|  |
| --- |
| **Concordia University**  **Department of Computer Science**  **and Software Engineering** |

|  |
| --- |
| **F.S.T.S.**  ***Family Services Tracking System*** |

|  |
| --- |
| **Testing Plan, Cases & Reports** |

|  |
| --- |
| **SOEN 390**  **Software Development Project**  **Winter 2012** |

**JAM PACK**

Katrina Anderson

Cynthia Donato

Joshua Hum

Mikhael Levkovsky

Adrian Lloyd

Patrick Modafferi

**Table of Contents**

[**Test Plan 7**](#_Toc321862399)

[1. Introduction 8](#_Toc321862400)

[1.1 Purpose 8](#_Toc321862401)

[1.2 Scope 8](#_Toc321862402)

[1.3 Intended Audience 8](#_Toc321862403)

[1.4 Document Terminology and Acronyms 8](#_Toc321862404)

[1.5 References 8](#_Toc321862405)

[2. Evaluation Mission and Test Motivation 9](#_Toc321862406)

[2.1 Background 9](#_Toc321862407)

[2.2 Evaluation Mission 9](#_Toc321862408)

[2.3 Test Motivators 9](#_Toc321862409)

[3. Target Test Items 11](#_Toc321862410)

[3.1 Features to be Tested 11](#_Toc321862411)

[3.2 Features Not to be Tested 11](#_Toc321862412)

[4. Test Approach 11](#_Toc321862413)

[4.1 Initial Test-Idea Catalogs and Other Reference Sources 12](#_Toc321862414)

[4.2 Testing Techniques and Types 12](#_Toc321862415)

[5. Testing Workflow 21](#_Toc321862416)

[5.1 Test Phase Table 21](#_Toc321862417)

[5.2 Test Phase Gantt Chart 23](#_Toc321862418)

[5.3 Test Management Tools 24](#_Toc321862419)

[5.4 Test Slippage 24](#_Toc321862420)

[5.5 Defects 24](#_Toc321862421)

[**Test Cases 25**](#_Toc321862422)

[1. Acceptance Test Cases 26](#_Toc321862423)

[TC01.2 Test Create User 26](#_Toc321862424)

[TC02.2 Test Update User 26](#_Toc321862425)

[TC03.2 Test Delete User 27](#_Toc321862426)

[TC04.2 Test Set User to Administrator 27](#_Toc321862427)

[TC05.1 Test View Manage Users 28](#_Toc321862428)

[TC06.1 Test User Login 28](#_Toc321862429)

[TC06.2Test User Login – Omitted Password Field 29](#_Toc321862430)

[TC06.3 Test User Login – Invalid Password 29](#_Toc321862431)

[TC06.4 Test User Login – Invalid Username 30](#_Toc321862432)

[TC06.5 Test User Login – Omitted UsernameField 30](#_Toc321862433)

[TC07-1 Test View Check-in for all Clients coming to an event – Successful 31](#_Toc321862434)

[TC08.1 Test View All Clients Coming to an Event –Success Scenario 31](#_Toc321862435)

[TC09-1 Test Closing an Event – Successful 32](#_Toc321862436)

[TC10.1 Test Appointment Creation – Success Scenario 32](#_Toc321862437)

[TC11.1 Test Appointment Update – Success Scenario 33](#_Toc321862438)

[TC12.1 Test Appointment Cancellation – Success Scenario 33](#_Toc321862439)

[TC13.1 Test Appointment Validation – Capacity Exceeded 34](#_Toc321862440)

[TC14.1 Test View Summary of Clients with Appointments to Events 34](#_Toc321862441)

[TC16.1 Test Create Client File - Minimal Information 35](#_Toc321862442)

[TC16.2 Test Create Client File – All Information 35](#_Toc321862443)

[TC16.3 Test Create Client File - Missing Required Fields 35](#_Toc321862444)

[TC16.4 Test Creating Client File - Address Exists 36](#_Toc321862445)

[TC17.1Test Update Client File - Main Success Scenario 36](#_Toc321862446)

[TC18.1 Test Set Client File to Active 37](#_Toc321862447)

[TC21.1Test Add New Flag – Success Scenario 37](#_Toc321862448)

[TC21.2Test Add New Flag – Failure Scenario 1 38](#_Toc321862449)

[TC21.3Test Add New Flag – Failure Scenario 2 38](#_Toc321862450)

[TC22.1Test Delete Flag – Success Scenario 39](#_Toc321862451)

[TC23.1Test Update Flag – Success Scenario 39](#_Toc321862452)

[TC23.2Test Update Flag – Failure Scenario 40](#_Toc321862453)

[TC23.3Test Update Flag to Inactive – Success Scenario 40](#_Toc321862454)

[TC23.4 Test Update Flag to Active – Success Scenario 41](#_Toc321862455)

[TC24.1 Test Set Flag on New Client File – Success Scenario 41](#_Toc321862456)

[TC24.2 Test Set Flag on Existing Client File – Success Scenario 42](#_Toc321862457)

[TC25.1 Test Remove Flag on Existing Client File – Success Scenario 42](#_Toc321862458)

[TC26.1 Test Set Note Text on New Client File – Success Scenario 43](#_Toc321862459)

[TC26.2 Test Set Note Text on Existing Client File – Success Scenario 43](#_Toc321862460)

[TC27.1 Test Remove Note Text on Existing Client File – Success Scenario 44](#_Toc321862461)

[TC28.1 Test Update Note Text on Existing Client File – Success Scenario 44](#_Toc321862462)

[TC29.1 Test Client File – Dependent Association 45](#_Toc321862463)

[TC29.2 Test Create Dependents – All Information Fields 45](#_Toc321862464)

[TC29.3 Test Create Dependents - Missing Required Fields 46](#_Toc321862465)

[TC30.1 Test Dependent Deletion 46](#_Toc321862466)

[TC31.1 Test Dependent Update 47](#_Toc321862467)

[TC32.3 Test Postal Code/City Association – with Region Mapped Valid Postal Code 47](#_Toc321862468)

[TC32.4 Test Postal Code/City Association – with Invalid Postal Code 47](#_Toc321862469)

[TC33.4 Test Create Country Field - Failure Scenario 48](#_Toc321862470)

[TC33.5 Test Create Language Field - Failure Scenario 48](#_Toc321862471)

[TC34.3 Test Updating Client File Field – Success Scenario 49](#_Toc321862472)

[TC34.4 Test Updating Client File Field – Failure Scenario 50](#_Toc321862473)

[TC35.3 Test Deleting Entry in Country Field 51](#_Toc321862474)

[TC36.1 Test View Client File Foreign Key Mapping 51](#_Toc321862475)

[TC36.2 Test View Client File User Acceptance of General Layout 52](#_Toc321862476)

[TC37.2 Test Search Client Files by Client ID 52](#_Toc321862477)

[TC38.9 Test Search Client Files by Last Name 52](#_Toc321862478)

[TC38.10 Test Search Client Files by Last Name and Medicare Number 53](#_Toc321862479)

[TC39.1 Test Event Template Creation– No Appointment & No Repetition 53](#_Toc321862480)

[TC39.2 Test Event Template Creation – No Appointment & Repetition 54](#_Toc321862481)

[TC39.3Test Event Template Creation - Appointment & Repetition 54](#_Toc321862482)

[TC39.4 Test Event Template Creation 55](#_Toc321862483)

[TC39.5Test Event Template Creation– Mandatory Field Omitted 55](#_Toc321862484)

[TC40.1Test Event Template Deletion 56](#_Toc321862485)

[TC41.1Test Event Template Update 56](#_Toc321862486)

[TC41.2Test Event Template Update - Mandatory Field Omitted 57](#_Toc321862487)

[TC42.1Test Display of a List of Event Templates 57](#_Toc321862488)

[TC42.2 Test Display of Event Template Details 58](#_Toc321862489)

[TC43.5 Event Occurrence Creation– No Appointment 58](#_Toc321862490)

[TC43.6 Test Event Occurrence Creation–Appointment by Hour 59](#_Toc321862491)

[TC43.7 Test Event Occurrence Creation–Appointment by Day 59](#_Toc321862492)

[TC43.8 Test Event Occurrence Creation–Mandatory Field Omitted 60](#_Toc321862493)

[TC44.2 Test Event Occurrence Cancellation 60](#_Toc321862494)

[TC45.3 Test Event Occurrence Update 61](#_Toc321862495)

[TC45.4Test Event Occurrence Update - Mandatory Field Omitted 61](#_Toc321862496)

[TC46.1Test Display List of Active Event Occurrences 62](#_Toc321862497)

[TC47.2 Test Display of Event Occurrence Details 62](#_Toc321862498)

[TC48.1Test Generate Operational Reports – Success Scenario 63](#_Toc321862499)

[TC50.1Test Generate Statistical Reports – Success Scenario 63](#_Toc321862500)

[TC51.2 Test for Creating an Income Source 64](#_Toc321862501)

[TC52.2 Test for Deleting an Income Source 64](#_Toc321862502)

[TC53.2 Test for updating an income source 65](#_Toc321862503)

[TC54.2 Test for the viewing of the income sources 65](#_Toc321862504)

[TC55.1 Test Add Client to Waiting List – Success Scenario 66](#_Toc321862505)

[TC55.2 Test Remove Client from Waiting List – Success Scenario 66](#_Toc321862506)

[TCD01.2 Test for creation of correct times for time fields 67](#_Toc321862507)

[TCD20.1 Test Deleting an Event with No Associated Appointments – Successful 67](#_Toc321862508)

[TCD33.1 Test Create Appointment per Household Member – Successful 68](#_Toc321862509)

[TCD34.1 Test Set Event Inactive – Successful 68](#_Toc321862510)

[TCD34.2 Test Set Event Amount – Successful 69](#_Toc321862511)

[TCD35.1 Test Closing an Event Sets the Last Visited Date for All Attendee 70](#_Toc321862512)

[TCD56.1 Test Generate Yearly Report – Successful 71](#_Toc321862513)

[2. Volume Test Cases 71](#_Toc321862514)

[TCV-01 Volume Test Generate Yearly Report (Excel) 71](#_Toc321862515)

[TCV-02 Volume Test Generate Report (PDF) 72](#_Toc321862516)

[TCV-03 Volume Test Client Search 72](#_Toc321862517)

[3. Usability Test Case 73](#_Toc321862518)

[TCU-01 Usability Test Client File, Search, Events & Admin Panel 73](#_Toc321862519)

[**Test Results Report 75**](#_Toc321862520)

[1. Iteration 2 Test Results Report 76](#_Toc321862521)

[1.1 Introduction 76](#_Toc321862522)

[1.2 Acceptance Testing 76](#_Toc321862523)

[1.3 Unit Testing 78](#_Toc321862524)

[1.4 Integration Testing 81](#_Toc321862525)

[1.5 System Testing 81](#_Toc321862526)

[2. Iteration 3 Test Results Report 82](#_Toc321862527)

[2.1 Introduction 82](#_Toc321862528)

[2.2 Acceptance Testing 82](#_Toc321862529)

[2.3 Unit Testing 84](#_Toc321862530)

[2.4 Integration Testing 88](#_Toc321862531)

[2.5 System Testing 88](#_Toc321862532)

[3. Iteration 4 Test Results Report 90](#_Toc321862533)

[3.1 Introduction 90](#_Toc321862534)

[3.2 Acceptance Testing 90](#_Toc321862535)

[3.3 Unit Testing 93](#_Toc321862536)

[3.4 Integration Testing 98](#_Toc321862537)

[3.5 System Testing 98](#_Toc321862538)

[4. Iteration 5 Test Results Report 100](#_Toc321862539)

[4.1 Introduction 100](#_Toc321862540)

[4.2 Acceptance Testing 100](#_Toc321862541)

[4.3 Unit Testing 103](#_Toc321862542)

[4.4 Integration Testing 108](#_Toc321862543)

[4.5 System Testing 108](#_Toc321862544)

[4.6 UI Test Table & Results 110](#_Toc321862545)

[4.7 Defects 111](#_Toc321862546)

[5. Iteration 6 Test Results Report 115](#_Toc321862547)

[5.1 Introduction 115](#_Toc321862548)

[5.2 Acceptance Testing 115](#_Toc321862549)

[5.3 Unit Testing 118](#_Toc321862550)

[5.4 Integration Testing 123](#_Toc321862551)

[5.5 System Testing 123](#_Toc321862552)

[5.6 UI Test Table & Results 125](#_Toc321862553)

[5.7 Volume Test 126](#_Toc321862554)

[5.8 Usability Test Results. 132](#_Toc321862555)

[5.9 Defects 139](#_Toc321862556)

[**Appendix A References 145**](#_Toc321862557)

|  |
| --- |
| **F.S.T.S.** |

|  |
| --- |
| **Test Plan** |

# Test Plan

|  |
| --- |
| **Version 6.17** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev.** | **Description** | **Author(s)** |
| 2012-02-08 | 2.0 | Set up document | Katrina Anderson |
| 2012-02-10 | 2.1 | Testing workflow | Mikhail Levkovsky |
| 2012-02-11 | 2.2 | Testing workflow table | Mikhail Levkovsky |
| 2012-02-11 | 2.3 | Target Test Items | Josh Hum |
| 2012-02-11 | 2.4 | Testing Workflow Gant Charts | Adrian Lloyd |
| 2012-02-11 | 2.5 | Revised Testing Workflow | Adrian Lloyd |
| 2012-02-11 | 2.6 | Test Approach | Katrina Anderson |
| 2012-02-12 | 2.7 | Introduction, Evaluation Mission and Test Motivation | Patrick Modafferi |
| 2012-02-12 | 2.8 | Document formatting and revision | Katrina Anderson |
| 2012-02-23 | 3.9 | Updated Gantt Chart to include weekends. Didn’t generate properly. | Katrina Anderson |
| 2012-02-24 | 3.10 | Updated Integration & System Test Plans | Katrina Anderson |
| 2012-02-24 | 3.11 | Revised Entire Document | Katrina Anderson |
| 2012-03-10 | 4.12 | Revised Integration and System Testing | Katrina Anderson |
| 2012-03-12 | 4.13 | Revised scope and introduction | Patrick Modafferi |
| 2012-03-20 | 5.14 | Revised and added specs for regression and UI testing. | Katrina Anderson |
| 2012-04-08 | 6.15 | Added Stress Test | Patrick Modafferi |
| 2012-04-10 | 6.16 | Updated test plan testing approach (usability testing) and workflow. | Katrina Anderson |
| 2012-04-10 | 6.17 | Updated Test Cases and Iteration 6 Test Report | Katrina Anderson |

# Introduction

## Purpose

The purpose of this Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This Test Plan for the F.S.T.S. supports the following objectives:

It will target all the feature groups worked on throughout the various iterations. This coverage list has been determined to be the optimal approach since it allows the team to remain focused on the tasks and user stories in the sprint backlog. It also allows for the user stories to be completed, closed and accepted due to the fact that closing user stories requires for all tests to be completed and successful.

It will establish the main approaches for testing: Acceptance testing, unit testing, integration testing, system testing, UI testing volume testing and regression testing.

It will indicate approximate phases of testing activities, illustrate them in a Gantt Chart and determine what tools will be used for detailed testing documentation and tracking.

## Scope

The main approaches for testing will be acceptance testing and unit testing. In iteration 3 we plan on introducing integration and system testing types. We will elaborate on acceptance, unit, integration and system tests in iteration 4 and introduce UI testing in iteration 5.All tests, excluding user acceptance tests, will be regressive. For acceptance, UI and system tests we must achieve 100% coverage of the user story feature groups. For unit and integration testing, an objective of 60% statement code coverage has been set. In the final iteration 6, we will place the system under extreme volume to test performance.

## Intended Audience

This document is intended for the testing team, developers and their managers. A semi-technically knowledgeable person in the area of software testing should be able to understand the document as a whole. The testing team notably, should be able to use this as a reference for what is planned, how it will be done and why.

