

Test Case: SystemTest 1

Test Name: GUI functional testing.

Description: Graphical User Interface (GUI) testing. Doing a thorough examination of all functional aspects of the interface such as buttons, text boxes, list boxes, menus, and toolbars.

Steps:

1. Clicking on every button that is available to observe the behavior of the application.
2. Checking all boxes that can readily accept inputs from the user.
3. Clicking on menus and toolbars to see if it opens the correct page or bar as intended.
4. After reviewing the results and recording it, go back to step 1 and repeat the process multiple times to check for consistency.

Test Data: Clicking through all available buttons using mouse & keyboard and touch screen responsiveness. Typing in words, numbers, symbols, and other inputs available on keyboard/tablets/smartphones.

Expected Results: All buttons are responsive through every hardware that will be utilizing this application. Typing into the software is responsive and outputs correctly when inputted from the user. Moving through this application utilizing the menu, and other features in the toolbar are responsive.

Actual Results: All buttons were responsive, and inputting various number of texts, symbols, and other keyboard inputs were outputting correctly. Clicking through the menu and tool bars worked as intended.

Pass/Fail: Pass

Remarks: Due to the POS system being partly reliant on a stable internet connection, I found that some elements on the menu/tool bar did not work if the system was not connected to the internet. Otherwise, everything worked just as intended when a reliant internet connection was present.

Test Case: SystemTest2

Test Name: GUI load testing.

Description: Doing a thorough examination of graphical aspect of the interfaces such as icons, logos, colors, and page layouts are loading properly.

Steps:

1. Refer to the Design Team's documentation for the proper layout of the GUI.
2. Install the software on all types of hardware.
  - a. PC (1080p, 1440p, 4k)
  - b. Phones/Tablets (Various resolution from vertical/horizontal POV)
  - c. POS hardware with cash registrar implementation. (Various Resolutions)
3. Load into the application and verify that all icons, logos, colors, and layouts are loading in properly.
4. Input data into list boxes and text boxes of the POS system to check if the list is outputting correctly when adding new data into it.
5. Check if uploading a logo from the user affects the output of the page layout.
6. Restart the application and repeat the process to be certain of consistency.

Test Data: Typing in words, symbols, and other inputs available from keyboards/tablet/phones. Utilizing different logos with differing dimensions to upload for testing purposes. Testing various resolutions from various devices.

Expected Results: All inputs from any hardware will not change the layout of the GUI. All resolutions from various devices are outputting all elements in the GUI properly.

Actual Result: There were unintended stretching in the logos and icons when utilizing 1440p or 4k resolution screens. All colors were implemented correctly. The page layouts were displaying correctly on almost all devices. There seemed to be inconsistent loading of the page layout on mobile devices such as tablets/phones when switching from horizontal to vertical view as well the opposite way.

Pass/Fail: Fail

Remarks: The unintended stretching comes from uploading logos and icons that the client requested. Testing a logo made in 1080p resolution and uploading it to a system that is utilizing 4k resolution resulted in some stretching of the image. Therefore, the POS system requires some image scaling technology to compensate the various number of variables in resolution, or the client must provide the hardware specs before we upload the correct image resolution to the software.

Test Case: SystemTest3

Test Name: Installation and Compatibility Testing

Description: Testing the POS application compatibility for all operating systems and hardware.

Steps:

1. Create an environment to replicate all hardware that are required to operate a clothing store business.
  - a. PC/Tablet/Phone (Windows/Android/MacOS/IOS)
  - b. Cash register
  - c. Receipt Printer
  - d. Scanner
  - e. Printer
2. Install the POS application onto the hardware.
3. Verify that the installation process is working as intended.
4. Verify after installation that the application is working as intended.
5. Uninstall application to verify that the uninstallation process is working as intended.

Test Data: Using all the various hardware that replicates a clothing store environment. Each hardware has their own operating system. Using different types of receipt printers/printers/scanners/cash registers available and frequently utilized in the clothing store business.

Expected Result: The application is compatible with all various OS, and hardware listed in the testing.

