Test Plan

for

FRSS v1.0

Submitted by: Group 14

Suryam Arnav Kalra : 19CS30050

Kunal Singh : 19CS30025

Abhinandan De : 19CS10069

Contents :

Test Plan Identifier

This document describes the framework and approach that has to be used for GUI and backend testing for FRSS application version 1.0 .

References

Documents that support this test plan include:

- Software Requirements Specification
- Use Case diagrams
- Class Diagrams
- Test Suite

Introduction

This process of testing our GUI application has the following objectives:

1. To identify and define the tools that have to be used in the process.
2. To define how the tests will be conducted.

Test Items

The testing process can be partitioned into two major categories:

1. To test the backend of the application using Unit Test APIs.
2. To test the GUI frontend using manual testing, following the test suites.

These will be tested on both Windows and Linux based platforms.

Features to Be Tested

1. As a user:
    1.1 Login
    1.2 Signup
    1.3 LogOut

2. As a customer:
    2.1 Searching for furniture
    2.2 Buying an item on rent.
    2.3 Buying an item on loan.
    2.4 Returning an item.
    2.5 Viewing and paying the amount due.
    2.6 Viewing the past order history.
    2.6 Viewing the various items available in the catalog.

3. As an admin:
    3.1 Creating a customer account.
    3.2 Deleting a customer account.
    3.3 Checking the notifications.
    3.4 Checking the investment and profit.
    3.5 Changing the price of any item.
    3.6 Verifying returns.
    3.7 Adding new furniture to the inventory.

Features Not to be tested

Every care has been taken to ensure that each and every function or feature of the code has been extensively checked through multiple unit tests and final integration testing.

No feature is not to be tested.

Approach

All the test cases that have been generated in Test Suite will be tested. For the frontend, manual verification of the tests have to be done. The testers will mark each test as pass/fail according to the degree of matching with the golden output.

If a particular test fails, the bug reports are documented and bug fixes are recommended.

Once complete, the whole report is then analysed and then bug fixes are made. The whole process is conducted once again (to ensure that new bugs haven't been introduced) until 100% accuracy is obtained.

Pass/Fail

The requirements that have to be met are different for the two partitions of our test plan
At the Unit Test level we could consider:

1. If all test cases passed successfully i.e. their output matches with the golden output.
2. If all methods associated with a particular class have been tested.

For the Front end testing the requirements are:

1. The visual output matches with the expected output.
2. All the various tools and buttons on all windows have been tested.

If any of the requirements are not met, the test is declared as failed.

Suspension Criteria and Resumption Requirements

If the code fails on two or more tests , the tests are suspended and the codebase is worked upon.
After thoroughly running unit tests on the refactored codebase , resumption of test sets will be done.

Software Risk Issues

There are some inherent software risks such as such as :

- Crashing of the database provided by MySQL.
- End of support from MySQL.
- Modification of tkinter library which may not support some old functionality.
- Some functions are complex which may rupture the normal flow of the program.

Environmental Needs

There are several environmental needs which should be fulfilled:

- Access to the databases or the master tables is required.
- The test.py file containing the test functions should run simultaneously.
- The databases must be filled with a variety of products to ensure testing of each and every aspect.

Schedule

The first test of sets will be done everyday for a week before the final completion of the codebase. To ensure no bugs remain a final set of tests will be done 12 hours prior to the submission of the code.

Planning Risks and Contingencies

If the first set of tests is not conducted , it may leave the system buggy and could delay bug fixes and final testing.
The failed tests will be worked upon to modify the codebase prior to final application testing which would then ensure the bug free nature of the code.

Glossary

GUI : Graphical User Interface
API : Application Programming Interface
SQL : Structured Query Language
FRSS : Furniture Rental Store System