## Document Terminology and Acronyms

Please refer to *Section 14- Glossary of the SRS document* for the listing of terms and their definitions

## References

Please refer to *Appendix A – References* for the listing of existing resources that will be referenced to stimulate the identification and selection of specific tests to be conducted

# Evaluation Mission and Test Motivation

## Background

The point of this testing undertaking is to assure a higher quality of code and final product. As any developer would acknowledge, mistakes are made at any level of a coding project. This application is destined to be used by a charitable organization, whose employees have limited computer skills, it is important that the software is relatively bug free. Furthermore, due to the fact that this project offers a highly customized solution to a specific problem, customer feedback will be very important and valuable in completing detailed acceptance and UI tests.

Please refer to the *Vision Document* for more information on the background of this project.

## Evaluation Mission

The mission of this evaluation effort is to coordinate testing efforts in order to ensure optimal efficiency and ultimately, achieve user acceptance and defect containment.

## Test Motivators

### Risk Motivators

The success of this project will be evaluated based on overall quality and functionality achieved by the final solution; Testing helps to ensure optimal quality and traceability of the software. Other project based risks can be associated to testing, based on the notion that finding a bug early in the development process can be greatly beneficial. The concept implies that the cost of fixing a mistake grows exponentially the further a project progresses. Due to strict time constraints and limited resources, it is important to implement testing early in the project to reduce the risk of errors being discovered too late.

### Testing for Functionality and User Stories

The management system, VersionOne, includes user stories which each contain a set of subtasks and testing tasks. In order to properly track the team’s progress in an agile way, the team uses burndown charts, which are updated as user stories are completed and closed. VersionOne helps the team keep track of what functional requirements or user stories are done, to do and in progress. Using traceability matrices, one can see how all these requirements, user stories and test cases trace with each other. In the agile web application, in order to be able to mark a user story as completed, one must first complete all subtask and tests linked to that story. An alternate motive to testing is to ensure full coverage and traceability from the requirements to implementation and finally to one or more tests. Full implementation of a user story is achieved, since a test cannot pass unless the functionality is actually implemented.

### Design and Customer Satisfaction

Possibly the most important motive for testing in this iteration and all future iterations will be to ensure customer satisfaction. In the test cases involving acceptance testing, user feedback is necessary for success of the test. As previously discussed, the target customer is very specific in this project so many design decision will be taken based on input received during demos and meetings with the client. The tailored solution will ultimately be rejected in the case where the user acceptance testing would yield negative results.

### Summary

To conclude, testing in this project will have many motivations from risk management, functionality completion and customer satisfaction. The team has agreed that a primary objective of the project will be to help a company who dedicates time and money to a humanitarian cause. Down the line, this software will be used to serve the less fortunate faster and more effectively, hopefully providing better service. The end result must be of professional quality and standard. This objective cannot be guaranteed unless testing on all components, systems and interfaces is strictly enforced.

# Target Test Items

## Features to be Tested

A high level list of feature groups to be tested is provided below. Each feature group is comprised of at least one user story. All tests must pass in order for an associated user story to be considered completed. Feature development will be the primary focus of testing. If time permits the target test item list may expand to include supporting product elements.

1. Client File Management
2. Client File Search
3. Event Management
4. Appointment Management
5. Appointment Fulfillment
6. Operating Reports
7. Statistical Reports
8. ~~Special Query and Reporting~~\*Has been scoped out
9. Administrative Controls
10. ~~Bilingual System (French and English)~~\*Has been scoped out

## Features Not to be Tested

1. Any code pre-generated by the Yii framework will not be tested. It is assumed that any functions in the Yii API have been tested by Yii developers already. However, if any modifications or changes are made to that code, those changes will be tested.
2. During development there will be no hardware, peripheral device, or operating system tests executed during development. Development is local, therefore we cannot test these items. If the F.S.T.S. software is chosen, basic functional tests will be done for the hardware, operating system, and peripheral devices to ensure that the F.S.T.S. software will run as expected.

# Test Approach

The testing for the F.S.T.S., will be approached using seven testing techniques: acceptance testing, unit testing, integration testing, system testing UI testing, regression testing and volume testing. This section of the test plan will indicate the objective, target items, testing goals, technique, oracles, required tools, success criteria, failure contingencies and any special considerations involved with each testing methodology.

## Initial Test-Idea Catalogs and Other Reference Sources

Please refer to *Appendix A – References* for the listing of existing resources that will be referenced to stimulate the identification and selection of specific tests to be conducted

## Testing Techniques and Types

### Acceptance Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objective of acceptance testing is twofold; It allows testers to verify the functionality of the F.S.T.S. and it provides a means of assuring stakeholder satisfaction. Testers and stakeholders verify that the execution of tasks and functionalities are as expected, that error conditions are handled appropriately and that the navigation and appearance of the system during the execution of tasks is satisfactory. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested.* |
| **Testing Goal:** | 100% Coverage  Every user story will have a minimum of one acceptance test to be executed. User Stories cannot be closed until all related acceptance tests have passed. In addition, the system will not be deployed without acceptance from the stakeholders, thus ensuring 100% acceptance. |
| **Technique:** | Every acceptance test will be defined by a fully dressed test case, which will be listed in the *Test Case* document. The outline for test cases is included in *Applying UML and Patterns* by Craig Larman [5]. Acceptance tests are executed manually by following the steps in the *Description* section of each test case. |
| **Oracles:** | Acceptance tests are executed manually, so that users observe and determine the validity of the outcomes to their actions. |
| **Required Tools:** | **F.S.T.S**. – Tests are executed manually on this system.  **Internet Explorer 9** – User interface is accessed with this browser. |
| **Success Criteria:** | An individual acceptance test will “pass,” if all users (testers and stakeholders) determine that the outcomes listed in the *Expected Results* sections of a test case are executed satisfactorily. Completion of a user story is contingent on the success of all related acceptance tests. |
| **Failure contingencies:** | An acceptance test will fail if either a tester or a stakeholder has determined that it has not completed the *Expected Results* criteria satisfactorily. At the end of an iteration, any user stories associated to failing acceptance tests must be moved to the backlog to be re-evaluated at a later date. Until the end of the iteration, developers and testers must work to implement the changes required for acceptance tests to pass. |
| **Special Considerations:** | Not all functionalities can be achieved the way stakeholders envision. It is important to point out risks and compromise with stakeholders in order to implement a quality system, without sacrificing time or cost. |

### Unit Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objectives of TDD and unit testing are; One, to establish a basis for the successful completion of F.S.T.S. functionalities. This means that a developer will know when he has successfully implemented a functionality, because the corresponding unit test will pass. Two, to assure that changes in code do not affect or hinder previously implemented functionalities. A passed unit test should continue to pass, despite code development. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested.* |
| **Testing Goal:** | 60% Coverage  Unit tests must cover a minimum of 60% of the functionalities implemented by developers. Coverage will be measured by establishing the percent of statements included during the execution of unit tests.  *\*Note: As of now there does not seem to be an open-source software available that will measure branch and path coverage in PHP. As such, coverage can only be measured by statement.* |
| **Technique:** | Unit tests will be hard coded into the F.S.T.S. system using PHPUnit and will be executed automatically through the Netbeans IDE [6]. Unit tests shall be programmed according to the standards documented in the PHPUnit manual [7] |
| **Oracles:** | After the execution of unit test cases, PHPUnit will create an overview report stating whether each unit test passed or failed [7]. XDebug will provide the percent statement coverage of unit tests overall as well as for individual files [8]. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run unit tests.  **Netbeans IDE**– Required for executing unit tests and viewing XDebug and PHPUnit reports [6].  **PHPUnit** – Provides the framework to write unit test cases and produce results [7].  **XDebug** – Generates Statement coverage reports [8]. |
| **Success Criteria:** | An individual unit test will “pass” if its programmed objective has completed successfully, meaning without error.Completion of a user story is contingent on the success of all related unit tests. |
| **Failure contingencies:** | A unit test will fail if the execution of its programmed objective results in an error. At the end of an iteration, any user stories associated to failing unit tests must be moved to the backlog to be revaluated at a later date. Until the end of the iteration, developers and testers must work to implement the changes required for unit tests to pass. |
| **Special Considerations:** | Not all properties of the F.S.T.S. can be accessed or tested through unit testing. |

See IC Document Section 2.3.7 for Information on how to set up environment for unit testing

### Integration Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objectives of integration testing is to test the collection of dependent components of the F.S.T.S at once, in order to assure that no errors in structure occur during the user’s workflow. In case of failure, integration tests also help to isolate the location of a fault [11]. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested.* |
| **Testing Goal:** | 60% Coverage  Integration tests must cover a minimum of 60% of the functionalities implemented by developers. Coverage will be measured by establishing the percent of statements included during the execution of integration tests. |
| **Technique:** | F.S.T.S is being built on top of the Yii Framework, who’s central components have been integrated already. In the interest of time and efficiency we will assume that integration testing was covered by the Yii Framework developers. |
| **Oracles:** | We will be assuming that integration testing was covered by the Yii Framework Development Team. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run system tests.  **Internet Explorer 9** – Use browser to check system behaviour. |
| **Success Criteria:** | An individual integration test will “pass” if its programmed objectives have completed successfully, meaning without error. Completion of the F.S.T.S. is contingent on the success of all integration tests. |
| **Failure contingencies:** | An integration test will fail if the execution of its programmed objectives results in errors, meaning that the system structure is not responding appropriately. Until the end of the iteration, developers and testers must work to implement the changes required for integration tests to pass. |
| **Special Considerations:** | Not all properties of the F.S.T.S. can be accessed or tested through integration testing. |

### System Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objective of system testing is to test the behaviour of the F.S.T.S. as a whole, meaning once all the components in an iteration have been integrated [11]. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested* and depicted in *Figure 4.1 F.S.T.S System Test Paths* |
| **Testing Goal:** | 100% Feature Group Coverage  Coverage will be measured by establishing how many feature group components have been covered; Feature groups are composed of user stories. |
| **Technique:** | System tests will be conducted at the end of each iteration, after integration testing, and will consist of a giant workflow implementing all the system functionalities to date. They will be created using Selenium and will be executed automatically through Internet Explorer 9 and the Netbeans IDE [6]. System tests shall be programmed according to the standards documented in the Selenium manual [12] |
| **Oracles:** | After the execution of system test cases, Selenium will create an overview report stating whether each system test passed or failed [12]. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run system tests.  **Netbeans IDE**– Required for executing system tests and viewing XDebug and Selenium reports [6].  **Internet Explorer 9** – Selenium uses an actual browser to implement tests and check system behaviour.  **Selenium** – Automates testing within a browser [12]. |
| **Success Criteria:** | A system test will “pass” if its programmed objectives have completed successfully, meaning without error. Completion of the F.S.T.S. is contingent on the success of all system tests. |
| **Failure contingencies:** | A system test will fail if the execution of its programmed objectives results in errors, meaning that the system behaviour is not responding appropriately. Until the end of the iteration, developers and testers must work to implement the changes required for a system test to pass. |
| **Special Considerations:** | Not all properties of the F.S.T.S. can be accessed or tested through system testing. |

See IC Document Section 2.3.7 for Information on how to set up environment for system testing



Figure 4‑1F.S.T.S System Test Paths

### User Interface Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objective of UI testingis to ensure that the UI provides the user with the appropriate access and navigation through the functions of the F.S.T.S. In addition, UI testing ensures that the objects within the UI function as expected and conform to the standards set out by the Welcome Hall Mission. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested* |
| **Testing Goal:** | 100% Coverage of the F.S.T.S. Screens |
| **Technique:** | UI tests will be conducted at the end of each iteration during and after system testing. UI functionalities and navigation from window to window will be tested through workflows implemented with Selenium. Perception and placements of fields and information will be approved through acceptance testing with clients. UI tests shall be programmed according to the standards documented in the Selenium manual [12] |
| **Oracles:** | After the execution of UI test cases, Selenium will create an overview report stating whether each UI test passed or failed [12]. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run system tests.  **Netbeans IDE**– Required for executing system tests and viewing XDebug and Selenium reports [6].  **Internet Explorer 9** – Selenium uses an actual browser to implement tests and check system behaviour.  **Selenium** – Automates testing within a browser [12]. |
| **Success Criteria:** | A UI test will “pass” if its programmed and acceptance objectives have completed successfully, meaning without error. Completion of the F.S.T.S. is contingent on the success of all UI tests. |
| **Failure contingencies:** | A UI test will fail if the execution of its programmed and acceptance objectives result in errors, meaning that the functional and navigational behaviour of the system is not responding appropriately. Until the end of the iteration, developers and testers must work to implement the changes required for a UI test to pass. |
| **Special Considerations:** | Not all properties of the F.S.T.S. can be accessed or tested through UI testing. In addition, not all functionalities can be achieved the way stakeholders envision. It is important to point out risks and compromise with stakeholders in order to implement a quality system, without sacrificing time or cost. |

### Regression Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objective of regression testing is to seek out bugs and in implemented features after changes have been made to the system. In other words, regression testing involves re-running tests to assure that the addition of new features or modification of old features, doesn’t affect the functionality of the F.S.T.S. |
| **Target Items** | The feature groups listed in *Section 3.1 – Features to be Tested* |
| **Testing Goal:** | 60% for Unit Tests and Integration Tests  100% for acceptance, system and UI tests. |
| **Technique:** | Regression will be enabled through the use of PHPUnit and Selenium, which allow for tests to be programmed and re-run at any time through the Netbeans IDE [7][12]. Specifically unit, system and UI tests will be programmed to run regressively. |
| **Oracles:** | PHPUnit and Selenium will create overview reports stating whether each test passed or failed [7][12]. In addition XDebug will indicate statement coverage for unit tests. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run system tests.  **Netbeans IDE**– Required for executing system tests and viewing XDebug and Selenium reports [6].  **Internet Explorer 9** – Selenium uses an actual browser to implement tests and check system behaviour.  **PHPUnit** – Provides the framework to write unit test cases and produce results [7].  **XDebug** – Generates Statement coverage reports [8].  **Selenium** – Automates testing within a browser [12]. |
| **Success Criteria:** | A regression test will “pass” if its programmed and acceptance objectives have completed successfully, meaning without error. Completion of the F.S.T.S. is contingent on the continued success of all regression tests. |
| **Failure contingencies:** | A regression test will fail if the execution of its programmed objectives result in errors, meaning that the system doesn’t responding appropriately to programmed functionalities. Until the end of the iteration, developers and testers must work to implement the changes required for regression tests to pass. |
| **Special Considerations:** | Not all properties of the F.S.T.S. can be accessed or tested through testing with PHPUnit and Selenium. |

### Volume Testing

|  |  |
| --- | --- |
| **Technique Objective:** | The objective in volume testing will be to test and push the system’s boundaries as far as raw data is concerned. Raw data in the F.S.T.S. is determined by the amount of client records in the database. Furthermore, the amount of appointments will be a factor when generating reports, so this aspect will also be tested. Ultimately, these tests will determine how the F.S.T.S. performs under stress of a large amount of records to simulate a real life working environment. |
| **Target Items** | The Client Search Component and Statistical Report component will be tested under the stress of a fully loaded database.  Refer to UC1.7, UC1.8 and UC5.3 |
| **Testing Goal:** | Push the limits of the system with up to 100,000 client files |
| **Technique:** | The database is loaded with, for example, 10,000 clients and 10,000 appointments over a specified year. A tester is then in charge of running queries and tasks that imply the usage of a large amount of records. They must also keep track of running times by using a timer at the code level. Namely, the tested task will be the client search and the statistical reports. |
| **Oracles:** | Rather than manually inputting all these records, a PHPUnit unit test will run a script which can assert that all these records are created and saved successfully in the database. |
| **Required Tools:** | **F.S.T.S.** – Access to the system code is required to run tests.  **Netbeans IDE**– Required for executing tests and tracking performance  **PHP microtime ()** – counts the system clock time to keep track of execution time  **Internet Explorer 9** – Used by tester to perform the specific tasks  **PHPUnit** – Provides the framework to write unit test cases and produce results [7]. |
| **Success Criteria:** | If the tests on search and reporting can still function under this load then the volume test will be deemed succesful. All of the other possible tasks of the software require significantly less resources. As for the time of execution, there are no set requirements for reporting since this feature will be used once a year. As for search, the time required should be less than 5 seconds. |
| **Failure contingencies:** | A volume test will be a failure if the volume added causes the tested component to fail or to slow into an infinite loop. |
| **Special Considerations:** | Execution time and allocated memory should be set to unlimited in order to simulate a professional grade server set-up the program will run on in reality. |

See IC Document Section 2.3.7 for Information on how to set up environment and run volume testing

### Usability Testing

|  |  |
| --- | --- |
| **Technique Objective:** | Overall, the goals of the usability testing are to evaluate the level of ease towards learning and using the F.S.T.S’s interface, as well as the efficiency of the interface. |
| **Target Items** | **The feature groups listed in Section 3.1 – Features to be Tested.** |
| **Testing Goal:** | 100% Coverage of the F.S.T.S. Screens |
| **Technique:** | The usability test will be defined by a fully dressed test case, which will be listed in the *Test Case* document. The outline for test cases is included in *Applying UML and Patterns* by Craig Larman [5]. The usability test is executed manually by following the steps in the *Description* section of the test case. |
| **Oracles:** | Direct observation will be used to document the performance of the usability tests. |
| **Required Tools:** | **F.S.T.S**. – Tests are executed manually on this system.  **Internet Explorer 9** – User interface is accessed with this browser. |
| **Success Criteria:** | A usability test will “pass,” if all users (testers and stakeholders) determine that the outcomes listed in the *Expected Results* sections of a test case are executed satisfactorily. |
| **Failure contingencies:** | An acceptance test will fail if either a tester or a stakeholder has determined that it has not completed the *Expected Results* criteria satisfactorily. Any problems found shall be added as defects in the Version One repository. Until the end of the iteration, developers and testers must work to implement the changes required for usability tests to pass. |
| **Special Considerations:** | Not all functionalities can be achieved the way stakeholders envision. It is important to point out risks and compromise with stakeholders in order to implement a quality system, without sacrificing time or cost. |

# Testing Workflow

This section of the test plan document will list the relative test phases of the F.S.T.S. system and approximate their time spans. To visualize the test phase workflow and illustrate dependencies a Gantt chart will also be provided. Finally, the tools we plan to use for testing will be described and test phase slippage contingencies will be established.