Actual Result: The POS system is compatible to the various OS listed in the testing. The installation/uninstallation worked as intended on all the compatible devices. Unfortunately, there were some hardware such as scanners/cash registers/printers that did not work with our POS system. These devices were recorded and logged in a separate file for review.

Pass/Fail: Fail

Remarks: There were some compatibility issues that should be fixed with a patch. Once this POS application is out in market, I am worried that new hardware might have compatibility issues as well. It would be wise to allow an avenue for patches to future hardware that need to work with our software.

Test Case: SystemTest4

Test Name: Payment Exception Handling

Description: Testing exception handling if an unexpected error occurs when processing a payment from a customer in a clothing store. Such as, if their credit/debit card does not have any money, or if there is a system error where the card was not read correctly, the pos system should alert the user that the payment did not go through and display the correct error message. The system then should process the incorrect transaction and recover accordingly to be able to process payment again. If the payment does go through, the test will check if exception will be thrown if there is any corruption of data when updating transactional and sales data. This can occur due to hardware malfunctioning, network error, or unknown circumstances.

Steps:

1. Utilize various payment methods that can be processed by the POS system.
2. Test various inputs such as negative balance in the credit/debit card, hardware issues where the credit/debit card reader does not scan properly or anything with human error such as inserting a card that is not a form of payment such as library cards, etc....
3. Observe the POS application behavior when the payment process does not go through.
4. Observe the POS application when the payment process does go through.
5. Record the observation and repeat the test and make sure the exception handling is working as intended.
6. Check if the system is recovering properly after an exception handling is set off.
7. Observe the POS application transaction history update.
8. Emulate a test to see if an exception is thrown if the customer data from the transaction is updated incorrectly due to corruption of data.
9. Observe the POS application sales data update.
10. Emulate a test to see if an exception is thrown if the sales data from the transaction is updated incorrectly due to corruption of data.

Test Data: Different dummy credit/debit cards with various amounts of money. Which includes negative balance, or a small balance which can only pay a portion of the net amount required for the transaction. Emulation of credit/debit card readers that can duplicate common errors that occur regularly. Emulating network error, hardware malfunction, and any other creative ideas of causing corruption in the system.

Expected Result: During the POS transaction process, the exception handling should alert the user if there were any error that occurred during the payment process.

Actual Result: Exception Error was thrown when expected and the application recovered accordingly well. Though the various emulation, whenever there was any sort of corruption of data occurred the software alerted accordingly and reverted any changes that might negatively affect the data set. The only issue that occurred was when we emulated a power outage during the payment process. When restarting the application there were some corrupted data present, and no exception handling was thrown.

Pass/Fail: Pass

Remarks: It would be wise to alert the client that in an event of a power outage that they will have to revert any changes that occurred during the outage manually.

Test Case: SystemTest 5

Test Name: POS Application Volume Testing

Description: Testing the system performance when it must process an enormous amount of data. For a clothing store POS system, it will require processing a long list of information on products which include ID, attributes, and pricing, as well as customer data which includes payment info, customer info, and sales data. The data set is uploaded and downloaded from the cloud system and therefore this test observes and verifies the capabilities of this POS system when handling various amount of data.

Steps:

1. Create a data set of various amounts.
2. Upload a data set into the POS system.
3. Check the performance of the system when the data set is entered.
4. Check if the POS system outputs at the required speeds when requesting data.
5. Check if the POS system is working at the required speeds when inputting new data.
6. Check if the POS system is deleting at the required speeds when erasing data.
7. Check if the POS system is optimized well enough to upload to the cloud at required speeds.
8. Repeat the test by entering a new data set with a different size of data.

Test Data: A various amount of data sets that include product/customer/sales info. The data sets range from a small amount to an extremely large amount.

Expected Results: As we test the various amount of data set, the POS application should be able to handle large amounts of volume that are expected for a clothing store.

Actual Result: During the volume testing, the application was able to handle the required data sets to operate a clothing store franchise. The system performance on various features our POS system offers ran accordingly to the required speeds. When we tested an excessive amount of data is when we observed performance slowing.

Pass/Fail: Pass

Remarks: Due to the POS system requiring only string/int/float data types, it would be extremely difficult and require an enormous amount of time to input the amount that would cause a system performance decrease.