

Test Plan

Project Name: Commerce Bank Project

Group: Team 3

Focus: Test Plan

<u>Version</u>	<u>Date</u>	<u>Author</u>	<u>Change Description</u>
1	04/10/06	Lucy Kull, Caleb Hillhouse, Nick King	Draft 1

TABLE OF CONTENTS

IDENTIFIER

3

INTRODUCTION

3

ITEMS AND FEATURES TESTED

3

TEST SPECIFICATIONS

3

TEST REPORTS

3

PROJECT PLAN

3

QUALITY ASSURANCE PLAN

4

ITEMS AND FEATURES TESTED

4

APPROACH

6

STRATEGY

6

TECHNIQUES/TOOLS

6

TEST GOALS

6

TEST DELIVERABLES

6

TESTING TASKS

7

SCHEDULE

7

REQUIRED RESOURCES

7

IDENTIFIER

Team 3's Commerce Bank Project Testing Plan

INTRODUCTION

This exercise essentially discovers faults and failures in the software. Faults lead to failures and testing is necessary to eliminate these errors in the software development process. The following document outlines the testing procedures for the Commerce Bank Capstone software project.

Items and Features Tested

The features tested include the individual application logic units (database abstract methods, webpage methods, chart drawing javascript) and the integrated webpage system that implements these functions to interact with the user and render data.

Test Specifications

This section highlights the different levels of testing performed during the software development process. It includes integrations tests performed by Nick King and [---], and system tests performed by Nick King and [---] to verify that the system fulfills the requirements and performs as expected.

Test Reports

Test reports offer feedback about the testing exercise and are informational to project stakeholders. The reports in this testing exercise track incidents, defects, and changes while testing the system.

Incident reports track descriptions of incidents, the originator, severity and priority, steps required to produce the defect, the responder, current status, cause, resolution and date the incident was addressed. The document will also track in what phase of the development process the incident was created or detected.

Defect reports reference the unique identifier in the incident report, list affected items, the owner, required changes, status, phases where the defect was introduced and detected, and general comments or notes.

Change reports summarize the total number of unit tests, integration tests, and system tests, the items and features tested, comprehensiveness of the assessment, the total number of incidents and defects, and unresolved incidents.

Project Plan

The project plan reconciles the time schedule and release of candidates with the unit, integration, and system testing.

Individual units are written as part of a web page module; testing is performed as the web page comes together and controls for testing the data retrieval and display are implemented. Revisions are made here when defects or oversight is found, so that these units that may be used in future modules are already sufficiently tested. Regression testing will be done consistently throughout the lifetime of the project. This leaves only system testing, where we test the transfer and display of a user's data across multiple pages of the webapp. System tests will be completed when the final module is implemented to the project to ensure that all desired features function as intended.

Quality Assurance Plan

The system will be examined by the team in the week before the presentation to the bank to conclude that any remaining bugs have been resolved and that the finished product addresses the concerns contained in the requirements document.

ITEMS AND FEATURES TESTED

Login Page

User verification and login testing takes place in the login page integration testing and system testing. The tester will test login functions and verification functions that contact the database, as well as UI functionality on the login page, and will retest them after features are added to make sure they still function correctly.

Test items:

- Verify Unique User

Reactive Site Design

Reactive layout testing takes place in individual web page integration testing and system testing. The tester will test the template layout on resizing and functions that display data, and will retest them after features are added to make sure they still function correctly.

Test items:

- Site is Useable in Mobile-view
- Site is Useable in Desktop-view

Goals Page

Goals page testing takes place in the goals page integration testing and system testing. The tester will test functions that display goals for the user, the ability to add a goal, and restrictions on input for adding or editing a goal, and will retest them after features are added to make sure they still function correctly.

Test items:

- Adding Goals
- Editing Goals
- Completing Goals
- Make Goal a favorite Goal

Budgets Page

Budgets page testing takes place in the login page integration testing and system testing. The tester will test functions that display budget data and the ability to add a budget for a user, and will retest them after features are added to make sure they still function correctly.

Test items:

- Adding budgets
- Make budget a favorite budget

Summary Page

Summary page testing takes place in the summary page integration testing and system testing. The tester will test functions that display favorite goals and budgets data and the methods that draw charts and present data, and will retest them after features are added to make sure they still function correctly.

Test items:

- View favorite Goals
- View favorite Budgets
- View spending chart

Notifications

Notifications testing takes place in the static database procedure testing and system testing. The tester will test procedures that send notifications to a user on database updates to goals or budgets, and will retest them after features are added to make sure they still function correctly.

Test items:

- Set notification frequency
- Prompt notification
- Send notification

Calendar

Calendar testing takes place in the transactions/summary page integration testing and system testing. The tester will test functions that display a calendar and allows a user to choose a date to show relevant data, and will retest them after features are added to make sure they still function correctly.

Test items:

- View current calendar
- Select and import date

Transactions Page

Transaction page testing takes place in the transactions page integration testing and system testing. The tester will test functions that use calendar entry to display data, and test asp control structures that serve to retrieve and modify specific data, and will retest them after features are added to make sure they still function correctly.

Test items:

- Display transactions by date, name, or category
- Search transactions
- Recategorize transactions

APPROACH

Strategy

The testing approach includes team members role-playing by acting as users of the website. The activities they carry out will be similar to activities carried out by users. These include viewing their spending summary, adding goals and budgets, and viewing their transaction data. Team members will also create unexpected scenarios such as invalid input, to test how the system will respond to such scenarios.

System testing will be performed by the group to measure how the system fulfills the requirements. Team members will be required to provide input on the system's functionality to determine if it measures up to the client's expectations.

Group members will also ask other students to user test the system as a whole as the project comes to a close and the system is under review for minor bug fixes in order to reveal unexpected errors from user input that we may not have anticipated.

Techniques/Tools

Developers will perform unit tests to check code functionality and execution of the software, and perform white-box testing. They will also perform necessary NUnit and regression tests as they develop the code.

Testers will develop test cases and perform black-box testing. Team members will also go through the code to find bugs and errors that the developers might not detect.

Test Goals

The goals of this testing exercise include finding various user situations and testing against these situations in order to discover any errors that may cause failures. The ultimate goal is to ensure that the team delivers an error-free functioning system.

TEST DELIVERABLES

- A test plan that delineates the testing phase of the development process
- A test specification document detailing steps in the testing procedures
- Test reports documenting incidents, defects and changes

TESTING TASKS

Testing tasks for integration and system testing are outlined in the test specification document.

SCHEDULE

Testing will occur at each phase of the development process. Test cases are outlined in the test specification document for the iterations.

REQUIRED RESOURCES

Human resources: Lucy Kull
Nick King
David Jones
Cameron L'Ecyuer
Candice Johnson
Caleb Hillhouse
Eilliot Kieffer

Material: C# using ASP.NET for development, Microsoft SQL server for deployment, Google Docs (.doc) and Adobe Acrobat (.pdf) for documentation