## Test Phase Table

|  |  |  |
| --- | --- | --- |
| **Test Phase ID** | **Phase Description** | **Phase Period** |
| **TP-01** | **Iteration 1 : Create acceptance tests for user stories in VersionOne and iteration 1 deliverable.** | **January 17 – 18** |
| **TP-02** | **Iteration 1 : Testers execute acceptance tests from iteration 1.** | **January 28 – 29** |
| **TP-03** | **Iteration 1 : Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 1 deliverable.** | **January 30** |
| **TP-04** | **Iteration 2 : Install unit testing environment.** | **January 31** |
| **TP-05** | **Iteration 2: Create/Code unit and acceptance tests for user stories in VersionOne and iteration 2 deliverable.** | **February 1 – 3** |
| **TP-06** | **Iteration 2 : Run iteration 2 unit tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **February 5 – April 10** |
| **TP-07** | **Iteration 2 : Testers execute acceptance tests from iteration 2.** | **February 11 – 12** |
| **TP-08** | **Iteration 2: Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 2 deliverable accordingly.** | **February 13** |
| **TP-09** | **Iteration 3: Create/Code unit, system, integration and acceptance tests for user stories in VersionOne and Iteration 3 deliverable.** | **February 14 – 17** |
| **TP-10** | **Iteration 3 : Run iteration 3 unit tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **February 18 – April 10** |
| **TP-11** | **Iteration 3 : Run System tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **February 25 – 26** |
| **TP-12** | **Iteration 3 : Run Integration tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **February 23 – 24** |
| **TP-13** | **Iteration 3 : Testers execute acceptance tests from iteration 3.** | **February 25 – 26** |
| **TP-14** | **Iteration 3: Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 3 deliverable accordingly.** | **February 27** |
| **TP-15** | **Iteration 4: Create/Code unit, system, integration and acceptance tests for user stories in VersionOne and Iteration 4 deliverable.** | **February 28 – March 2** |
| **TP-16** | **Iteration 4 : Run iteration 4 unit tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 3 – April 10** |
| **TP-17** | **Iteration 4 : Run System tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 10 – 11** |
| **TP-18** | **Iteration 4 : Run Integration tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 8 – 9** |
| **TP-19** | **Iteration 4: Testers execute acceptance tests from iteration 4.** | **March 10-11** |
| **TP-20** | **Iteration 4: Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 4 deliverable accordingly.** | **March 12** |
| **TP-21** | **Iteration 4: Update Version One and Iteration 4 Document.** | **March 12** |
| **TP-22** | **Iteration 5: Create/Code unit, system, integration and acceptance tests for user stories in VersionOne and Iteration 5 deliverable.** | **March 13 – 16** |
| **TP-23** | **Iteration 5 : Run iteration 5 unit tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 17 – April 10** |
| **TP-24** | **Iteration 5 : Run System tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 24 – 25** |
| **TP-25** | **Iteration 5 : Run Integration tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 22 – 23** |
| **TP-26** | **Iteration 5: Run UI tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 24 – 25** |
| **TP-27** | **Iteration 5: Testers execute acceptance tests from iteration 5.** | **March 24 – 25** |
| **TP-28** | **Iteration 5: Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 5 deliverable accordingly.** | **March 26** |
| **TP-29** | **Iteration 6: Execute usability tests with volunteers and stakeholders.** | **March 26 – 30** |
| **TP-30** | **Iteration 6: Create/Code unit, system, integration and acceptance tests for user stories in VersionOne and Iteration 6 deliverable.** | **March 27 – 30** |
| **TP-31** | **Iteration 6 : Run iteration 6 unit tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **March 31 – April 10** |
| **TP-32** | **Iteration 6: Run System tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **April 9 – 10** |
| **TP-33** | **Iteration 6: Run Integration tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **April 8 – 7** |
| **TP-34** | **Iteration 6: Run UI tests and fix bugs that are found. Update VersionOne and iteration documents accordingly.** | **April 9 – 10** |
| **TP-35** | **Iteration 6: Testers execute acceptance tests from iteration 6.** | **April 9 – 10** |
| **TP-36** | **Iteration 6: Testers execute volume tests.** | **April 9 – 10** |
| **TP-37** | **Iteration 6: Execute stakeholder acceptance tests during demo. Update VersionOne and iteration 6 deliverable accordingly.** | **April 11** |

## ****Test Phase Gantt Chart****



## ****Test Management Tools****

We will be using a combination of three tools to organize and document our testing activities. Firstly, we will be using VersionOne as our centralised management tool for this project. User stories will be created within VersionOne and tests will be directly associated to user stories. Test creation and execution will be assigned to group members dynamically and can be viewed in a to-do list format. Secondly, we will be using Microsoft Excel to document traceability matrices; User Stories vs. Use Cases and Test Cases vs. Use Cases. The traceability matrices will simplify document navigation and establish clear links between features and test items. Lastly, we will be using Microsoft Word to report on testing progress and backup VersionOne data.

## Test Slippage

Test slippage is the result of testers being unable to accomplish testing goals due to time constraints. The F.S.T.S. testing workflow should aid in foreseeing if test slippage will occur. If developers and testers feel that slippage is inevitable, they must speak to the stakeholders and establish which defects are tolerable or reduce the feature set in order to deliver a viable system [9].

## Defects

Defects throughout the project will be documented in Version One. Starting in Iteration 5, defects and all associated data will be documented in the testing Report. Defects will be analyzed to establish quality metrics in the Iteration Report.

Please see Section 4- Reliability of the SRS Document for bug classifications.

Please see Test Reports for defect data.

Please see Iteration Planning and Reports for Quality metrics and analysis.

|  |
| --- |
| **F.S.T.S.** |

|  |
| --- |
| **Test Cases** |

# Test Cases

|  |
| --- |
| **Version 6.19** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev.** | **Description** | **Author(s)** |
| 2012-01-28 | 1.0 | Added Test Cases | Katrina Anderson |
| 2012-01-28 | 1.1 | Added Test Cases | Cynthia Donato |
| 2012-01-28 | 1.2 | Added Test Cases | Josh Hum |
| 2012-01-29 | 1.3 | Added Test Cases | Adrian Lloyd |
| 2012-01-29 | 1.4 | Added Test Cases | Patrick Modafferi |
| 2012-01-30 | 1.5 | Revised Document | Katrina Anderson |
| 2012-02-10 | 2.6 | Added Test Cases | Katrina Anderson |
| 2012-02-12 | 2.7 | Added Test Cases | Josh Hum |
| 2012-02-12 | 2.8 | Added Test Cases | Adrian Lloyd |
| 2012-02-12 | 2.9 | Added Test Cases | Patrick Modafferi |
| 2012-02-13 | 2.10 | Revised Document | Katrina Anderson |
| 2012-02-20 | 3.11 | Added Test Cases | Katrina Anderson |
| 2012-02-24 | 3.12 | Revised Document | Adrian Lloyd |
| 2012-02-24 | 3.13 | Added Test Cases | Josh Hum |
| 2012-02-24 | 3.14 | Added Test Cases | Patrick Modafferi |
| 2012-02-25 | 3.15 | Added Test Cases | Cynthia Donato |
| 2012-03-09 | 4.16 | Added Test Cases | Mikhail Levkovsky |
| 2012-03-19 | 5.17 | Added Test Cases | Cynthia Donato |
| 2012-04-09 | 6.18 | Added Volume Test Cases | Patrick Modafferi |
| 2012-04-10 | 6.19 | Added Usability Test Case & Reviewed Document | Katrina Anderson |

# Acceptance Test Cases

## TC01.2 Test Create User

**Test Case ID:** TC01-2

**Title:** Test Create User

**Requirement:**US01

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The current user logged in is an administrator.
3. The user has navigated to the Manage Users page.

**Description:**

1. The userselects to create a User.
2. The user fills out the form with valid input for a user and submits the form.

**Expected Results:**

1. The application will display a Create User form*.*
2. A user will be created and the system will redirect to view the summary information of the new user.

## TC02.2 Test Update User

**Test Case ID:** TC02-2

**Title:** Test Update User

**Requirement:**US02

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The current user logged in is an administrator.
3. The user has navigated to the Manage Users page.

**Description:**

1. The userselects a User to update.
2. The user changes at least one field in the form and submits it.

**Expected Results:**

1. The application will display an Update User form*.*
2. A user will be updated and the system will redirect to view the summary information of the user.

## TC03.2 Test Delete User

**Test Case ID:** TC03-2

**Title:** Test Delete User

**Requirement:**US03

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The current user logged in is an administrator.
3. The user has navigated to the Manage Users page.

**Description:**

1. The userselects a User to delete.
2. The user confirms that the selected User should be deleted.

**Expected Results:**

1. The application will display a confirmation box*.*
2. The selected User will be deleted.

## TC04.2 Test Set User to Administrator

**Test Case ID:** TC02-2

**Title:** Test Update User

**Requirement:**US02

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The current user logged in is an administrator.
3. The user has navigated to the Manage Users page.

**Description:**

1. The userselects a User to update.
2. The user changes the user type to *Administrator* and submits the form.

**Expected Results:**

1. The application will display an Update User form*.*
2. The user type will be updated to *Administrator* and the system will redirect to view the summary information of the user.

## TC05.1 Test View Manage Users

**Test Case ID:** TC05-1

**Title:** Test View Manage Users

**Requirement:**US05

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The current user logged in is an administrator.

**Description:**

1. The usernavigates to the *Admin Panel*.
2. The user navigates to the *User* management page.
3. The user navigates to *Manage User*.

**Expected Results:**

1. The application will display the administration page*.*
2. The application will display a list of users.
3. The application will display a table of users with links to update, delete, or view each one.

## TC06.1 Test User Login

**Test Case ID:** TC06-1

**Title:**Test User Login

**Requirement:** SR2.8, SR4.1, SR5.1, SR10.1, US6

**Type:**Acceptance

**Preconditions:**

1. The user must have a valid username and password registered with FSTS
2. The user must have a web browser open.

**Description:**

1. The user navigates to the login page.
2. The user enters a valid username and password combination in the login form and submits the form.

**Expected Results:**

1. The application displays the login form.
2. The application will log the user in and will allow access to functionality that a non-logged in user doesn’t have.

## TC06.2Test User Login – Omitted Password Field

**Test Case ID:** TC06-2

**Title:** Test User Login – Omitted Password Field

**Requirement:** SR2.8, SR4.1, SR5.1, SR10.1, US6

**Type:**Acceptance

**Preconditions:**

1. The user must have a valid username and password registered with FSTS
2. The user must have a web browser open.

**Description:**

1. The user navigates to the login page.
2. The user enters a valid username and leaves the password blank in the login form and submits the form.

**Expected Results:**

1. The application displays the login form.
2. The application will not allow the user to be logged in and will display a message indicating that the password cannot be blank.

## TC06.3 Test User Login – Invalid Password

**Test Case ID:** TC06-3

**Title::** Test User Login – Invalid Password

**Requirement:** SR2.8, SR4.1, SR5.1, SR10.1, US6

**Type:** Acceptance

**Preconditions:**

1. The user must have a valid username and password registered with FSTS
2. The user must have a web browser open.

**Description:**

1. The user navigates to the login page.
2. The user enters a valid username and an invalid password in the login form and submits the form.

**Expected Results:**

1. The application displays the login form.
2. The application will not allow the user to be loggedand will display a messageindicating that an incorrect username or password was entered.

## TC06.4 Test User Login – Invalid Username

**Test Case ID:** TC6-4

**Title:** Test User Login – Invalid Username

**Requirement:** SR2.8, SR4.1, SR5.1, SR10.1, US6

**Type:** Acceptance

**Preconditions:**

1. The user must have a valid username and password registered with FSTS
2. The user must have a web browser open.

**Description:**

1. The user navigates to the login page.
2. The user enters an invalid username and their password in the login form and submits the form.

**Expected Results:**

1. The application displays the login form.
2. The application will not allow the user to be loggedand will display a messageindicating that an incorrect username or password was entered.

## TC06.5 Test User Login – Omitted UsernameField

**Test Case ID:** TC06-5

**Title:**Test User Login – Omitted Username Field

**Requirement:** SR2.8, SR4.1, SR5.1, SR10.1, US6

**Type:** Acceptance

**Preconditions:**

1. The user must have a valid username and password registered with FSTS
2. The user must have a web browser open.

**Description:**

1. The user navigates to the login page.
2. The user enters their password but leaves the username field blank and submits the form.

**Expected Results:**

1. The application displays the login form.
2. The application will not allow the user to be logged in and will display a message indicating that the username cannot be blank.

## TC07-1 Test View Check-in for all Clients coming to an event – Successful

**Test Case ID:** TC07-1

**Title:** Test View All Clients Coming to an Event - Successful

**Requirement:** UC3.4, UC4.1, UC4.3

**Type:**Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged
3. Appointments to the event have been made

**Description:**

1. The user navigates to the Current Events section
2. The user selects the Check-in option for the particular event occurrence

**Expected Results:**

1. The application displays a list of all active events
2. The application displays the event occurrence details, a list of attended and the total number of people attending the event.

## TC08.1 Test View All Clients Coming to an Event –Success Scenario

**Test Case ID:** TC08-1

**Title:** Test View All Clients Coming to an Event – Success Scenario

**Requirement:** UC3.4, UC4.1, UC4.3

**Type:**Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged
3. Appointments to the event have been made

**Description:**

1. The user navigates to the Current Events section
2. The user selects to view the particular event occurrence

**Expected Results:**

1. The application displays a list of all active events
2. The application displays the event occurrence details, a list of attended and the total number of people attending the event.

