Liam Brew Professor Xiao

SSW 567 24 November 2020

Homework 12: Test Planning

Introduction

Description

The system under test (SUT) of this plan is a Banking ATM application. This application can perform several financial actions that are user-facing, such as letting customers check their balances, withdraw/deposit cash, and move funds among accounts. This application consists of a kiosk component which is loaded on the ATM itself and facilitates actions such as those that are cash-based in addition to the others mentioned. Additionally, there is also a mobile application component, which is accessible on smart phones and deals with only checks and balance transfers. Both components utilize touch screen technology, albeit with the kiosk component working with proprietary technology developed solely for the use of this application.

Test Strategy

This testing strategy focuses on the entire overall development cycle, viewing each step and stage from requirements elicitation through to deployment and maintenance to be important and warrant significant testing. This plan consists of a general plan and outline for the system's delivery and schedule as well as a framework from which to test it.

Development Approach

The development approach most suited to this project would be Agile. The goal of this approach is to facilitate improved project flexibility and give more leeway in terms of development planning. Due to the complexity of the system involved, with two separate components acting concurrently and in synchronization, testing and quality assurance is a very valid concern for this project. Agile helps facilitate an improved approach to this by enabling a quicker, more efficient reaction to bugs and defects. Additionally, Agile's distribution of testing throughout the entire development process will help prevent the emergence of major defects late in the development term and provide a more seamless and integrated approach to quality assurance.

Document Reference

Several documents should be referenced to assist with the development of this system. Firstly, there are the financial institution's operating procedures and documentation on pre-existing infrastructure such as databases and account management. These documents will help interface this new system and allow it to fit well with the tools already in use by the bank. In addition to this, there are also the technical specifications of both the kiosk and smartphones to consider. This will directly impact the success of the system, with development heavily depending on data regarding, for example, a phone's operating system. Finally, regulatory information that is pertinent to this system and its use cases should also be considered. One may potentially find there minimum technical standards which a system must adhere to for it to be acceptable by the wider financial community.

Testing Scope

Testing Targets

This system will be extensively tested according to its use cases and the requirements that it must adhere to. As one can imagine, the tests will primarily cover functionality and user-facing features as the system will feature significant direct usage. These features will primarily revolve around the core functionality of the application, such as its ability to withdraw and deposit money, accommodate checks, and perform balance transfers. Each of these features will be analyzed on several metrics such as their security, performance, accessibility, and maintainability. These metrics will be compared to what is expected of the target deployment environment as well as the baseline values described in the project's requirements documentation.

Ignored Testing

Due to time and resource constraints, not all aspects of the system will be prioritized for testing. Aspects that do not greatly facilitate the achievement of the system's core objectives will not achieve much in the way of analysis. This is to ensure that the major features receive as much testing as possible to ensure that the system will complete its core functionality as required.

Testing Criteria

The following criteria are ordered according to their importance, with the first one being most critical .

1. **Core Functionality**: these are the features explicitly mentioned in the project's requirements documentation, and include the withdrawal, deposit, transfer, balance, and check actions. As these features have a high and direct impact on the system's success, they are to be given top priority when it comes to testing and development. As these features will be used the most on the system, performance enhancements and optimizations will also be tested and conducted as well. These actions will be monitored individually to ensure the smooth and efficient conduction, both from the point of view of the user and the system itself.
2. **Accessibility**: accessibility is an important criterion of this system's success. As this system is very user-facing, it is important to ensure that many users can successfully operate and utilize it despite any difficulties that they may endure. When compared to its core functionality, the accessibility of a system has a more moderate impact as it does not directly satisfy the project requirements. However, accessibility does play a large role in ensuring that the system is attractive enough to its users to warrant continued use. Accessibility testing should take place when there is the potential for users to have trouble in performing an application action. It should be rectified by strict adherence to accepted accessibility guidelines.
3. **Performance Cost**: while the performance cost of an aspect of the application in terms of resources and time should be considered, this criterion should have the lowest impact on the plan and be tested for only in the absence of more pressing matters. It can be advantageous to examine the performance of the system to help detect if are any potentially run-away or other unoptimized processes. This can indicate other potential failures that are hidden in these processes, which may be revealed under more detailed inspection. Additionally, a higher performing system indirectly benefits the previously stated criteria as well.

Testing Approach

Key Factors

The most important factors pertaining to this SUT are its infrastructure integration, availability, and development plan.

