Point of Sale (POS) Testing

Author:
Divya Madaan
Practice Head – Retail
Testing Challenges in POS
- Long Test Cycles
- Multiple Build Releases
- Multiple Configuration Settings
- Manual Backend Verification & Missing Transaction Validation
- Customized feature Complexity
- Testing POS on multiple platforms
- Peripheral issues
- Security & Regulatory Compliance

Solution through Test Automation

Designing an Automation Framework

Challenges in POS Automation
- Interaction with Peripheral devices
- Custom UI Objects
- Dynamic UI
- Multiple Configuration and Interaction with other interfaces

Conclusion
Point of Sale (POS) Testing

Introduction

A POS (point of sale) is a computer which is connected to a receipt printer, cash drawer, credit/debit card reader and a bar code scanner etc. Retailers use an automated retail system where the store cash registers are linked to computer processing systems. Merchandise is ticketed with coloured bar code tags, which are read with wand readers at the checkout counter. The computer accumulates sales transaction information on magnetic tape for daily input into the computer memory bank or storage system. It is input into the sales journal, which is rolled up into the stock ledger.

Testing Challenges in POS

Testing a POS software package manually can pose many challenges

- Long Test Cycles
  The world of retail is dynamic; retailers have to be on their toes to meet the market demand and stay ahead in the competition. Manual testing leads to long regression test cycles which prolongs the time to hit the market.

- Multiple Build Releases
  Rapid technological advancements necessitate frequent hardware and software upgrades which require more testing effort hence slowing down the QA build certification process.

Testing Point-Of-Sale systems can be challenging for the QA team for various reasons. This paper discusses the testing challenges surrounding POS systems and how a comprehensive test strategy can be developed to help retailers in achieving their quality goals.

Multiple Configuration Settings

Testing a POS application with different settings and configurations is a cumbersome task. Test cases pertaining to user profiles and access levels should be designed covering each and every scenario (positive or negative) in detail. Therefore, significant budget and effort needs to be put in testing of such applications to prevent any major issues at the customer end.

Manual Backend Verification & Missing Transaction Validation

The POS transaction verification is incomplete without validating the data at the backend i.e. in the database and in the Journal application. Testers need to validate that the data correctly gets stored at the backend and additionally there should not be any missing transactions. This verification requires time and is prone to human error.

Customized feature Complexity

Integration of POS System involves numerous interconnected systems and third party elements which poses testing challenges to the QA.

Testing POS on multiple platforms

As most of the retailers are now moving to mobile POS, effort of testing increases as the system has to be tested on desktop and mobile-OS combinations.

Peripheral issues

The peripheral issues may be related to devices which are connected to a POS like barcode scanners, scales, printers, towers and cash drawers.

Security & Regulatory Compliance

As customer’s sensitive data is being processed during the transactions, it’s critical to test POS system as per the regulatory compliances.
Solution through Test Automation

To ensure quality of POS software, proper testing of the application is crucial. Talking about software testing, automation testing has gained a lot of mileage over the last decade. Automation Testing is nothing but a software development process that lends itself to test software. Automation testing practices if implemented efficiently can yield effective results in delivering high quality software while keeping the cost at bay. Studies have proved that, companies that increase the proportion of automated testing have a decisive advantage over their competitors. To save manual testing time, a sound test automation strategy is paramount and an automation framework can be built. It is proven that automation testing has an edge over manual testing because it provides enhanced test coverage, saves testing time and cost, gives objective testing evidence in the form of customized reports, easy defect tracking for faster troubleshooting.

While most of the POS testing challenges can be overcome by automation, security testing can be a tricky area to deal with. Care must be taken to adopt PCI-compliant, tamper-proof infrastructure at all POS terminals to protect cardholder data and identity.

Let's now see how an effective test automation solution can be developed for POS.

Designing an Automation Framework

For automating the test cases of POS software, a test automation tool is required which can recognize the UI controls of the application. Selecting an appropriate automation test tool for a given application involves a step-by-step process. Without a proper process being followed, one might end up in either wastage of effort or selecting inappropriate tool(s) for the application under test (AUT). There are plenty of commercial and open source automation test tools available in the market. A proof-of-concept (PoC) exercise can be performed to select the best-suited tool for the POS application.

The framework for POS should be designed using best practices of automation testing coupled with the needs to test the POS systems effectively. The architecture and design of the automation framework decides its effectiveness. Just to name a few of the best practices – Page object model, Test data and UI elements externalization, implementing easy to interpret and informative Test Reports, integrating the framework with test case management tool and continuous integration tool etc.

An automation framework can be designed on the lines depicted in the diagram below:
The first & foremost is an Automation Tool – It can be Open Source or commercial tool that best suits the POS system under test. There can be various parameters to judge a tool e.g. it’s capability to identify the UI elements in the application, whether it supports automation on multiple platforms or not, what languages does it support, cost and post-sales support etc.

Second component is the Test Case Management Tool – As the name suggests, a test case management tool can be used to store the test cases. This tool should be integrated with the automation tool so that after test execution, the status of test cases like Pass/Fail/Not run etc should get updated in the test case management tool itself.

Thirdly, a Continuous Integration Tool – It’s an important component of an effective automation solution. This tool can enable the user to kick-start execution by just the click of a button and after the execution the reports can be delivered to the inbox of the stakeholders.

And lastly the Test Reports - An integral component of the framework, is easy to interpret and informative test report. The test reports helps you to understand useful information of the test cases executed like Pass/Fail%, Test Coverage, screenshots of the failure which is very critical to understand what went wrong during the execution.

In a nutshell, all of these components can be combined to build an integrated solution. In order to get the maximum ROI from automation testing, it’s utmost important to design the framework keeping the simplicity, usability, flexibility and scalability parameters in mind.
Conclusion

For complicated and business critical systems like POS, test strategy can be a combination of automation and manual testing. Also one should understand that testing of POS systems is different from other software and requires in-depth understanding of POS-specific challenges. To overcome such challenges and mitigate risks, the subject matter expert should carefully design the test strategy and approach in order to achieve their quality goals.