## TC09-1 Test Closing an Event – Successful

**Test Case ID:** TC09-1

**Title:** Test Closing an Event – Successful

**Requirement:**

**Type:**Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged
3. Appointments to the event have been made
4. The current date is greater than or equal to the start date of the event

**Description:**

1. The user navigates to the Current Events section
2. The user selects the Check-in option for the particular event occurrence
3. The user presses the *Close Event* button

**Expected Results:**

1. The application displays a list of all active events
2. The application displays the event occurrence details, a list of attended and the total number of people attending the event.
3. The system displays a list of active event occurrences.

## TC10.1 Test Appointment Creation – Success Scenario

**Test Case ID:** TC10-1

**Title:** Test Appointment Creation – Success Scenario

**Requirement:**US10, UC3.1, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event occurrence for *Event 1*(appointment by hour) is created.

**Description:**

1. The usernavigates to a client’s file.
2. The usernavigates to the *Appointment*s section of the client file and selects *Add*.
3. The user selects an appointment for event occurrence *Event 1* and selects “Create.”

**Expected Results:**

1. The application displays the client file.
2. The application redirects the user to the *Appointment Management* page and displays a list of available appointment times.
3. The application displays the original client file, with the added appointment field.

## TC11.1 Test Appointment Update – Success Scenario

**Test Case ID:** TC11-1

**Title:** Test Appointment Update – Success Scenario

**Requirement:**US11, UC3.2, TC10-1, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event occurrence for *Event 1*(appointment by hour) is created.
4. The appointment for *Event 1* is created.

**Description:**

1. The usernavigates to a client’s file.
2. The usernavigates to the *Appointment*s section of the client file and selects to cancel the appointment for *Event 1*.
3. The usernavigates to the *Appointment*s section of the client file and selects *Add*.
4. The user selects an appointment and selects “Create.”

**Expected Results:**

1. The application displays the client file.
2. The application refreshes the client file and the appointment field is removed from the *Appointment* section of the client file.
3. The application redirects the user to the *Appointment Management* page and displays a list of available appointment times.
4. The application displays the original client file, with the added appointment field.

## TC12.1 Test Appointment Cancellation – Success Scenario

**Test Case ID:** TC12-1

**Title:** Test Appointment Cancellation – Success Scenario

**Requirement:**US12, UC3.3, TC10-1, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event occurrence for *Event 1*(appointment by hour) is created.
4. The appointment for *Event 1* is created.

**Description:**

1. The usernavigates to a client’s file.
2. The usernavigates to the *Appointment*s section of the client file and selects to cancel the appointment for *Event 1*.

**Expected Results:**

1. The application displays the client file.
2. The application refreshes the client file and the appointment field is removed from the *Appointment* section of the client file.

## TC13.1 Test Appointment Validation – Capacity Exceeded

**Test Case ID:** TC13-1

**Title:** Test Appointment Validation – Capacity Exceeded

**Requirement:**US10, UC3.1, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event occurrence for *Event 1*(appointment by hour) is created.

**Description:**

1. The usernavigates to a client’s file.
2. The usernavigates to the *Appointment*s section of the client file and selects *Add*.
3. The user selects an appointment for event occurrence *Event 1* and selects “Create.”
4. The user selects to exceed the appointment capacity.

**Expected Results:**

1. The application displays the client file.
2. The application redirects the user to the *Appointment Management* page and displays a list of available appointment times.
3. The application displays a dialogue box indicating that the capacity for an appointment slot will be exceeded.
4. The application displays the original client file, with the added appointment field.

## TC14.1 Test View Summary of Clients with Appointments to Events

**Test Case ID:** TC14-1

**Title:** Test View Summary of Clients with Appointments to Events

**Requirement:** UC3.4, UC4.1, UC4.3

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged
3. Appointments to the event have been made

**Description:**

1. The user navigates to the Current Events section
2. The user selects to view the particular event occurrence

**Expected Results:**

1. The application displays a list of all active events
2. The application displays the event occurrence details, a list of attended and the total number of people attending the event.

## TC16.1 Test Create Client File - Minimal Information

**Test Case ID:** TC16-1

**Title:**Test Create Client File - Minimal Information

**Requirement:** UC 1.1, US16

**Type:** Acceptance

**Preconditions:**

1. Navigate to client creation form

**Description:**

1. The user inputs first name : Olga Input client number: 123456 Input last name: Ormandjieva
2. The user creates the file

**Expected Results:**

1. The system displays the form with inputted values.
2. The system creates a file with the provided information.

- Default values are given from drop down lists.

- New residence with blank fields is created and added to the database.

## TC16.2 Test Create Client File – All Information

**Test Case ID:** TC16-2

**Title:**Test Create Client File – All Information

**Requirement:** UC 1.1, US16

**Type:** Acceptance

**Preconditions:**

1. Navigate to client creation form

**Description:**

1. The user fills in all fields contained in the form, including text and dropdown lists
2. The user creates the file

**Expected Results:**

1. The system displays client information in the form
2. The system creates the file with the provided information.

-Read only field for the creation time matches current time and date.

## TC16.3 Test Create Client File - Missing Required Fields

**Test Case ID:** TC16-3

**Title:**Test Create Client File - Missing Required Fields

**Requirement:** UC 1.1, US16

**Type:** Acceptance

**Preconditions:**

1. Navigate to client creation form

**Description:**

1. The user does not fill in any fields found on the form
2. The user then attempts to create the file

**Expected Results:**

1. The system form’s text fields are blank
2. The system highlights the fields marked with a red \* and outputs a message indicating that some of the required fields have not been filled.

The system does not create client file.

## TC16.4 Test Creating Client File - Address Exists

**Test Case ID:** TC16-4

**Title:**Test Create Client file - Address exists

**Requirement:** UC 1.1, US16, TC16-2

**Type:** Acceptance

**Preconditions:**

1. Navigate to client creation form.
2. A client file exists that has the address #123 Repeat and apartment number 123.

**Description:**

1. The user inputs name and number for mandatory fields, Input Address: 123, Input Street: Repeat, Input Apartment #: 123
2. The user attempts to create the file

**Expected Results:**

1. The system form contains the same address as a previous record
2. The system displays an error message

## TC17.1Test Update Client File - Main Success Scenario

**Test Case ID:** TC17-1.

**Title:**Test Update Client File - – Main Success Scenario

**Requirement:** UC1.1, US17

**Type:** Acceptance

**Preconditions:**

1. A client file must exist in the system.

**Description:**

1. The user selects a client file to be edited
2. The user changes all fields in the file
3. The user saves the file
4. The user reloads the client file to view the information

**Expected Results:**

1. The system displays the update form
2. The system updates the fields from the form.
3. The system updates all the edited fields to the new information input in step 2.
4. The system does not create a new file; the existing entry is simply updated. The system displays the updated client file.

## TC18.1 Test Set Client File to Active

**Test Case ID:** TC18-1

**Title:** Test Set Client File to Active

**Requirement:** UC1.3, US18

**Type:** Acceptance

**Preconditions:**

1. The user must have a web browser open.
2. The user must be on the FSTS home page or the client file admin page.

**Description:**

1. The user navigates to the client file to update.
2. The user ensures that IsActive is checked.

**Expected Results:**

1. The update client form will be displayed
2. The client file will be set to active.

## TC21.1Test Add New Flag – Success Scenario

**Test Case ID:** TC21-1

**Title:** Test Add New Flag – Success Scenario

**Requirement:** UC6.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the create flag page

**Description:**

1. The user enters a name for the flag in the name text box
2. The user enters a description for the flag in the description text box
3. The user selects a level for the flag from the level of importance drop down list
4. The user selects if the flag will be active or not
5. The user presses the Create Button

**Expected Results:**

1. The application displays the name being entered into the name text box
2. The application displays the description being entered into the description text box
3. The application displays the selection made instead of the blank (which is the default)
4. The application displays the selection made for setting the flag as active or not
5. The application displays redirects the user to the view flag page where the newly created flag information can be found. The application enters into the flag database table the data about the newly created flag

## TC21.2Test Add New Flag – Failure Scenario 1

**Test Case ID:** TC21-2

**Title:** Test Add New Flag – Failure Scenario 1

**Requirement:** UC6.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the create flag page

**Description:**

1. The user enters a name for the flag in the name text box
2. The user enters a description for the flag in the description text box
3. The user does not select a level for the flag from the level of importance drop down list (Blank option is selected)
4. The user presses the Create Button

**Expected Results:**

1. The application displays the name being entered into the name text box
2. The application displays the description being entered into the description text box
3. The application displays the selection made as blank (which is the default)
4. The application redirects the user to the same page and an error message will be displayed showing what the user was missing. In this case, the error will be “Level of Importance cannot be blank”.

## TC21.3Test Add New Flag – Failure Scenario 2

**Test Case ID:** TC21-3

**Title:** Test Add New Flag – Failure Scenario 2

**Requirement:** UC6.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the create flag page

**Description:**

1. The user presses the Create Button leaving all the fields blank

**Expected Results:**

1. The application redirects the user to the same page and an error message will be displayed showing what the user was missing. In this case, the error message will indicate that the name, description and level of importance fields cannot be blank.

## TC22.1Test Delete Flag – Success Scenario

**Test Case ID:** TC22-1

**Title:** Test Delete Flag – Success Scenario

**Requirement:** UC6.2

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. There is at least one flag in the flag list
4. The user is at the manage flags page or the view flag page

**Description:**

1. The user selects the “Delete Flag” link
2. The user selects “ok” when prompted if they are sure they wish to delete the flag

**Expected Results:**

1. The application presents the user with a confirm box with the buttons “ok” and “cancel”
2. The application redirects the user to the list of flags page where all the flags are shown. The flag deleted will not be shown in this list and the database table containing the flag data will have the data of the flag deleted

## TC23.1Test Update Flag – Success Scenario

**Test Case ID:** TC23-1

**Title:** Test Update Flag – Success Scenario

**Requirement:** UC6.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the manage flags page
4. There is at least one flag in the flag list

**Description:**

1. The user selects the “Update Flag” link on one of the flags listed
2. The user edits the name, the description and/or the level of importance
3. The user presses the Save Button

**Expected Results:**

1. The application redirects the user to the update flag page where the user can edit the fields of the flag
2. The application updates the name, the description and/or the level of importance fields as new data is updated
3. The application redirects the user to the view flag page where the updated flag information can be found and application updates the database table containing the flag data with the new data about the updated flag

## TC23.2Test Update Flag – Failure Scenario

**Test Case ID:** TC23-2

**Title:** Test Update Flag – Failure Scenario

**Requirement:** UC6.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the manage flags page
4. There is at least one flag in the flag list

**Description:**

1. The user selects the “Update Flag” link
2. The user edits the name, the description or the level of importance
3. The user leaves one of the fields mentioned in step 2) blank
4. The user presses the Save Button

**Expected Results:**

1. The application redirects the user to the update flag page where the user can edit the fields
2. The application renders the changes in the name, the description and the level of importance fields
3. The application leaves one of the fields blank (as chosen by the user)
4. The application redirects to the same page and displays to the user an error message showing what the user was missing.

## TC23.3Test Update Flag to Inactive – Success Scenario

**Test Case ID:** TC23-3

**Title:** Test Update Flag to Inactive – Success Scenario

**Requirement:** UC6.1, UC6.2

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the manage flags page
4. There is at least one active flag in the flag list

**Description:**

1. The user selects the “Update Flag” link on an active flag in the flags listed
2. The user set the flag to inactive
3. The user presses the Save Button

**Expected Results:**

1. The application redirects the user to the update flag page where the user can edit the fields of the flag
2. The application sets the Is Active drop down list to “0” (false)
3. The application redirects the user to the view flag page where the updated flag information can be found. The application updates the database table containing the flag data with the new data about the updated flag where in both cases the is active field should be set to 0

## TC23.4 Test Update Flag to Active – Success Scenario

**Test Case ID:** TC23-4

**Title:** Test Update Flag to Active – Success Scenario

**Requirement:** UC6.1, UC6.2

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in as an administrator
3. The user is at the manage flags page
4. There is at least one inactive flag in the flag list

**Description:**

1. The user selects the “Update Flag” link on an inactive flag in the flags listed
2. The user set the flag to active
3. The user presses the Save Button

**Expected Results:**

1. The application redirects the user to the update flag page where the user can edit the fields of the flag
2. The application sets the Is Active drop down list to “1” (true)
3. The application redirects the user to the view flag page where the updated flag information can be found. The application updates the database table containing the flag data with the new data about the updated flag where in both cases the is active field should be set to 1

## TC24.1 Test Set Flag on New Client File – Success Scenario

**Test Case ID:** TC24-1

**Title:** Test Set Flag on New Client File – Success Scenario

**Requirement:** UC1.1, UC1.5, US24, TC16-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. The user can successfully make a client file
4. The user is at the create client file page

**Description:**

1. The user fills out an entire valid form for creating a client
2. The user selects a flag from the flag list on the form
3. The user presses the Create Button

**Expected Results:**

1. The application renders all the required fields on the client form being filled
2. The application views that the flag chosen by the user is selected
3. The application redirects the user to the view client page where the selected flag has been set. The application updates the database table containing the client flag data with the data about the flag attached to the client file

## TC24.2 Test Set Flag on Existing Client File – Success Scenario

**Test Case ID:** TC24-2

**Title:** Set Flag on Existing Client File – Success Scenario

**Requirement:** UC1.2, UC1.5, US24, TC17-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. There is at least one client file in the database
4. The user is at the update client file page

**Description:**

1. The user selects a flag from the flag list on the form
2. The user presses the Save Button

**Expected Results:**

1. The application views that the flag chosen by the user is selected
2. The application redirects the user to the view client page where the selected flag has been set. The application updates the database table containing the client flag data with the data about the flag attached to the client file

## TC25.1 Test Remove Flag on Existing Client File – Success Scenario

**Test Case ID:** TC25-1

**Title:** Remove Flag on Existing Client File – Success Scenario

**Requirement:** UC1.2, UC1.12, US24, US25, TC17-1, TC24-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. There is at least one client file in the database
4. The user is at the update client file page of the file selected
5. The client file selected must have at least one flag set

**Description:**

1. The user deselects a flag from the flag list on the form
2. The user presses the Save Button

**Expected Results:**

1. The application views that the flag chosen by the user is deselected
2. The application redirects the user to the view client page where the selected flag has been deselected. The application updates the database table containing the client flag data and will not have the data about the flag that was attached to the client file

## TC26.1 Test Set Note Text on New Client File – Success Scenario

**Test Case ID:** TC26-1

**Title:** Test Set Note Text on New Client File – Success Scenario

**Requirement:** UC1.1, UC1.4, US26, TC16-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. The user can successfully make a client file
4. The user is at the create client file page

**Description:**

1. The user fills out an entire valid form for creating a client
2. The user submits the form
3. The user enters a note about the client in the notes text area
4. The user presses the Add Button

**Expected Results:**

1. The application renders the form as it is being filled out
2. The application makes the client and returns to the update page
3. The application renders the text in the notes text area
4. The application redirects the user to the view client page where the note has been set. The application updates the database table containing the client file data with the note text in the note field

## TC26.2 Test Set Note Text on Existing Client File – Success Scenario

**Test Case ID:** TC26-2

**Title:** Test Set Note Text on Existing Client File – Success Scenario

**Requirement:** UC1.2, UC1.4, US26, TC16-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. There is at least one client file in the database
4. The user is at the update client file page of the file selected
5. The client file selected must have the note field set

**Description:**

1. The user enters a note about the client in the note text area
2. The user presses the Save Button

**Expected Results:**

1. The application renders the text in the notes text area
2. The application redirects the user to the view client page where the note has been set. The application updates the database table containing the client file data with the note text in the note field

## TC27.1 Test Remove Note Text on Existing Client File – Success Scenario

**Test Case ID:** TC27-1

**Title:** Test Remove Note Text on Existing Client File – Success Scenario

**Requirement:** UC1.2, US24, US27, TC17-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. There is at least one client file in the database
4. The user is at the update client file page of the file selected
5. The client file selected must have the note field set