- **Infrastructure Integration**: the SUT must be able to integrate with the existing technology used by the financial institution. For example, the SUT must be able to access the account database to query balances and utilize encryption systems to protect user login information. The SUT should interface with pre-existing corporate infrastructure as much as possible to reduce cost and improve reliability.
- **Availability**: this system will be used by many users simultaneously in different locations and from different types of devices. This coupled with its importance on day-to-day life necessitates significant resources being dedicated to ensuring a constant online and accessible status.
- **Development Plan**: the development plan directly impacts how the testing of the SUT will proceed. Not only does the plan deal with testing responsibilities and objectives, but it also carries other influences such as the order in which features will be developed, implementation plans and (perhaps most importantly) how the SUT satisfies user requirements.

Key Risks

Major risks pertaining to the SUT are its security and legal liabilities.

- **Security**: due to the nature of this system and the data it is responsible for, the upmost security precautions must be taken to ensure that user's accounts are protected from malicious manipulation. Therefore, the entire system should be designed and tested with security in mind.
- **Legal Liabilities**: data leaked or modified as the result of either accidental or intentional actions may subject the company to lawsuits and fines. This is especially so when considering the vast regulation that financial institutions must abide by. The threat of legal action should serve as further motivation for extensive testing.

Success Criteria

The success criteria for the SUT have been defined as the following:

- Each core feature specified by the requirements documentation has a success rate of greater than ninety-nine percent
    - On failure, these features display both a client-facing error message and internal logging and **preserve data integrity**
- The application must be sufficiently accessible to at least ninety five percent of users, determined through extensive user experience testing
- Core feature transactions are completed with a maximum latency time of fifteen seconds on at least ninety nine percent of occasions
- All user data must be up-to-date, and changes reflected live

Contingency Plans

The following contingency plans for the SUT have been identified:

- Continuous testing and integration to immediately identify and resolve defects

- Allocate funds for overtime hours and potential contracting of additional developers to ensure timely completion
- Build additional time into the project schedule to serve as a buffer
- Coordinate with other teams such as those working on hardware and the business aspect of the project

Item Pass/Fail Criteria

The following pass/fail criteria have been determined for the items of the SUT:

- Successful queries for the most-recent relevant data
- Bidirectional confirmation of data modification between the database and the application itself
- Acceptable run and latency times as previously defined
- Proper error catching and handling; no infinite loops or system crashes
- Adherence to all item-specific regulation

Entry/Exit Criteria

The following general entry/exit criteria have been determined for the SUT:

- Entry
    - All relevant code completed development
    - Requirements for relevant code defined
    - Allocated objectives and resources to specific testing session
- Exit
    - Sufficient testing coverage as agreed upon by the development team
        - All use cases tested and satisfied
    - Sufficient tests ran and passed as agreed upon by the development team
    - Acceptable performance as agreed upon by the development team
    - Completed peer-review by other developers who did not contribute to the code under test

Testing Criteria and Checkpoints

Major testing checkpoint should occur based on the calculated reliability of the SUT. For example, these can be at twenty-five, fifty, and seventy five percent. Once above ninety nine percent reliability, the testing criteria is satisfied, and testing is complete.

Test Deliverables

Important test deliverables should be:

- **Documentation**: documentation details the testing process, most notably what defects the team found and how they were fixed
- **Logs**: logs are the “raw” results of testing, such as outputs. They should be present to show a before and after view of system data and the results of testing

Budget

The testing budget should be significant. As previously mentioned, the importance and risk of this project are significant, with the potential liabilities being great. This necessitates a large amount of resources being dedicated to the testing process to ensure that it is extensive enough. Due to the scale

of this program and the lucrativeness of the financial technology market, no cost should be considered too great as this system can recoup it through successful usage and deployment. The testing stage should therefore not be constrained by resources in any way, whether they be time or material based.

Testing Methods, Tools and Automation

The following testing methods and their associated tools will be used:

