working internet connection

Software:

The requirements for a client are:

- Operating System:- Windows/ Linux
- Html/CSS/Javascript
- python
- Flask
- Mysql