**Description:**

1. The user presses the Delete Buttonfor a note

**Expected Results:**

1. The application redirects the user to the view client page where the note has been set to blank. The application updates the database table containing the client file data and will not have the note text in the note field

## TC28.1 Test Update Note Text on Existing Client File – Success Scenario

**Test Case ID:** TC28-1

**Title:** Test Update Note Text on Existing Client File – Success Scenario

**Requirement:** UC1.2, UC1.4, US28, TC17-1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. The user is logged in
3. There is at least one client file in the database
4. The user is at the update client file page of the file selected
5. The client file selected must have the note field set

**Description:**

1. The user updates the note about the client in the note text area
2. The user presses the Save Button

**Expected Results:**

1. The application views the updated text of the note about the client in the note text area
2. The application redirects the user to the view client page where the new note text has been set. The application updates the database table containing the client file data with the new note text in the note field

## TC29.1 Test Client File – Dependent Association

**Test Case ID:** TC29-1

**Title:**Test Client File – Dependent Association

**Requirement:** UC 1.9

**Type:** Acceptance

**Preconditions:**

1. Complete successfully TC29-2 and TC16-1
2. Navigate to client creation form

**Description:**

1. The user inputs in the client form; client#: 1 first name: Al last name: J
2. The user inputs in the dependant form; first name: Bo last name: K Medicare#: 1
3. The user adds the dependant to the client file
4. The user creates the client file
5. The application redirects the user to the view page

**Expected Results:**

1. The system displays client information in the form
2. The system displays dependant information in the form
3. The system creates the dependant with the provided information
4. The system creates the client file with the provided information and has dependants associated to it
5. Information about the client and the dependant associated are displayed on the screen

## TC29.2 Test Create Dependents – All Information Fields

**Test Case ID:** TC29-2

**Title:**Test CreateDependents – All Information Fields

**Requirement:** UC 1.9

**Type:**Acceptance

**Preconditions*:***

1. Navigate to the dependent creation page

**Description:**

1. The user inputs information for all fields
2. The user creates the dependent

**Expected Results:**

1. The system displays information for the dependent in the form.
2. The system creates the dependent with the provided information and stores the record in the database

## TC29.3 Test Create Dependents - Missing Required Fields

**Test Case ID:** TC29-3

**Title:**Test Create Dependents - Missing Required Fields

**Requirement:** UC 1.9

**Type:**Acceptance

**Preconditions*:***

1. Navigate to the dependent creation page

**Description:**

1. The user does not input anything in the form
2. The user then attempts to create the dependent

**Expected Results:**

1. The form’s text fields are all blank
2. The system highlights the fields marked with a red \* and outputs a message indicating that some of the required fields have not been filled.

The system does not create client file.

## TC30.1 Test Dependent Deletion

**Test Case ID:** TC30-1

**Title:**Test Dependent Deletion

**Requirement:** UC 1.11

**Type:**Acceptance

**Preconditions:**

1. TC 29-1 must be complete and successful
2. Navigate to the client file page create in 29-1

**Description:**

1. The user navigates to the client file
2. The user clicks the link to the dependant associated to this client file
3. The user selects the option to delete the dependant

**Expected Results:**

1. The application directs to the client view page and a list of associated dependants is located on this page
2. The system displays the dependant view page
3. The dependant is deleted from the database and is no longer in the client view page

## TC31.1 Test Dependent Update

**Test Case ID:** TC31-1

**Title:**Test Dependent Update

**Requirement:** UC 1.10

**Type:** Acceptance

**Preconditions:**

1. A client file with dependants must exist in the system like in TC 29-1
2. The user is on the main page and can see the client created in 29-1

**Description:**

1. The user click the pencil to navigate to the update page
2. The user changes the name of the dependant from Bo to Lee
3. The user saves the changes made to the dependant and the client file
4. The application redirects the user to the view page

**Expected Results:**

1. The system displays the update form
2. The system updates the fields from the form.
3. The system updates all the edited fields to the new information input in step 2.
4. The system does not create a new dependant; the existing entry is simply updated. The system displays the updated client file with an updated dependant.

## TC32.3 Test Postal Code/City Association – with Region Mapped Valid Postal Code

**Test Case ID:** TC32-3

**Title:** Test Postal Code/City Association - with Region Mapped Valid Postal Code

**Requirement:** UC1.1, UC1.2, US32

**Type:** Acceptance

**Preconditions:**

1. The user must be on the create client page on the FSTS system.
2. The user must have a web browser open.

**Description:**

1. The user enters 3 characters of a postal code mapped to a city in the database.

**Expected Results:**

1. The city associated with that postal code will automatically appear in the city field of the form.

## TC32.4 Test Postal Code/City Association – with Invalid Postal Code

**Test Case ID:** TC32-4

**Title:** Test Postal Code/City Association – with Invalid Postal Code

**Requirement:** UC1.1, UC1.2, US32

**Type:** Acceptance

**Preconditions:**

1. The user must be on the create client page on the FSTS system.
2. The user must have a web browser open.

**Description:**

1. The user enters 3 characters of a postal code not mapped to a city in the database.

**Expected Results:**

1. Nothing should appear differently to the user.

## TC33.4 Test Create Country Field - Failure Scenario

**Test Case ID:** TC33-4

**Title:**Test Create Country Field - Failure Scenario

**Requirement:** UC1.1,S2.8, US33

**Type:** Acceptance

**Preconditions:**

1. The user has navigated to the Admin Panel

**Description:**

1. The user selects the Country option from the Admin Panel
2. The user chooses to Create a Country
3. The user attempts to create a country without inputting data

**Expected Results:**

1. The system displays the countries index page is show with a side menu for operations
2. The application redirects to the country creation form
3. The system highlights the fields marked with a red \* and outputs a message indicating that some of the required fields have not been filled.

The system does not create a country

## TC33.5 Test Create Language Field - Failure Scenario

**Test Case ID:** TC33-5

**Title:**Test Create Language Field - Failure Scenario

**Requirement:** UC1.1, S2.8, US33

**Type:** Acceptance

**Preconditions:**

1. The user has navigated to the Admin Panel

**Description:**

1. The user selects the language option from the Admin Panel
2. The user chooses to Create a language
3. The user attempts to create a language without inputting data

**Expected Results:**

1. The system displays the languages index page is show with a side menu for operations
2. The application redirects to the language creation form
3. The system highlights the fields marked with a red \* and outputs a message indicating that some of the required fields have not been filled.

The system does not create a language

## TC34.3 Test Updating Client File Field – Success Scenario

**Test Case ID:** TC34-3

**Title:**Test Updating Client File Field

**Requirement:** UC1.1,S2.8, US34

**Type:** Acceptance

**Preconditions:**

1. TC34-1 and TC34-2 must be successful
2. The user has navigated to the country creation page (see TC 33-4)

**Description:**

1. The user inputs Name: Cada
2. The user creates the country
3. The user selects to Update the country
4. The user inputs Name: Canada
5. The user saves the country
6. The user navigates to the client creation form
7. The user selects the Country field’s drop down list

**Expected Results:**

1. The system displays the country’s information in the form
2. The application redirects to the view page for the new country
3. The system displays the update form for countries
4. The system updates all the edited fields to the new information input in step 4.
5. The system does not create a new country; the existing entry is simply updated. The system displays the updated country with the new name
6. The system displays the client creation form
7. The system generates a list of countries including “Canada”

## TC34.4 Test Updating Client File Field – Failure Scenario

**Test Case ID:** TC34-4

**Title:**Test Updating Client File Field – Failure Scenario

**Requirement:** UC1.1, S2.8, US34

**Type:** Acceptance

**Preconditions:**

1. TC34-3 is completed and successful
2. User is in the country section of the admin panel

**Description:**

1. The user locates the country created by TC34-3
2. The user selects the link to the view for that country
3. The user selects to Update the country
4. The user removes input from the name field
5. The user attempts to save the country

**Expected Results:**

1. The country is visible in the list of countries
2. The application redirects to the view page for the country
3. The system displays the update form for countries
4. The system contains empty fields everywhere in the form
5. The system highlights the fields marked with a red \* and outputs a message indicating that some of the required fields have not been filled.

The system does not create a new country or update the country

## TC35.3 Test Deleting Entry in Country Field

**Test Case ID:** TC35-3

**Title:**Test Deleting Entry In Country Field

**Requirement:** UC1.1,S2.8, US35

**Type:** Acceptance

**Preconditions:**

1. TC34-3 is completed and successful
2. Only one instance of the country named “Canada” exists
3. User is in the country section of the admin panel

**Description:**

1. The user locates the country created by TC34-3
2. The user selects the link to the view for that country
3. The user selects to Delete the country
4. The user confirms the decision
5. The user navigates to the client creation form
6. The user selects the Country field’s drop down list

**Expected Results:**

1. The country is visible in the list of countries
2. The application redirects to the view page for the country
3. The system prompts the user for confirmation
4. The system removes the country and its information from the database
5. The system displays the client creation form
6. The system generates a list of countries and “Canada” is no longer included

## TC36.1 Test View Client File Foreign Key Mapping

**Test Case ID:** TC36-1

**Title:**Test View Client File Foreign Key Mapping

**Requirement:** UC 1.1, US36

**Type:**Acceptance

**Preconditions:**

1. Test Case TC16-2 must be completed and successful
2. Navigate to client view page and select the existing client created by TC16-2

**Description:**

1. The user has completed TC16-2 and make sure all fields are filled in.
2. The user opens the client file on the client view page

**Expected Results:**

1. The database contains a client file with full information from TC16-2
2. The system displays strings like English, Canada, etc. instead of the foreign keys

-System will display detailed residence information like street and number instead of residence ID.

-The database associates foreign keys to the appropriate strings.

## TC36.2 Test View Client File User Acceptance of General Layout

**Test Case ID:** TC36-2

**Title:**Test View Client File User Acceptance of General Layout

**Requirement:** UC 1.1, US36

**Type:**Acceptance

**Preconditions:**

1. Complete Design of the UI for a client file

**Description:**

1. The developers will show the user the prototype

**Expected Results:**

1. The user will provide feedback.

- Adjustments are made according to feedback.

- Ultimately, the user reaches an agreement on what the UI should look like

## TC37.2 Test Search Client Files by Client ID

**Test Case ID:** TC37-2

**Title:** Test Search Client Files by Client ID

**Requirement:** UC1.7, US37

**Type:** Acceptance

**Preconditions:**

1. The user must have a web browser open.
2. The user must be on the FSTS home page or the client file admin page.

**Description:**

1. The user enters an existing client id in the client id search field and presses enter.

**Expected Results:**

1. The client file will be presented to the user.

## TC38.9 Test Search Client Files by Last Name

**Test Case ID:** TC38-9

**Title:** Test Search Client Files by Last Name

**Requirement:** UC1.8, US38

**Type:** Acceptance

**Preconditions:**

1. The user must have a web browser open.
2. The user must be on the FSTS home page or the client file admin page.

**Description:**

1. The user navigates to the Advanced Search form.
2. The user enters a last name and presses enter.

**Expected Results:**

1. The Advanced Search form will be displayed.
2. The client file(s) will be displayed that match the last name entered.

## TC38.10 Test Search Client Files by Last Name and Medicare Number

**Test Case ID:** TC38-10

**Title:** Test Search Client Files by Last Name and Medicare Number

**Requirement:** UC1.8, US38

**Type:** Acceptance

**Preconditions:**

1. The user must have a web browser open.
2. The user must be on the FSTS home page or the client file admin page.

**Description:**

1. The user navigates to the Advanced Search form.
2. The user enters a last name and a Medicare card number and presses enter.

**Expected Results:**

1. The Advanced Search form will be displayed.
2. The client file(s) will be displayed that match both the last name and Medicare card number entered.

## TC39.1 Test Event Template Creation– No Appointment & No Repetition

**Test Case ID:** TC39-1

**Title:** Test Event Template Creation – No Appointment and No Repetition

**Requirement:** UC2.1, US39

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator selects the *Create Event* operation.
3. The administrator specifies *Name* to be “Event 1”, *Appointment Needed* to be “No” and *Repeatable* to be “No”. Leave all other fields empty and click *Create*.

**Expected Results:**

1. The application displays a list of current event templates along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the administrator to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC39.2 Test Event Template Creation – No Appointment & Repetition

**Test Case ID:** TC39-2

**Title:** Test Event Template Creation – Appointment Needed and No Repetition

**Requirement:** UC2.1, US39

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator selects the *Create Event* operation.
3. The administrator specifies *Name* to be “Event 2”, *Appointment Needed* to be “Yes”, *Appointment Taken By* “Hour” and *Repeatable* to be “No”. Leave all other fields empty and click *Create*.

**Expected Results:**

1. The application displays a list of current event templates along with a list of operations to *Create Event* and *Manage Event*.
2. The application displays an empty form for creating events.
3. The application redirects the administrator to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC39.3Test Event Template Creation - Appointment & Repetition

**Test Case ID:** TC39-3

**Title:** Test Event Template Creation – Appointment Needed and Event Repeats

**Requirement:** UC2.1, US39

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator selects the *Create Event* operation.
3. The administrator specifies *Name* to be “Event 3”, *Appointment Needed* to be “Yes”, *Appointment Taken By* “Hour”, *Repeatable* to be “Yes” and *Weekday* to be “Monday” and “Wednesday”. Leave all other fields empty and click Create.

**Expected Results:**

1. The application displays a list of current event templates along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the administrator to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC39.4 Test Event Template Creation

**Test Case ID:** TC39-4

**Title:** Test Event Template Creation – No Appointment Needed and Event Repeats

**Requirement:** UC2.1, US39

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator selects the *Create Event* operation.
3. The administrator specifies *Name* to be “Event 4”, *Appointment Needed* to be “No”, *Repeatable* to be “Yes” and *Weekday* to be “Monday” and “Wednesday”. Leave all other fields empty and click Create.

**Expected Results:**

1. The application displays a list of current event templates along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the administrator to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC39.5Test Event Template Creation– Mandatory Field Omitted

**Test Case ID:** TC39-5

**Title:** Test Event Template Creation – Mandatory Field Omitted

**Requirement:** UC2.1, US39

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator selects the Create Event operation.
3. The administrator specifies *Appointment Needed* to be “No”, *Repeatable* to be “Yes” and *Weekday* to be “Monday” and “Wednesday”. Leave all other fields empty and click Create.

**Expected Results:**

1. The application displays a list of current event templates along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application will display a red error box displaying the “Name cannot be blank” error message. The Name field in the form is also highlighted red and re-displays the error message below the field.

## TC40.1Test Event Template Deletion

**Test Case ID:** TC40-1

**Title:** Test Event Template Deletion

**Requirement:** US2.3, US40, TC39-1, UC2.3

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. The event *Event 1* must already exist in the system.

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator clicks the delete option for *Event 1.*
3. The administrator selects *OK*.

**Expected Results:**

1. The application displays a searchable list of all event templates along with a menu bar which allows for view, update and delete of individual event templates.
2. The application displays a confirmation window with the text “Are you sure you want to delete this item?”
3. The application will display the list of all events, which is updated to no longer display *Event 1*

## TC41.1Test Event Template Update

**Test Case ID:** TC41-1

**Title:** Test Event Template Update

**Requirement:** UC2.2, US41, TC39-2, UC2.3

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. The event *Event 2* must already exist in the system

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator clicks the update option for Test Event 2
3. The administrator updates the *Appointment Taken By* field to be “Day” and clicks Save

**Expected Results:**

1. The application displays a searchable list of all event templates along with a menu bar which allows for view, update and delete of individual event templates.
2. The application will display an editable form, containing the current event template information
3. The application the updated event in read-only mode

## TC41.2Test Event Template Update - Mandatory Field Omitted

**Test Case ID:** TC41-2

**Title:** Test Event Template Update – Mandatory Field Omitted

**Requirement:** UC2.2, US41, TC39-2, UC2.3

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. The event *Event 2* must already exist in the system

**Description:**

1. The administrator navigates to the event template management page.
2. The administrator clicks the update option for Event 2
3. The administrator updates the *Name* field to be an empty string and clicks Save

**Expected Results:**

1. The application displays a searchable list of all event templates along with a menu bar which allows for view, update and delete of individual event templates.
2. The application form is displayed containing the current event template information.
3. The application will display a red error box, displaying the “Name cannot be blank” error message. The Name field in the form is also highlighted red and re-displays the error message below the field.

