<u>IT-632-Software Engineering</u>

Cafeteria Management System (Cashless Canteen)

System Test Plan 1.2

Team-2

Instructor – Professor Asim Banerjee

10 October, 2013

DA-IICT, Gandhinagar

Overview

The test plan is to refer to once the coding on small individual modules of the project is done, so that it is easy to integrate the system with least bugs later on. This is to assist in coordinating software and test versions within configuration management.

Target Audience

Mentors
Developers / Coders
Testing Team

Document Revision History

Version	Primary Author(s)	Description	Reviewer(s)	Date
1.0	Team #2	First Draft	Aakash Thakkar, Utkarsh Patel	16 September, 2013
1.1	Aakash Thakkar, Utkarsh Patel	Changes updated after review	Jay Parikh	10 October, 2013
1.2	Jay Parikh	Final Draft	Jay Parikh	10 November, 2013

Contents

1. References

The following documents are needed to proceed with the development of an efficient test plan for the product:

- Project Plan v1.0 - To complete the product development in timely and cost effective manner.
- Software Requirements Specification v1.0 - To clearly comprehend what features are needed by the client/user of the product.

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements gathering process used to elicit requirements from the product stakeholders. It will serve as a guide for the developers on one hand and a software validation document for the prospective client on the other. It discusses the functional, non-functional, performance and development requirements and deliverables. It is an organized document with an aim to assist in designing, coding and testing of the software and reduce unnecessary development effort by clearly enlisting all the system requirements specification. It'll also help the development team visualize the scope of the project clearly.

1.2 Scope

The overall aim of the project is to simplify the cafeteria transactions by making them automated (computerized), cashless and quick. The average time of the transactions is to be reduced comprehensively. A web portal is to be developed with the help of which the cafeteria transactions would be carried out without any cash involved in it. The website would be accessible to the whole DA-IICT community as it would be hosted in the DA-IICT intranet server itself. One of the goal of the project is to eliminate the coupon system that is prevailing in the current scenario.

1.3 Stakeholders

The stakeholders would include:

- Customers
- Admin Staff
- Canteen Owners

1.4 Deliverables

The deliverables will include:

- Project Proposal
- Feasibility Report
- Project Plan

- Software Life Cycle Development Model
- Software Requirement Specifications
- User Manual
- Requirements Traceability Matrix
- System Design Documents
- System Test Plan
- Test Cases
- Test Reports
- User Interface Design Document
- Quality Assurance Plan
- Tools and Libraries
- Risk Monitoring, Management & Mitigation Plan
- Software Configuration Management Plan
- Coding Conventions
- Documentation Standards
- Cost Estimation Plan
- Deployment Plan
- Termination Analysis
- Product – Website

2. Test items

Within the scope of the project, we intend to test the following:

- **RFID-** we need to check whether the RFID is reading the ID card correctly.

- **Web portal**- A complete check of the portal is needed starting from login to logout at the end.

- **Validation -** Check that both the username and password is validated when a user logs in. some action is to be taken if they are left blank. The same applies in case of registration.

- **Transaction -** Check that the transaction is being done in less time and with accuracy. Money transfer from one customer's account to the other customer's account should be secure and fast.
  - In case of order cancellation, it is to be checked that the amount is again credited to the customer's account.
  - In case of card loss, the card should be immediately blocked to avoid theft.

- **User interface -** We need to check that the user interface is simple and navigable as the canteen operators are not computer engineers. They need not to search anything they required. Everything should be found easily.

- **Access:** It is to be checked that the access control is implemented successfully. Only limited access is to be provided to the customer, admin (cashier) and canteen operators according to their necessity.

3. Software Risk Issues

Following are the various risks associated with the website:

- System freeze at server/client side.
- Card Reader not responding due to damage/malfunction.
- Safety of monetary data.
- Users finding faults, leading to abuse of the system.
- Ability of the Canteen owners to understand the system easily.
- Half-transaction (If the money gets debited from one account but doesn't credit into another).
- Poorly documented modules and change requests.

4. Features to be tested

This is a listing of what is to be tested from the 'Users' viewpoint of what the system does. This is not a technical description of the software, but a user's view of the functions. As per the requirements captured by the SRS document the following modules would be tested in accordance with the test schedule.

- Login Module
- RFID scan module
- Registration Module
- Order Creation and cancellation module
- Transactions Module
- Update account module
- Notification System Module
- Reviews module

5. Features not to be tested

There are various features that are already provided by the JAVA framework itself and we will not be testing them. These features include:

- Security
- Platform Independency
- Multi-user Interface
- Web Browser's ability to execute scripts like JavaScript, CSS, HTML

6. Approach

- The choice of test approaches and strategies is one of the most powerful factor in the success of the test effort and the accuracy of the test plans and estimates. This factor is under control of the testers and test leaders.
- We will be using the bottom up approach for testing our entire project.
- We will start with unit testing in which every component's functioning will be tested (this is to be done concurrently).
- After checking all different components we will integrate all these components, as we progress, Integrity of different modules will be checked on the basis of some assumed input.
- Then we perform System testing, i.e. once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.
- Now we do Regression testing, which ensures that a change, such as a bug fix did not result in another fault being uncovered in the application.
- At each step of integration, modules will be tested.

7. Test Strategy

7.1 Types of testing

7.1.1 Unit Testing

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

Limitations of Unit Testing

Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing.

There is a limit to the number of scenarios and test data that the developer can use to verify the source code. So after he has exhausted all options there is no choice but to stop unit testing and merge the code segment with other units.

