Test Approach: Iteration 1

Woodgrove Bank Reach Portal: Online Bill Payment System

## Document Status

|  |  |
| --- | --- |
| Title | Test Approach: Iteration 1 |
| Author(s) | Tom Perham |
| Team | Reach Portal project |
| Version | v0.2 |
| Status | Draft |

## Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change reference |
| 17 March | T Perham | v0.1 | Initial draft for review/discussion |
| 14 May | T Perham | v0.2 | Changes based on feedback |
|  |  |  |  |

## Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version reviewed | Position | Date |
| A Orekhov | v0.2 | Test Manager | 15 May |
|  |  |  |  |

# Purpose of This Document

Test Approach captures the overall test approach for testing the system being developed. It includes the scope of what needs to be tested, the important milestones for test, exit criteria that are required to be met before the software can be released, and how the team will manage the bugs.

# Overview

Reach Portal project is an ambitious project for Woodgrove Bank to improve its presence in the online banking services. The online bill payment system being developed in this iteration has stringent testing requirements because it interacts with multiple external systems. Security testing is the most important part of the testing so that the customer and bank data is secured and protected.

The following user stories are planned to be developed in this iteration:

1. The user sees the bill payment option displayed prominently.
2. The user adds a payee.
3. The user deletes a payee.
4. The user adds a payment method.
5. The user deletes a payment method.
6. The user receives a bill and schedules a payment.
7. The user cancels a payment.

# Test Approach

Testers will write acceptance test cases for each of the preceding user stories and will record them in the Test Case work item in TFS. The following sections help the testers write identify and write the test cases.

## Scope of Testing

The next sections describe the types of testing the testing team will perform.

### Functional Testing

Functional testing will include the following:

* Identifying acceptance tests.
* Creating UI tests to make sure the UI is consistent across the bill payment system and the checking and savings account system.

### Integration Testing

Verify that the bill payment system integrates well with the following systems:

* Checking and savings account system
* Third-party payment gateway
* Bill processing system

### Security Testing

Security testing will include the following:

* Create a threat model (A document template is available in the folder ../Documents/Samples and Templates/Security)
* Security code reviews

### Load and Performance Testing

Run automated performance and load tests to meet or exceed the performance goals in the Release Criteria section.

### Stress Testing

Perform stress testing with 1,500 concurrent users.

## Release Criteria

The following release criteria must be met before the bill payment system can be deployed.

|  |  |
| --- | --- |
| **Release criteria** | **Threshold** |
| Code Coverage in Unit Testing | Minimum 70% |
| Bugs | No Severity 1 Bugs  No more than 3 Severity 2 Bugs |
| Performance | With 1000 concurrent users:   * Web page response time < 1.5 seconds * Bill payment confirmation time < 15 seconds |
| Automated Tests | At-least 80% of the tests should be automated |

## Test Configurations

The following configurations will be tested for the bill payment system.

|  |  |
| --- | --- |
| Server | Windows Server 2003 |
| Client | All versions of Internet Explorer 5.0 and higher  All versions of Firefox 1.0 and higher |

# Logistics

## Roles and Responsibilities

The following are the roles in the Reach Portal project team:

* Project Manager: Mark Hanson
* Developers: Rob Barker, Dave Barnett, David Jones.
* Testers: David Simpson, John Smith, Mark Harrington
* User Experience Lead: René Klčo

## Bug Management

The following sections describe the processes the Reach Portal project team will use to manager bugs for the bill payment system.

### Procedure for Filing and Tracking Bugs

We will follow the following process to file and track the bugs in the TFS:

* All the bugs will be filed in TFS as a Bug work item.
* All the bugs will be linked to the User Story or the Task as appropriate.
* Bugs will be assigned Severity and Priority as suggested in the “Definition of Severity and Priority Levels” section.

### Definition of Severity and Priority Levels

The following definitions for Severity and Priority will be used for the bugs.

#### Severity

* **1 = Critical Failure**. Completely breaks a functionality or large set of functionality in a user story. Unusable. Significant security or legal risk.
* **2 = Major Impact / Functionality Broken**. Breaks major functionality, contributes to overall instability in this area, non-fatal assertions. For example, Statement Completion not active at all or memory leaks. Regression from prior release.
* **3 = Minor Impact / Functionality Impaired**. Breaks major functionality in a minor way or breaks minor functionality completely. For example, an item is missing from the list for statement completion.

#### Priority

* **1 =** Security bugs, crash/exception in mainline scenario, main line scenario is broken, significant features are unusable, data loss, hang/deadlock, regulatory requirement.
* **2 =** Workaround is too cumbersome, renders an area/feature untestable through automation, workaround too expensive for QA, risk/benefit is acceptable, automated functionality results in an error, performance regression of more than 10%.
* **3 =** No visible customer/scenario impact, high risk to gain “lower” customer impact, “fit and finish”, workaround is easy/obvious, documentable

### Triage Process

The bugs will be triaged in a meeting attended by the developers, testers, project manager, and the customer representative. The bugs approved for fixing will be added to the Iteration Backlog. The team will meet for triage every Wednesday and Friday morning at 10:00 A.M.

## Testing Tools

We will use Visual Studio Team Edition for Software Testers for the following:

* Performance testing
* Web testing
* Automation testing

We will use Visual Studio Team System for the following:

* Creating test cases
* Creating bugs

## Schedules and Milestones

The following is the list of important milestones for Test.

|  |  |
| --- | --- |
| **Test Activity** | **Date** |
| Create Test Plan draft and share with the team | 03/17/2007 |
| Finalize Test Plan | 03/19/2007 |
| Start writing automation tests | 04/30/2007 |
| All user stories dev complete | 05/21/2007 |
| Final integration test pass complete | 06/11/2007 |
| Test signoff | 06/18/2007 |