## TC42.1Test Display of a List of Event Templates

**Test Case ID:** TC42-1

**Title:** Test Display of a List of Event Templates

**Requirement:** UC2.4, US42, TC39-1, TC39-2, TC39-3, TC39-4

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. Four events; *Event 1, Event 2, Event 3 and Event 4*, must already exist in the database.

**Description:**

1. The administratornavigates to the event template management section by selecting *Manage Event*
2. The administrator navigates to the list of event templates by selecting *List Event.*

**Expected Results:**

1. The application displays a summary of events as well as a navigation menu containing the operations *List Event* and *Create Event*.
2. The application displays a detailed list of events *Event 1, Event 2, Event 3 and Event 4*.

## TC42.2 Test Display of Event Template Details

**Test Case ID:** TC42-2

**Title:** Test Display of Event Template Details

**Requirement:** UC2.4, US42, TC39-1

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. One event, *Event 1*, must already exist in the database.

**Description:**

1. The administrator navigates to the event template management section by selecting *Manage Event*
2. The administrator selects the *View* icon, which is located in the same row as Event 1.

**Expected Results:**

1. The application displays a summary of events as well as a navigation menu containing the operations *List Event* and *Create Event*.
2. The application displays the event details specified in TC39-1.

## TC43.5 Event Occurrence Creation– No Appointment

**Test Case ID:** TC43-5

**Title:** Test Acceptance of Event Occurrence Creation– No Appointment

**Requirement:** US43, UC2.6, TC39-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user selects the *Create Event* operation.
3. The user specifies *Event* to be “Event 1”, *Appointment Needed* to be “No”, *Date* to be “February 13 2012” and *Max Capacity* to be “300”.

**Expected Results:**

1. The application displays a list of current event occurrences along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the user to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC43.6 Test Event Occurrence Creation–Appointment by Hour

**Test Case ID:** TC43-6

**Title:** Test Acceptance of Event Occurrence Creation–Appointment by Hour

**Requirement:** US43, UC2.6, TC39-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user selects the *Create Event* operation.
3. The user specifies *Event* to be “Event 1”, *Appointment Needed* to be “Yes”, *Appointments Taken By* to be ‘Hour”, *Date* to be “February 13 2012” and *Max Capacity* to be “300”.

**Expected Results:**

1. The application displays a list of current event occurrences along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the user to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC43.7 Test Event Occurrence Creation–Appointment by Day

**Test Case ID:** TC43-7

**Title:** Test Acceptance of Event Occurrence Creation–Appointment by Hour

**Requirement:** US43, UC2.6, TC39-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user selects the *Create Event* operation.
3. The user specifies *Event* to be “Event 1”, *Appointment Needed* to be “Yes”, *Appointments Taken By* to be ‘Hour”, *Date* to be “February 13 2012” and *Max Capacity* to be “300”.

**Expected Results:**

1. The application displays a list of current event occurrences along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application redirects the user to a non-modifiable view of the created event and its details. The event details should correspond to those specified in Step 3 of the Description section.

## TC43.8 Test Event Occurrence Creation–Mandatory Field Omitted

**Test Case ID:** TC43-8

**Title:** Test Acceptance of Event Occurrence Creation–Mandatory Field

**Requirement:** US43, UC2.6, TC39-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user selects the *Create Event* operation.
3. The user specifies *Event* to be “Event 1”, *Appointment Needed* to be “No” and leaves all other fields blank.

**Expected Results:**

1. The application displays a list of current event occurrences along with a list of operations to *Create Event* and *Manage Event.*
2. The application displays an empty form for creating events.
3. The application will display a red error box displaying the “Please fix the following input errors: Date cannot be blank; Max Capacity cannot be blank” error message. The *Date* and *Max Capacity* fields in the form are also highlighted red and display the error messages “Date cannot be blank” and “Max Capacity cannot be blank” below the respective fields.

## TC44.2 Test Event Occurrence Cancellation

**Test Case ID:** TC44-2

**Title:** Test Acceptance of Event OccurrenceCancellation

**Requirement:** US44, UC2.9, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event *Event 1* must already exist in the system.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user clicks the cancel option for *Event 1.*
3. The user selects *OK*.

**Expected Results:**

1. The application displays a searchable list of all event occurrences along with a menu bar which allows for view, update and cancel of individual event occurrences.
2. The application displays a confirmation window with the text “Are you sure you want to cancel this item?”
3. The application will display the list of all events, which is updated to no longer display *Event 1.*

## TC45.3 Test Event Occurrence Update

**Test Case ID:** TC45-3

**Title:** Test Acceptance of Event Occurrence Update

**Requirement:** US45, UC2.8, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event *Event 1* must already exist in the system.

**Description:**

1. The user navigates to the event template management page.
2. The user selects the update option for *Event 1*
3. The user updates the *Date* field to be “March 13 2012” and clicks *Save*

**Expected Results:**

1. The application displays a searchable list of all event occurrences along with a menu bar which allows for *view*, *update* and *cancel* of individual event occurrences.
2. The application will display an editable form, containing the current event occurrence information
3. The application redirects the user to a non-modifiable view of the updated event and its details. The changed event details should correspond to those specified in Step 3 of the Description section.

## TC45.4Test Event Occurrence Update - Mandatory Field Omitted

**Test Case ID:** TC45-4

**Title:** Test Acceptance of Event Occurrence Update– Mandatory Field Omitted

**Requirement:** US45, UC2.8, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event *Event 1* must already exist in the system.

**Description:**

1. The user navigates to the event occurrence management page.
2. The user selects the update option for *Event 1*
3. The user leaves the *Date* and *Max Capacity* fields blank.

**Expected Results:**

1. The application displays a searchable list of all event occurrences along with a menu bar which allows for *view*, *update* and *cancel* of individual event occurrences.
2. The application will display an editable form, containing the current event occurrence information
3. The application will display a red error box displaying the “Please fix the following input errors: Date cannot be blank; Max Capacity cannot be blank” error message. The *Date* and *Max Capacity* fields in the form are also highlighted red and display the error messages “Date cannot be blank” and “Max Capacity cannot be blank” below the respective fields.

## TC46.1Test Display List of Active Event Occurrences

**Test Case ID:** TC46-1

**Title:** Test Display List of Active Event Occurrences

**Requirement:** UC2.10, US46, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event *Event 1* must already exist in the system.

**Description:**

1. The user navigates to the event occurrence management section by selecting *Manage Events.*
2. The user navigates to the list of event occurrences by selecting *List Event Occurrences.*

**Expected Results:**

1. The application displays a summary of events as well as a navigation menu containing the operations *List Event* and *Create Event*.
2. The application displays a list containing one event, *Event 1*.

## TC47.2 Test Display of Event Occurrence Details

**Test Case ID:** TC47-2

**Title:** Test Display of Event Occurrence Details

**Requirement:** UC2.11, US46, TC39-1, TC43-1

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event *Event 1* must already exist in the system.

**Description:**

1. The user navigates to the event occurrence management section by selecting *Manage Events.*
2. The user selects the *View* icon, which is located in the same row as *Event 1*.

**Expected Results:**

1. The application displays a summary of events as well as a navigation menu containing the operations *List Event* and *Create Event*.
2. The application displays the event details specified in TC43-1

## TC48.1Test Generate Operational Reports – Success Scenario

**Test Case ID:** TC48-1

**Title:** Test Generate Operational Report – Success Scenario

**Requirement:** UC5.1, UC5.2

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running and is in the administrator panel
2. There is information of an event that has passed (including associated information such as appointments etc.)

**Description:**

1. The administrator navigates to the generate operational reports section
2. The administrator selects a predefined operational report or selects to make a new report by defining a time period
3. The administrator uses the filtering to select the fields that should appear on the report
4. The administrator selects the generate report button
5. The administrator exports the report

**Expected Results:**

1. The system displays to the user the operational reports section
2. The system displays to the user the report filtering options
3. The system reflects to the user the fields selected for the report
4. The system displays the generated operational report to the user
5. The system exports the report to a file (onto the desktop by default) or prints the report from the screen the system displays

## TC50.1Test Generate Statistical Reports – Success Scenario

**Test Case ID:** TC50-1

**Title:** Test Set Generate Statistical Report – Success Scenario

**Requirement:** UC5.1, UC5.3

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running and is in the administrator panel
2. There is at least one client file in the database

**Description:**

1. The administrator navigates to the generate statistical reports section
2. The administrator selects a predefined statistical report or selects to make a new report by defining a time period
3. The administrator uses the filtering to select the fields that should appear on the report
4. The administrator selects the generate report button
5. The administrator exports the report

**Expected Results:**

1. The system displays to the user the statistical reports section
2. The system displays to the user the report filtering options
3. The system reflects to the user the fields selected for the report
4. The system displays the generated statistical report to the user
5. The system exports the report to a file (onto the desktop by default) or prints the report from the screen the system displays

## TC51.2 Test for Creating an Income Source

**Test Case ID:**TCD51-2

**Title:** Test for creating an income sources

**Requirement:**UC1.1, US51

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.

2. The user has created a client.

3. The user is on the client update page.

**Description:**

1. The user fills out the income form.
2. The user saves the income form.

3. The user navigates to the client file view page

**Expected Results:**

1. The system displays the updated client file with the newly created income source.

## TC52.2 Test for Deleting an Income Source

**Test Case ID:**TCD52-2

**Title:** Test for deleting an income sources

**Requirement:**US52

**Type:**Acceptance

**Preconditions:**

1. 1. TC 51.2 has successfully passed
2. The user is on the client update page

**Description:**

1. The user selects to delete an existing income.

2. The user navigates to the client file view page

**Expected Results:**

1. The system displays the updated income information in the form

2. The system deletes the income record from the database

## TC53.2 Test for updating an income source

**Test Case ID:**TCD53-2

**Title:** Test for updating an income sources

**Requirement:**UC1.2, US53

**Type:**Acceptance

**Preconditions:**

1. TC 51.2 has successfully passed
2. The user is on the client update page

**Description:**

1. The user changes the information currently in the existing income form.

2. The user saves the income form

3. The user navigates to the client file view page

**Expected Results:**

1. The system displays the updated income information in the form

2. The system updates the database with the new income inform that was provided.

3. The page displays who updated this income as well as when.

## TC54.2 Test for the viewing of the income sources

**Test Case ID:**TCD54-2

**Title:** Test for the viewing of the income sources

**Requirement:**US54

**Type:**Acceptance

**Preconditions:**

1. TC 51.2 has successfully passed.
2. The user is on the client update page.

**Description:**

1. The user fills out the income section of the form of a client file
2. The user saves the income form
3. The user navigates to the client file view page

**Expected Results:**

1. The system displays the income information in the form
2. The system creates the income with the information that was provided
3. The page displays who registered this income as well as when.

## TC55.1 Test Add Client to Waiting List – Success Scenario

**Test Case ID:** TC55-1

**Title:** Test Add Client to Waiting List – Success Scenario

**Requirement:** UC3.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. There is at least one client file in the database eligible for an appointment
3. There is at least one event in the database

**Description:**

1. The user opens client file
2. The user selects to add an appointment from the appointment section
3. The user selects an event to add the appointment to
4. The user selects to add to waiting list

**Expected Results:**

1. The system displays the client file information to the user
2. The system displays a pop up that contains a list of open events
3. The system navigates to the time slots of the event chosen
4. The system displays a message saying that the client was added to the waiting list

## TC55.2 Test Remove Client from Waiting List – Success Scenario

**Test Case ID:** TC55-2

**Title:** Test Remove Client from Waiting List – Success Scenario

**Requirement:** UC3.1

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running
2. There is at least one event in the database
3. There is at least one client file in the database that’s on the waiting list for an event

**Description:**

1. The user opens client file
2. The user selects to view all of the client’s appointments from the appointment section
3. The user selects to remove the client from the waiting list

**Expected Results:**

1. The system displays the client file information to the user
2. The system displays a pop up that contains a list of the client’s current appointments along with the household’s waiting list
3. The system navigates to the add appointment page for the client and displays a message saying that the client was remove from the waiting list

## TCD01.2 Test for creation of correct times for time fields

**Test Case ID:**TCD01-02

**Title:** Test for creation of correct times for time fields – Success Scenario

**Requirement:**UC1.1, UC1.2

**Type:**Functional

**Preconditions:**

1. The user is logged into the system.
2. The user is on the client creation page.

**Description:**

1. The userinputs valid name and client number
2. The user creates the client file
3. The user navigates to the view page for this client

**Expected Results:**

1. The system displays client information in the form
2. The system creates the client with the provided information
3. The page contains the current date for fields: Registration Date, Last Updated and Last Visited

## TCD20.1 Test Deleting an Event with No Associated Appointments – Successful

**Test Case ID:** TCD20-1

**Title:** Test Deleting an Event with No Associated Appointments – Successful

**Requirement:** UC2.9, D20

**Type:** Acceptance

**Preconditions:**

1. The user has the web application running.
2. The user is logged .
3. *Event Occurrence 1* exists.
4. No appointments have been made for the particular *Event Occurrence 1.*

**Description:**

1. The user navigates to the Current Events section.
2. The user selects to view *Event Occurrence 1.*
3. The user selects to delete the *Event Occurrence 1.*

**Expected Results:**

1. The application displays a list of all active events.
2. The application displays the details, a list of attended and the total number of people attending *Event Occurrence 1*.
3. The system redirects to the list of active events which no longer contains *Event Occurrence 1.*

## TCD33.1 Test Create Appointment per Household Member – Successful

**Test Case ID:** TCD33-1

**Title:** Test Create Appointment per Household Member – Successful

**Requirement:**US10, UC3.1, TC39-1, TC43-1, D33

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.
3. The event occurrence for *Event 1*(appointment by hour) is created.

**Description:**

1. The user navigates to a client’s file.
2. The user navigates to the *Appointment*s section of the client file and selects *Add*.
3. The user selects a household member and a timeslot for event occurrence *Event 1* and selects “Create.”
4. The user acknowledges the confirmation message.

**Expected Results:**

1. The application displays the client file.
2. The application redirects the user to the *Appointment Management* page and displays a list of available appointment times.
3. The application displays a confirmation message.
4. The application displays the original client file, with the added appointment field.

## TCD34.1 Test Set Event Inactive – Successful

**Test Case ID:** TCD34-1

**Title:** Test Set Event Inactive – Successful

**Requirement:**UC2.8,D34

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event template for *Event 1* must be created.

**Description:**

1. The user navigates to the Current Events section.
2. The user selects to view *Event Occurrence 1.*
3. The user selects to set the status inactive*.*

**Expected Results:**

1. The application displays a list of all active events.
2. The application displays the details, a list of attended and the total number of people attending *Event Occurrence 1*.
3. The system displays a confirmation message indicating that *Event Occurrence 1* was saved*.*

## TCD34.2 Test Set Event Amount – Successful

**Test Case ID:** TCD34-2

**Title:** Test Set Event Amount – Successful

**Requirement:**UC2.8,D34

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event occurrence for *Event Occurrence 1* must be created.