- **Integration**: a crucially important part of testing, this method ensures that the components of the system will work well both with each other as well as with other systems. To accomplish this, build pipelines can be established on this project's version control repository to help automate the testing process and ensure continuous testing and integration.
- **Configuration Management**: in addition to the standard configuration management practices, for the kiosk component a local source of all dependencies for the current version of the software will be maintained. This is to ensure the existence of a stable build on-site to serve as a potential backup should the need arise. This is crucial as the kiosk component of the SUT directly handles cash and should therefore have a redundancy available should the standard methods of configuration management fail.
- **Unit**: developers will incorporate unit testing into their code as they write it. Basically, for a user story or sprint ticket to be considered complete, it must contain functional unit test cases related to the code developed. These tests will make use of a relevant framework (such as JUnit or PyUnit) and will feed into the project's build pipeline.
- **Requirement and Design Reviews**: the members of the development team responsible for requirements elicitation and system design will be consulted heavily during this stage of the testing process. As these members are experts in their respective fields and were responsible for this phase of the project, their input will be most useful in helping test the SUT. Additionally, this stage will feature further stakeholder involvement to ensure that they are satisfied with the developed project's solutions to their needs.
- **Usability**: usability testing will feature both stakeholder involvement as well as the inclusion of trial users. These users will interact with the system through their completion of the desired features and will relay their thoughts and experiences regarding the SUT's usability to the development team.
- **Performance and Reliability**: the system will be stress and load tested to ensure acceptable standards and reliability are maintained even under adverse conditions. These adverse conditions should include many scenarios ranging from increased load on the institution's servers to purposefully weathered hardware, simulating experiences that may occur in the actual deployment environment. Due to the importance of the SUT's functions, a very high success rate should be the bare minimum for approval.
- **Alpha/Beta Testing**: limited alpha and beta testing should occur in controlled environments. These tests should be closely observed and reported, with the collected data serving as a valuable insight into the potential real-world operating conditions of the SUT. This may also be a good time to reassess the system's development to see if there are any now-revealed features that are lacking or included features that are a detriment.

Testing Progress

The project's Agile development structure will play a role in measuring the progress of its testing. During each sprint's retrospective meeting, time will be dedicated to updating the team on the status of the SUT's testing, with special attention given to any major issues or recurring trends noticed by the testers. Also, at this time, testing data can be tracked like how the sprint's progress is updated in a burn down chart. This approach will help ensure continuous testing documentation that is current and applicable.

Shipping Decision

The SUT is ready to ship once it has attained the agreed upon acceptable values previously mentioned in this plan. No defects should be outstanding, and sufficient resources are queued on-standby to ensure an immediate attack to zero-day threats should the reveal themselves. The shipping decision itself should be settled upon once the testing stage has begun based on initial results.

Schedule

The project's tentative testing schedule alongside its development schedule is detailed below:

1 Jan – 7 Jan	Requirements Review	Review the project's requirements and settle on the core goals of the system
7 Jan – 14 Jan	Development Planning	Plan the initial project development and prioritize the most important features
7 Jan – 14 Jan	Test Planning	Using the information from Requirements Review, create a list of testing criteria to demonstrate system acceptability.
14 Jan – 15 Mar	Kiosk Development	Develop the kiosk and its related functionality requested by the client
13 Feb – 15 Mar	Kiosk Test Planning	Plan performance, reliability, usability, and security testing of the kiosk system based on work completed thus-far.
15 Mar – 14 May	Application Development	Develop the mobile application and its related functionality requested by the client
15 Mar – 29 Apr	Kiosk Testing	Execute the previously created kiosk test plan. Communicate feedback with developers and management.
29 Apr – 14 May	Application Test Planning	Plan performance, reliability, usability, and security testing of the application based on work completed thus-far.
14 May – 28 Jun	Application Testing	Execute the previously created application test plan. Communicate feedback with developers and management.
14 May – 1 Jun	Integration Test Planning	Plan integration testing of the combined system as well as it's interfacing with existing corporate infrastructure
1 Jun – 21 Jun	Integration Testing	Execute the previously created integration test plan. Communicate feedback with developers and management.
1 Jun – 21 Jun	Alpha Test Planning	Plan the limited alpha test period. This will serve as a trial run to gain operating environment experience with the system.
21 Jun – 15 Jul	Alpha Testing	Execute the previously created alpha test plan. Communicate feedback with developers and management.
15 Jul – 1 Aug	Deployment Planning	Prepare for the SUT's final deployment.
1 Aug – 1 Sep	Deployment	Execute the previously created deployment plan. Have sufficient resources on standby to quickly resolve any issues, ensuring a working system by the September 1st deadline.

Approval

The most important role for approval would be the project stakeholder, as they have the ultimate say over development decisions. Requirements personnel will assist in translating this test plan to the stakeholders. Next, project management must approve this plan in terms of it fitting within the larger development plan of the project and the resources that it requests. Any dedicated testing experts on the team with sufficient experience to classify them as such should also approve this plan to ensure that it is technically competent. Finally, the approval of the general body development team, though not required, would be a good indication of this plan's quality.