

# Acceptance Test Plan Document

REPortal Migration Team

April 10, 2007

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Document Structure . . . . .	4
1.2	References . . . . .	4
<b>2</b>	<b>Approach</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Objectives . . . . .	5
2.3	Structure . . . . .	5
2.4	Assumptions . . . . .	5
2.5	Exclusions . . . . .	5
<b>3</b>	<b>Criteria</b>	<b>6</b>
<b>4</b>	<b>Responsibilities</b>	<b>7</b>
4.1	Introduction . . . . .	7
4.2	Roles and Responsibilities . . . . .	7
4.3	Requirments . . . . .	7
4.4	Reporting . . . . .	7
<b>5</b>	<b>Test Cases</b>	<b>8</b>
5.1	Introduction . . . . .	8
5.2	User Interface . . . . .	8
5.2.1	Login page . . . . .	8
5.2.2	Registration page . . . . .	8
5.2.3	Main page . . . . .	8
5.3	Logger . . . . .	9
5.3.1	Initial: . . . . .	9
5.3.2	Actions: . . . . .	10
5.3.3	Consequences: . . . . .	10
5.4	Instrumentation . . . . .	10
5.4.1	Initiate Aspect Instrumentation . . . . .	10
5.4.2	Initiate Aspect Instrumentation During Runtime . . . . .	10

# 1 Introduction

This document outlines the Acceptance test plan for the REportal Migration Project. The goal of this project is to make the REportal system more robust than its current iteration, incorporating modern programming techniques such as a Service Oriented Architecture, and Aspect Oriented Programming. This new version of REportal will also be made more modular to allow for future projects to expand upon its current programming.

## 1.1 Document Structure

- Section 2 explains the approach of the acceptance test plan that we are using. It explains what is being tested in the acceptance test plan and what is not.
- Section 3 lists what requirements for the acceptance test plan to begin.
- Section 4 explains who has what responsibilities during the acceptance test.
- Section 5 is the test cases to be used during the acceptance test plan. The cases have an initial state REportal must be in, what actions must be done for the test, and the consequences of the actions.

## 1.2 References

1. Requirements Document

## **2 Approach**

### **2.1 Introduction**

This section describes the approach to the testing of reportal to ensure that it meets all requirements.

### **2.2 Objectives**

The Acceptance test will define how REportal is tested to ensure that it meets the requirements defined in the Requirements document by testing defined test cases.

### **2.3 Structure**

The tests are structured in a way so that each test checks to see if one requirement is met. This way we can ensure that all the requirements are met as listed in the Requirements Document. It is important that all tests pass and to record all results of testing.

### **2.4 Assumptions**

The Acceptance test assumes that all other tests are satisfactory. This test will cover the following.

1. The Functional Requirements as defined in the Requirements Document
2. Usability of the system

### **2.5 Exclusions**

The acceptance test will not cover the following because they will be covered by other tests.

1. The nonfunctional requirements defined in the Requirements Document
2. Integrity of the Source Code

### **3 Criteria**

The Acceptance test can begin after the following have been met.

1. All other tests are completed.
2. A proper environment to conduct the Acceptance Test is set up.
3. A copy of the latest version of the Requirements Document is available.
4. The latest version of REportal is available.
5. Consent from the Team Leader
6. Consent from the Client
7. Consent from the Project Leader

## **4 Responsibilities**

### **4.1 Introduction**

This section describes the responsibilities of all parties during the Acceptance test.

### **4.2 Roles and Responsibilities**

Acceptance Test Leader:

- Justin Wilcox

Acceptance Testers:

- Thomas Shortell
- Dan Cardillo
- Tim O'Neill
- Umut Akdag

Customer Representatives:

- William Mongan
- Spiros Mancoridis

### **4.3 Requirements**

All parties during the testing of the Acceptance Test should be familiar with the interface for REportal and basic understanding of how REportal works.

### **4.4 Reporting**

All tests done during the Acceptance Testing need to be recorded by the tester. Action will be taken reactively as problems arise during the testing phase.

## **5 Test Cases**

### **5.1 Introduction**

Each test case has a name, a state REportal should be in to initiate the case, what action(s) need to occur to perform the test, and what should happen after the actions are completed.

### **5.2 User Interface**

#### **5.2.1 Login page**

1. Initial state: User first views the login page. Both login and password fields are blank.
2. Actions: User enters a username and a password in the appropriate boxes. User then pressed the login button.
3. Consequences: If a valid username and password have been entered, the site will log the user in and take them to the main page. If there is a problem, they will be taken back to the login page where they will see an error message.

#### **5.2.2 Registration page**

1. Initial state: Clicking on the Sign Up button on the Login page brings the user to the Registration page. All fields are initially blank.
2. Actions: User enters appropriate information in each of the fields.
3. Consequences: If all boxes are correctly filled out, then the user will be sent an email with the pertinent login information. If there is a problem, they will receive an error message.

#### **5.2.3 Main page**

1. Initial state: Page loads, user views their uploaded project folders, including the default DEMO folder.
2. Actions: User uploads a new project.

3. Consequences: The page reloads and displays the new project folder, with files intact.
1. Initial state: Page loads, user views their uploaded project folders, including the default DEMO folder.
2. Actions: User expands a folder.
3. Consequences: All of the files and subfolders inside the appropriate project folder are now viewable. Subfolders display the + icon next to them, indicating that they can be expanded.
1. Initial state: Page loads, user views their uploaded project folders, including the default DEMO folder.
2. Actions: User double clicks on a project folder, or right-clicks and then selects OPEN from the menu.
3. Consequences: User is brought to the entity search page for the appropriate project.

### 5.3 Logger

- a Start up database
- b Logging service
- c Initiate service means start the service which needs to do the logging
- d Service reads in logging configuration
- e Service runs logging events

Consequences: Data logged by service should get reflected in database

#### 5.3.1 Initial:

- Start the database
- Start the logging server
- Start the service

### **5.3.2 Actions:**

Service registers with logging server

Logging server provides logging configuration to service

Service starts running and logging events on the logging server

Logging server processes the log entries and updates to database/xml file

### **5.3.3 Consequences:**

Database/XML should have the relevant logged info by the service

## **5.4 Instrumentation**

### **5.4.1 Initiate Aspect Instrumentation**

1. Initial: Source Code Uploaded
2. Actions: Enable Aspect Instrumentation for entire source code, run the code
3. Consequences: User should be able to view graphs provided by the Instrumentation

### **5.4.2 Initiate Aspect Instrumentation During Runtime**

1. Initial: Source code to be tested is running
2. Actions: Start the Instrumentation, Stop the Instrumentation after X time units
3. Consequences: User should be able to view graphs provided by the Instrumentation for the section of time selected