**Description:**

1. The user navigates to the Current Events section.
2. The user selects to view *Event Occurrence 1.*
3. The user selects to set the amount value to 2000*.*

**Expected Results:**

1. The application displays a list of all active events.
2. The application displays the details, a list of attended and the total number of people attending *Event Occurrence 1*.
3. The system displays a confirmation message indicating that *Event Occurrence 1* was saved*.*

## TCD35.1 Test Closing an Event Sets the Last Visited Date for All Attendee

**Test Case ID:** TCD35-1

**Title:** Test Closing an Event Sets the Last Visited Date for All Attendee

**Requirement:**UC2.8,D35

**Type:**Acceptance

**Preconditions:**

1. The user is logged into the system.
2. The event occurrence for *Event Occurrence 1* must be created.
3. At least two appointments must exists *Appointment1*  and *Appointment2*
4. The last appointment date for *Appointment1*  and  *Appointment2*  is not the current date

**Description:**

1. The user navigates to the Current Events section.
2. The user selects to check-in for*Event Occurrence 1.*
3. The user checks-in *Appointment1.*
4. The users selects to close the event.
5. The user selects to search a client.
6. The user enters the client number for the household of *Appointment1*
7. The user selects the appropriate client.
8. The user selects to search a client.
9. The user enters the client number for the household of *Appointment2*
10. The user selects the appropriate client

**Expected Results:**

1. The application displays a list of all active events.
2. The application displays the details, a list of attended and the total number of people attending *Event Occurrence 1*.
3. The system refreshes and displays *Appointment1*  as checked-in.
4. The system redirects to the list of event occurrences*.*
5. The system displays the search screen.
6. The system displays a list of matching client files.
7. The system displays the client file with the last appointment field set to todays’ date.
8. The system displays the search screen.
9. The system displays a list of matching client files.
10. The system displays the client file with the last appointment field not set to todays’ date.

## TCD56.1 Test Generate Yearly Report – Successful

**Test Case ID:** TCD56-1

**Title:** Test Generate Yearly Report – Successful

**Requirement:**UC5.3

**Type:**Acceptance

**Preconditions:**

1. The administrator is logged into the system.
2. The administrator is on the Admin Panel

**Description:**

1. The administrator selects the Statistical report option in the admin panel
2. The administrator inputs a start date in the past: 01-04-2012
3. The administrator checks the option to generate a yearly report
4. The administrator attempts to create a report

**Expected Results:**

1. The report creation form is displayed
2. The system updates the form to contain the date
3. The system updated the form to show the checkbox checked
4. The system generated and downloads an excel type report

# Volume Test Cases

## TCV-01 Volume Test Generate Yearly Report (Excel)

**Test Case ID:** TCV-01

**Title:** Volume Test Generate Yearly Report

**Requirement:**UC5.3

**Type:**Volume - Performance

**Preconditions:**

1. The system is pre-loaded to contain X client files and X appointments
2. The system contains events of type food in every month of 2012 with appoints to them
3. The user is logged in as administrator

**Description:**

1. The administrator selects the Statistical report option in the admin panel
2. The administrator inputs a start date: 01-01-2012 and event type: Food
3. The administrator checks the option to generate a yearly report
4. The administrator attempts to create a report

**Expected Results:**

1. The report creation form is displayed
2. The system updates the form to contain the date
3. The system updated the form to show the checkbox checked
4. The system generated and downloads an excel type report

**Repetitions:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clients | 10000 | 15000 | 20000 | 25000 |
| Appointments | 10000 | 15000 | 20000 | 25000 |

## TCV-02 Volume Test Generate Report (PDF)

**Test Case ID:** TCV-02

**Title:** Volume Test Generate Report

**Requirement:**UC5.2

**Type:**Volume - Performance

**Preconditions:**

1. The system is pre-loaded to contain X client files and X appointments
2. The system contains events of type food in every month of 2012 with appoints to them
3. The user is logged in as administrator

**Description:**

1. The administrator selects the Statistical report option in the admin panel
2. The administrator inputs a start date: 2011-04-01 end date: 2013-04-25 event type: Food
3. The administrator attempts to create a report
4. The administrator inputs a start date: 2012-03-01 end date: 2012-04-03 event type: Food
5. The administrator attempts to create a report

**Expected Results:**

1. The report creation form appears
2. The system updates the form to contain the information
3. The system generated and redirects to a PDF type report
4. The system updates the form to contain the information
5. The system generated and redirects to a PDF type report

**Repetitions:**

|  |  |  |  |
| --- | --- | --- | --- |
| Clients | 10000 | 15000 | 20000 |
| Appointments | 10000 | 15000 | 20000 |

## TCV-03 Volume Test Client Search

**Test Case ID:** TCV-03

**Title:** Volume Test Generate Report

**Requirement:**UC1.7 – UC1.8

**Type:**Volume - Performance

**Preconditions:**

1. The user is logged in
2. The system is pre-loaded to contain X client files

**Description:**

1. The user inputs a single character search ex: 0
2. The user selects to search client files

**Expected Results:**

1. The search bar contains the search string
2. X results are returned in under 5 seconds

**Repetitions:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Clients | 10000 | 15000 | 20000 | 25000 | 100000 |
| Results | 100 | 1000 | 10000 | To be repeated for every repletion of client file amounts |  |

# Usability Test Case

## TCU-01 Usability Test Client File, Search, Events & Admin Panel

**Test Case ID:** TCV-01

**Title:** Volume Test Generate Yearly Report

**Requirement:**Completion of all user stories.

**Type:** Usability

**Preconditions:**

* 1. The user name and password are in the database.

1. Client File fields are loaded with countries, languages etc.…

Description:

1. Login to the system:

Username: admin

Password admin

1. Create a client file for the following person:

Name: Linda Mayer

Medicare: MAYL91551505

Address: 56 Fake Street, Kirkland, QC. H9H5G8,

Country Origin: Canada

Citizenship: Citizen

Phone Number” 514 555-9999

Mother Tongue: Hebrew

Spoken: English

Marital Status: Divorced

Work Status: Full Time

Referral: None

Income Type: Welfare

Amount: 200$ Monthly

1. Create an event template for the following event.

Name: Toy Drive

Appointment Needed: Yes

Appointments taken by: Person and by Hour

Duration: 1 Day

Appointment time slot lengths: 2 Hours

Repeatable: No

Opening Hours: 9:00 to 17:00

1. Create an occurrence of the following event:

Event Type: Toy Drive

Start Date: March 30th 2012

End Date: March 30th 2012

Max Capacity: 50 people/time slot

1. Search and open Linda Mayer’s file.
2. Make an appointment to the toy drive for Linda Mayer from 11:00 to 13:00.
3. Logout of the system.

**Expected Results:**

1. The system logs the user in and all system functions are made available.
2. The system creates the client file successfully, which the system indicates with the appearance of a green success message.
3. The system creates the event template successfully, which the system indicates with the appearance of a green success message.
4. The system creates the event occurrence successfully, which the system indicates with the appearance of a green success message.
5. The system outputs a series of search results matching the search criteria entered by the user. Linda Mayer appears first on the list of results.
6. The system makes the appointment successfully, which the system indicates with an update to the client file. Meaning the appointment appears in a list of all appointments currently booked to this client.
7. The system logs the user out of the system. The user no longer has access to system functionality.

|  |
| --- |
| **F.S.T.S.** |

|  |
| --- |
| **Test Results Reports** |

# Test Results Report

|  |
| --- |
| **Version 6.11** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev.** | **Description** | **Author(s)** |
| 2012-02-08 | 2.0 | Set up document | Katrina Anderson |
| 2012-02-14 | 2.1 | Included Acceptance and Unit Testing Sections | Katrina Anderson |
| 2012-02-18 | 2.2 | Document Revision | Cynthia Donato |
| 2012-02-24 | 3.3 | Test Name Revision | Adrian Lloyd |
| 2012-02-25 | 3.4 | Iteration 3 Test Report | Katrina Anderson |
| 2012-03-10 | 4.5 | Iteration 4 Test Report System Tests | Katrina Anderson |
| 2012-03-11 | 4.6 | Iteration 4 Test Report Acceptance and Unit testing | Katrina Anderson |
| 2012-03-22 | 5.7 | Iteration 5 Test Report Acceptance, Unit, Integration, System and UI. | Katrina Anderson |
| 2012-03-24 | 5.8 | Defect Cases | Katrina Anderson |
| 2012-04-10 | 6.9 | Volume Test Results | Patrick Modafferi |
| 2012-03-22 | 6.10 | Iteration 6 Test Report Acceptance, Unit, Integration, System, UI, Usability & Defects | Katrina Anderson |
| 2012-04-10 | 6.11 | Updated Results and final review | Patrick Modafferi |

# ****Iteration 2 Test Results Report****

## ****Introduction****

**This test result report contains the testing results for the F.S.T.S system. It will be comprised of 4 sections; Acceptance, Unit, Integration and System, which are outlined in the Test Plan (Included in the main deliverable).**

## ****Acceptance Testing****

### ****Iteration 2 Acceptance Test Results****

**Iteration 2 acceptance tests were conducted on February 13th 2012, during our class demo. 27 acceptance tests were conducted, all of which passed. Accompanying acceptance test cases can be found in the Test Cases section of the iteration 2 deliverable.**

### ****Acceptance Test Table & Status****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID & Title** | **Iteration** | **Change Date** | **Type** | **Status** |
| TC06-1 Test User Login | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-2 Test User Login – Omitted Password Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-3 Test User Login – Invalid Password | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-4 Test User Login – Invalid Username | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-5 Test User Login – Omitted Username Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-1 Test Create Client File - Minimal information | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-2 Test Create Client File - All information | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-3 Test Create Client File - Missing Require Fields | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-4 Test Create Client File – Address Exists | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC17-1 Test Update Client File – Main Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC18-1 Test Client File Set To Active | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC21-1Test Add New Flag - Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-2Test Add New Flag - Failure Scenario 1 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-3 Test Add New Flag – Failure Scenario 2 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC22-1 Test Delete Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-1 Test Update Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-2 Test Update Flag – Failure Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-3 Test Update Flag to Inactive – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-4 Test Update Flag to Active – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC24-1 Test Set Flag on New Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC24-2 Test Set Flag on Existing Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC25-1 Test Remove Flag on Existing Client File – Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Client File - Dependent Association | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Create Dependents– All Information Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Create Dependent – Missing Required Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC30-1 Test Dependent Deletion | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC31-1 Test Dependent Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-3 Test Postal Code/City Association – with Region Mapped Valid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-4 Test Postal Code/City Association – with Invalid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-4 Test Create Country Field - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-5 Test Create Language Field - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-3 Test Updating Client File Field - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-4 Test Updating Client File Field - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC35-3 Test Deleting Entry in Country Field | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC37-2 Test Search Client Files by Client ID | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC38-10 Test Search Client Files by Last Name and Medicare Number | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC39-1 Test Case to Test Event Template Creation – No Appointment and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-2 Test Case to Test Event Template Creation – Appointment Needed and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-3 Test Case to Test Event Template Creation – Appointment Needed and Event Repeats | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-4 Test Case to Test Event Template Creation – No Appointment Needed and Event | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-5 Test Case to Test Event Template Creation – Mandatory Field Omitted | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC40-1 Test Case to Test Event Template Deletion | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-1 Test Case to Test Updating Event Templates – Successful Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-2 Test Case to Test Updating Event Templates – Missing Mandatory Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-1 Test Case to Test Display of a List of Event Templates – Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-2 Test Case to Test Display of Event Template Details– Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC43-5 Test Event Occurrence Creation– No Appointment | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-6 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-7 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-8 Test Event Occurrence Creation–Mandatory Field | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC44-2 Test Event Occurrence Cancellation | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-3 Test Event Occurrence Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-4 Test Event Occurrence Update – Mandatory Field Omitted | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC46-1 Test Display List of Active Event Occurrences | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC47-1 Test Display of Event Occurrence Details | Sprint 2 | 2/13/2012 | Acceptance | Pass |

## ****Unit Testing****

### ****Iteration 2 Unit Test Results****

**In Iteration 2 we used PHPUnit to create a unit test suite comprised of 28 unit tests. Figure 1.3.1 illustrates the report generated by PHPUnit; As of February 12th 2012, all 28 unit tests passed successfully in 11.629 seconds. To test overall unit test coverage we used XDebug, which measures coverage by counting the statements executed, versus all statements, in a project. Figure 1.3.2 illustrates the report generated by XDebug, which indicates the percent statement coverage of our unit tests to be 70.55%, surpassing our goal of 60%.**

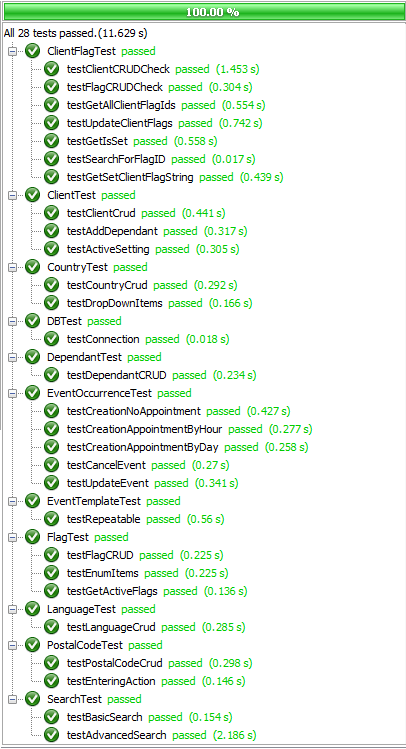
****

Figure 1.3.1 Iteration 2 Unit Test Results

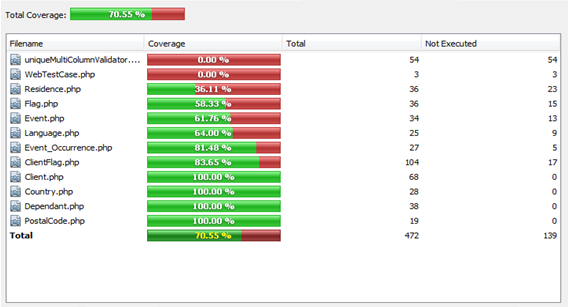


Figure 1.3.2 Iteration 2 Unit Test Coverage

### ****Unit Test Table & Status****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Title** | **Associated Method** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC18-2 Unit Test Set Client File to Inactive | ClientTest::testActiveSetting() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC18-3 Unit Test Set Client File to Active | ClientTest::testActiveSetting() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC24-3 Unit Test Client Flag (Part 1/2) | TestFlag::testFlagCRUD | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC25-2 Unit Test Client Flag (Part 2/2) | TestFlag::testFlagCRUD | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC29-4 Unit Test Create Dependent | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC29-5 Unit Test Add Dependent to Client | ClientTest::testAddDependants() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC30-2 Unit Test Delete Dependents | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC31-2 Unit Test Update Dependant | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC32-1 Unit Test Create-Update-Delete Postal Code | PostalTestCode::testPostalCodeCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC32-2 Unit Test Ajax Request Passes to the Client Controller | PostalTestCode::testEnteringAction() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-1 Unit Test for drop down lists | CountryTest::testDropDownItems() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-2 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-3 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC34-1 Unit Test Update Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC34-2 Unit Test Update Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC35-1 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC35-2 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC37-1 Unit Test Search by Client ID | SearchTest::testBasicSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-1 Unit Test Search by Name | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-2 Unit Test Search by Medicare | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-3 Unit Test Search by Address | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-4 Unit Test Search by Street | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-5 Unit Test Search by Apartment | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-6 Unit Test Search by City | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-7 Unit Test Search by Province | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-8 Unit Test Search by Postal Code | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-1 Unit Test Event Occurrence Creation - No Appointment | EventOccurrenceTest::testCreationNoAppointment | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-2 Unit Test Event Occurrence Creation - Appointment By Hour | EventOccurrenceTest::testCreationAppointmentByHour() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-3 Unit Test Event Occurrence Creation - Appointment By Day | EventOccurrenceTest::testCreationAppointmentByDay() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC44-1 Unit Test Event Occurrence Cancellation | EventOccurrenceTest::testCancelEvent() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC45-1 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC45-2 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 2/12/2012 | Unit | Pass |

## ****Integration Testing****

Will be included as of Iteration 3

## ****System Testing****

Will be included as of Iteration 3

# ****Iteration 3 Test Results Report****

## ****Introduction****

**This test result report contains the testing results for the F.S.T.S system. It will be comprised of 4 sections; Acceptance, Unit, Integration and System, which are outlined in the Test Plan.**