7.1.2 Integration Testing

The purpose of integration testing is to confirm that the interfaces between module works correctly i.e. the modules work well when they are combined together to build up the system. Modules would be subject to testing after being integrated with other modules to make sure that integration has not affected the individual functionality of the module. Modules will be added one by one to the already tested modules and then tested again.

7.1.3 Validation Testing

Verification and validation testing are two important tests which are carried out on a software before it has been handed over to the customer. The aim of both verification and validation is to ensure that the product is made according to the requirements of the client, and does indeed fulfill the intended purpose. While verification is a quality control process, the quality assurance process carried out before the software is ready for release is known as validation testing.

7.1.4 System Level Testing

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. This type of testing is performed by a specialized testing team.

System testing is so important because of the following reasons:

- System Testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

- The application is tested thoroughly to verify that it meets the functional and technical specifications.

- The application is tested in an environment which is very close to the production environment where the application will be deployed.

- System Testing enables us to test, verify and validate both the business requirements as well as the applications architecture.

7.1.5 Regression Testing

Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change. To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing. The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

Regression testing is so important because of the following reasons:

- Minimize the gaps in testing when an application with changes made has to be tested.

- Testing the new changes to verify that the change made did not affect any other area of the application.

- Mitigates Risks when regression testing is performed on the application.

- Test coverage is increased without compromising timelines. Increase speed to market the product.

7.1.6 Acceptance Testing

This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements. The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated. Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application.

By performing acceptance tests on an application the testing team will deduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

7.1.7 Performance Testing
It is mostly used to identify any bottlenecks or performance issues rather than finding the bugs in software.

Performance testing is considered as one of the important and mandatory testing type in terms of following aspects:

- Speed (i.e. Response Time, data rendering and accessing)

- Capacity

- Stability

- Scalability

7.1.8 Recovery Testing

Recovery testing will be done to test the system against crashes, system failure and other similar problems. Recovery will in this case be done with the help of the institute's system administrator as the entire database can be recovered through the administrator's help.

7.1.9 Stress Testing

This testing type includes the testing of Software behavior under abnormal conditions. Taking away the resources, applying load beyond the actual load limit is Stress testing.

The main intent is to test the Software by applying the load to the system and taking over the resources used by the Software to identify the breaking point.

This testing can be performed by testing different scenarios such as:

- Shutdown or restart of Network ports randomly.

- Turning the database on or off.

- Running different processes that consume resources such as CPU, Memory, server etc.

8. Item Pass/Fail Criteria

An item will pass or fail based on the testing phase result.
The pass criterion is:

- The output of the feature is exactly the same as the expected output.
- Execution occurs with no errors.
- Updating and execution happens in an optimal amount of time, which is small for the user experience to go bad.

9. Test deliverables

The following are the test deliverables:

- Test Plan Document
- Test Cases
- Error logs and execution logs
- Problem reports and corrective actions

As we proceed with the coding phase of the project, using the bottom up approach, we will do parallel test checks as per the test plan document.

10. Entry Criteria

It ensures that the proper environment is in place to support the entire system test process. Item must meet in entry criteria:

- All test hardware platforms must have been successfully installed, configured and functioning properly.

- All standard software tools including testing tools must have been successfully installed and functioning properly.

- All documentation and design of the architecture must be made available.

- All personnel involved in the system test effort must be trained in tools to be used during testing process.

- Proper test data is available.

11. Exit Criteria

It ensures that the project application has been satisfactorily completed before exiting the system test stage and clarifying the application as complete. Items must be met:

- Application must provide the required services.
- Ensure all application documentation has been completed and is up to date.
- 100% of all Priority 1 and priority 2 bugs must be resolved.

12. Software and hardware needs

12.1 Hardware

The minimum hardware requirements for the testing are:
- Computing device with intranet access
- Server
- RFID Card Reader and RFID Card/Tag
- Our product has a client-server application and the client side will have to be
- Connected to the server side (DA-IICT's server) using Intranet.

12.2 Software

Software	For
Apache Struts	Open-source web application framework for developing Java EE web applications.
Apache Tomcat	"pure Java" HTTP web server environment for Java code to run in
SMTP Server	Sending an automated emails.
Mysql	Database Management
JDK 1.5+	Its primary components a collection of programming tools for building applications in Java.

13. Responsibilities

Name	Role	Responsibility
Ashish	Test Leader	System testing System test plan Test case design Unit testing
Dipen	Test Planner	Test Case Report System Test plan Unit testing
Ankush	Tester	System testing Test Case Report Unit testing
Utkarsh	Tester	Integration testing Unit testing

14. Schedule

Following is the list of tests scheduled to be performed on the developed website:

1. Unit testing: This would be done by all the coders on specific unit that they develop and hence will be done simultaneously and independently after each unit is coded.
2. Integration testing: This will be done after integrating all the components.
3. System Testing: This will be performed after integration testing. The entire system will be configured in a controlled environment and testing would be done against test cases developed to simulate real life scenarios.

15. Planning Risks

The overall risks to the project are as under:

- Lack of personnel resources when testing is to begin.
- Late delivery of the software.
- Changes to the original requirements or designs.

Other possible risks can be

1) Team member/s not able to deliver on time due to unforeseen reasons, then following consequences may imply:

- Other members might have to deliver and work more to fill in for the non performing members.
- Team co-ordination can become imbalanced.
- Quality of the product can suffer, so can delivery in time.

2) Requirement definition will be complete by 15th September, 2013, but if the requirements change after that, any of the following may be the consequences:

- The scope of the plan may change.
- Cost of the project may change and hence cost reviewing and approving will be needed.
- The number of tests performed might be reduced or number of allowable defects might be increased.
- Test schedule and development schedule may be shifted accordingly.

● ● ●