## ****Acceptance Testing****

### ****Iteration 3 Acceptance Test Results****

**Iteration 3 acceptance tests are to be conducted on February 27th 2012, during our class demo. Up to 16 acceptance tests will be conducted, time permitting. Accompanying acceptance test cases can be found in the Test Cases section of the Testing document.**

### ****Acceptance Test Table & Status****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID & Title** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-2 Test Create User | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC02-2 Test Update User | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC03-2 Test Delete User | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC04-2 Test Set User to Administrator | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC05-1 Test View Manage Users | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC06-1 Test User Login | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-2 Test User Login – Omitted Password Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-3 Test User Login – Invalid Password | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-4 Test User Login – Invalid Username | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-5 Test User Login – Omitted Username Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC08-1 Test View All clients Coming to an Event - Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC10-1 Test Appointment Creation – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC11-1 Test Appointment Update – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC12-1 Test Appointment Cancellation – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC13-1 Test Appointment Validation – Capacity Exceeded | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC14-1 Test View Summary of Clients with Appointments to Events | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC16-1 Create a user file with minimal information | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-2 Create complete user file | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-3 Missing Require Fields | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-4 Attempt to enter an existing address | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC17-1 Check that edited record is properly updated in the database | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC18-1 Test Client File Set To Active | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC21-1: Add New Flag - Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-2: Add New Flag - Failure Scenario 1 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-3: Add New Flag – Failure Scenario 2 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC22-1: Delete Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-1: Update Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-2: Update Flag – Failure Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-3: Update Flag to Inactive – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-4: Update Flag to Active – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC24-1 Test Set Flag on New Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC24-2: Test Set Flag on Existing Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC25-1: Test Remove Flag on Existing Client File – Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC26-1 Test Set Note Text on New Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC26-2 Test Set Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC27-1 Test Remove Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC28-1 Test Update Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | TBD |
| TC29-1 Test Client File Dependent Association | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - All Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - Missing Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC30-1 Test Dependent Deletion | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC31-1 Test Dependent Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-3 Test Postal Code/City Association – with Region Mapped Valid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-4 Test Postal Code/City Association – with Invalid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-4 Test case for creating a new country - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-5 Test case for creating a new language - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-3 Test case for editing a row that was mispelled - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-4 Test case for changing a row - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC35-3 Test Deleting Country from drop down lists | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC37-2 Test Search Client Files by Client ID | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC38-10 Test Search Client Files by Last Name and Medicare Number | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC39-1 Test Case to Test Event Template Creation – No Appointment and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-2 Test Case to Test Event Template Creation – Appointment Needed and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-3 Test Case to Test Event Template Creation – Appointment Needed and Event Repeats | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-4 Test Case to Test Event Template Creation – No Appointment Needed and Event | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-5 Test Case to Test Event Template Creation – Mandatory Field Omitted | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC40-1 Test Case to Test Event Template Deletion | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-1 Test Case to Test Updating Event Templates – Successful Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-2 Test Case to Test Updating Event Templates – Missing Mandatory Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-1 Test Case to Test Display of a List of Event Templates – Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-2 Test Case to Test Display of Event Template Details– Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC43-5 Test Event Occurrence Creation– No Appointment | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-6 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-7 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-8 Test Event Occurrence Creation–Mandatory Field | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC44-2 Test Event Occurrence Cancellation | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-3 Test Event Occurrence Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-4 Test Event Occurrence Update – Mandatory Field Omitted | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC46-1 Test Display List of Active Event Occurrences | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC47-1 Test Display of Event Occurrence Details | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TCD01-02 Test for creation of correct times for time fields | Sprint 3 | 2/27/2012 | Acceptance | TBD |

## ****Unit Testing****

### ****Iteration 3 Unit Test Results****

**In Iteration 3 we used PHPUnit to create a unit test suite comprised of 30 unit tests. Figure 2.3.1 illustrates the report generated by PHPUnit; As of February 25th 2012, all 30 unit tests passed successfully in 10.784s seconds. To test overall unit test coverage we used XDebug, which measures coverage by counting the statements executed, versus all statements, in a project. Figure 2.3.2 illustrates the report generated by XDebug, which indicates the percent statement coverage of our unit tests to be 63.46%, surpassing our goal of 60%.**

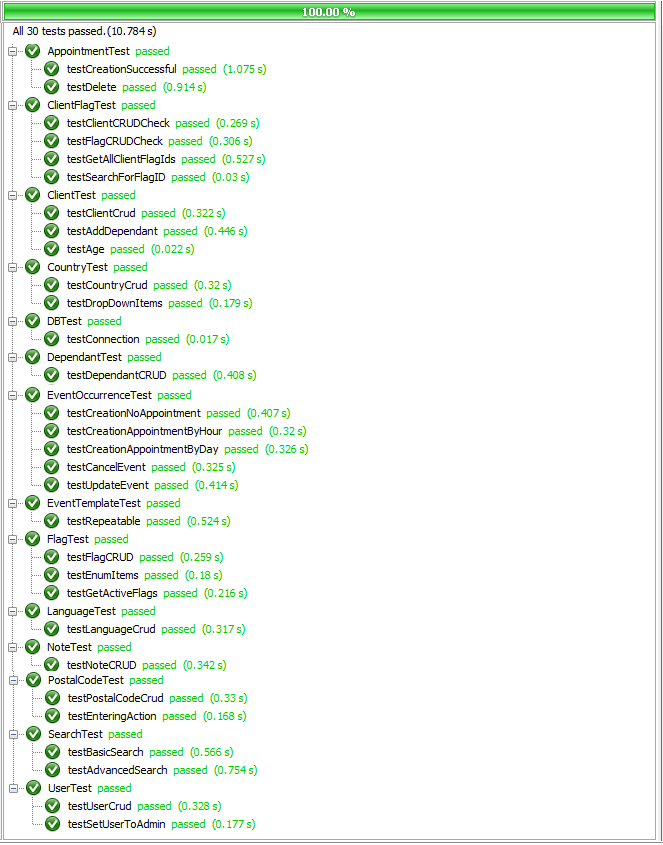


Figure 2.3.1 Iteration 3 Unit Test Results

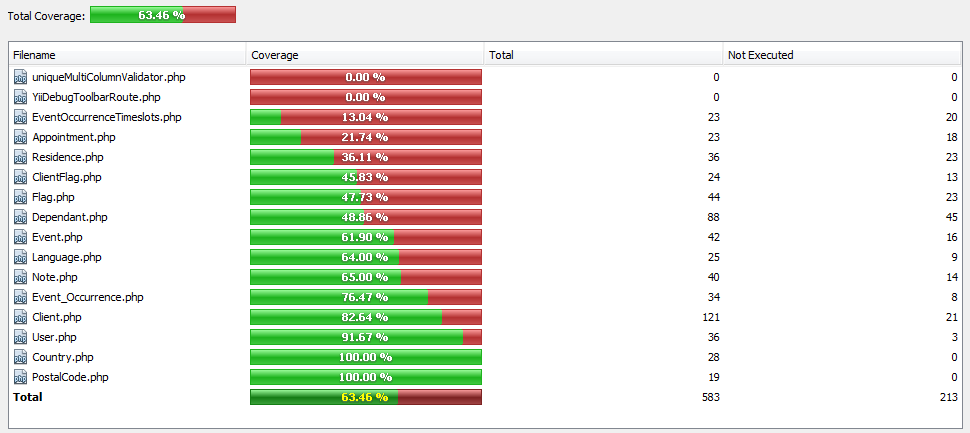


Figure 2.3.2 Iteration 3 Unit Test Coverage

### ****Unit Test Table & Status****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Title** | **Associated Method** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-1 Unit Test Create User | UserTest::testUserCrud() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC02-1 Unit Test Update User | UserTest::testUserCrud() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC03-1 Unit Test Delete User | UserTest::testUserCrud() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC04-1 Unit Test Set User to Administrator | UserTest::testUserToAdmin() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC10-2 Unit Test Appointment Creation - Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC11-2 Unit Test Appointment Update – Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC12-2 Unit Test Appointment Cancellation – Success Scenario | AppointmentTest::testDelete() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC13-2 Unit Test Appointment Validation – Capacity Exceeded | AppointmentTest::testCreationSuccessful() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC18-2 Unit Test Set Client File to Inactive | ClientTest::testActiveSetting() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC18-3 Unit Test Set Client File to Active | ClientTest::testActiveSetting() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC24-3 Unit Test Client Flag (Part 1/2) | TestFlag::testFlagCRUD | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC25-2 Unit Test Client Flag (Part 2/2) | TestFlag::testFlagCRUD | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC26-3 Unit Test Note - Add Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC27-2 Unit Test Note - Delete Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC28-2 Unit Test Note - Update Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 2/25/2012 | Unit | Pass |
| TC29-4 Unit Test Create Dependent | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC29-5 Unit Test Add Dependent to Client | ClientTest::testAddDependants() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC30-2 Unit Test Delete Dependents | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC31-2 Unit Test Update Dependant | DependantTest::testDependantCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC32-1 Unit Test Create-Update-Delete Postal Code | PostalTestCode::testPostalCodeCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC32-2 Unit Test Ajax Request Passes to the Client Controller | PostalTestCode::testEnteringAction() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-1 Unit Test for drop down lists | CountryTest::testDropDownItems() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-2 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC33-3 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC34-1 Unit Test Update Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC34-2 Unit Test Update Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC35-1 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC35-2 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC37-1 Unit Test Search by Client ID | SearchTest::testBasicSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-1 Unit Test Search by Name | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-2 Unit Test Search by Medicare | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-3 Unit Test Search by Address | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-4 Unit Test Search by Street | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-5 Unit Test Search by Apartment | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-6 Unit Test Search by City | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-7 Unit Test Search by Province | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC38-8 Unit Test Search by Postal Code | SearchTest::testAdvancedSearch() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-1 Unit Test Event Occurrence Creation - No Appointment | EventOccurrenceTest::testCreationNoAppointment | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-2 Unit Test Event Occurrence Creation - Appointment By Hour | EventOccurrenceTest::testCreationAppointmentByHour() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC43-3 Unit Test Event Occurrence Creation - Appointment By Day | EventOccurrenceTest::testCreationAppointmentByDay() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC44-1 Unit Test Event Occurrence Cancellation | EventOccurrenceTest::testCancelEvent() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TC45-1 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 2/12/2012 | Unit | Pass |
| TCD01-01 Unit test for Medicare algorithm | ClientTest::testAge() | Sprint 3 | 2/23/2012 | Unit | Pass |

## ****Integration Testing****

See Master Test Plan 4.2.3 Integration Testing

## ****System Testing****

### ****Iteration 3System Test Results****

**In Iteration 3 we used Selenium to create a system test that covered 100% of F.S.T.S. functionalities up until February 25th 2012. Figure 2.5.1. illustrates the Selenium system test.**

**

Figure 2.5.1 Iteration 3System Test Results

### ****System Test Table & Status****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Title** | **Associated Method** | **Path** | **Sprint** | **Change Date** | **Type** | **Status** |
| STC01 | System Test Administrative Components | AdministrationComponents:: testAdministrationComponents() | 1, 27 | Sprint 3 | 25/02/2012 | System | Pass |
| STC02 | System Test Appointment Management | AppointmentManagement Components::testAppointment Components() | 1, 17, 23, 25.1,  25.2, 27 | Sprint 3 | 25/02/2012 | System | Pass |
| STC03 | System Test Client File Management | ClientManagementComponents:: testClientManagement Components() | 1, 18.1, 18.3,  19.1, 19.3, 19.2,  20.1, 20.3, 20.2,  21.1, 21.2, 22.1,  22.2, 18.2, 27 | Sprint 3 | 25/02/2012 | System | Pass |
| STC04 | System Test Search Components | ClientSearchComponents:: testClientSearch() | 1, 17, 27 | Sprint 3 | 25/02/2012 | System | Pass |
| STC05 | System Test Event Management | EventManagementComponents:: testEventManagement() | 1, 2, 15, 24.1, 24.3,  24.2, 27 | Sprint 3 | 25/02/2012 | System | Pass |

# ****Iteration 4 Test Results Report****

## ****Introduction****

**This test result report contains the testing results for the F.S.T.S system. It will be comprised of 4 sections; Acceptance, Unit, Integration and System, which are outlined in the Test Plan.**

## ****Acceptance Testing****

### ****Iteration 4 Acceptance Test Results****

**Iteration 4 acceptance tests are to be conducted on March 12th 2012, during our class demo. Up to 12acceptance tests will be conducted, time permitting. Accompanying acceptance test cases can be found in the Test Cases section of the testing document.**

### ****Acceptance Test Table & Status****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID & Title** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-2 Test Create User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC02-2 Test Update User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC03-2 Test Delete User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC04-2 Test Set User to Administrator | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC05-1 Test View Manage Users | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC06-1 Test User Login | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-2 Test User Login – Omitted Password Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-3 Test User Login – Invalid Password | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-4 Test User Login – Invalid Username | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-5 Test User Login – Omitted Username Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC07-1 Test View Check-in For All Clients Coming to an Event - Successful | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC08-1 Test View All clients Coming to an Event - Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC09-1 Test Closing an Event – Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC10-1 Test Appointment Creation – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC11-1 Test Appointment Update – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC12-1 Test Appointment Cancellation – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC13-1 Test Appointment Validation – Capacity Exceeded | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC14-1 Test View Summary of Clients with Appointments to Events | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC16-1 Create a user file with minimal information | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-2 Create complete user file | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-3 Missing Require Fields | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-4 Attempt to enter an existing address | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC17-1 Check that edited record is properly updated in the database | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC18-1 Test Client File Set To Active | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC19-1 Test Set Inactive Client Files Script Execution – Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC20-1 Test Set Archive Client Files Script Execution – Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC21-1: Add New Flag - Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC21-2: Add New Flag - Failure Scenario 1 | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC21-3: Add New Flag – Failure Scenario 2 | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC22-1: Delete Flag – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-1: Update Flag – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-2: Update Flag – Failure Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-3: Update Flag to Inactive – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-4: Update Flag to Active – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC24-1 Test Set Flag on New Client File - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC24-2: Test Set Flag on Existing Client File - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC25-1: Test Remove Flag on Existing Client File – Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC26-1 Test Set Note Text on New Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC26-2 Test Set Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC27-1 Test Remove Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC28-1 Test Update Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Association | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - All Fields | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - Missing Fields | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC30-1 Test Dependent Deletion | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC31-1 Test Dependent Update | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC32-3 Test Postal Code/City Association – with Region Mapped Valid Postal Code | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC32-4 Test Postal Code/City Association – with Invalid Postal Code | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC33-4 Test case for creating a new country - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC33-5 Test case for creating a new language - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC34-3 Test case for editing a row that was misspelled - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC34-4 Test case for changing a row - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC35-3 Test Deleting Country from drop down lists | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC37-2 Test Search Client Files by Client ID | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC38-9 Test Search Client Files by Last Name | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC38-10 Test Search Client Files by Last Name and Medicare Number | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC38-9 Test Search Client Files by Last Name | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC39-1 Test Case to Test Event Template Creation – No Appointment and No Repetition | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-2 Test Case to Test Event Template Creation – Appointment Needed and No Repetition | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-3 Test Case to Test Event Template Creation – Appointment Needed and Event Repeats | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-4 Test Case to Test Event Template Creation – No Appointment Needed and Event | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-5 Test Case to Test Event Template Creation – Mandatory Field Omitted | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC40-1 Test Case to Test Event Template Deletion | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC41-1 Test Case to Test Updating Event Templates – Successful Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC41-2 Test Case to Test Updating Event Templates – Missing Mandatory Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC42-1 Test Case to Test Display of a List of Event Templates – Successful Creation | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC42-2 Test Case to Test Display of Event Template Details– Successful Creation | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC43-5 Test Event Occurrence Creation– No Appointment | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-6 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-7 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-8 Test Event Occurrence Creation–Mandatory Field | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC44-2 Test Event Occurrence Cancellation | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC45-3 Test Event Occurrence Update | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC45-4 Test Event Occurrence Update – Mandatory Field Omitted | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC46-1 Test Display List of Active Event Occurrences | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC47-1 Test Display of Event Occurrence Details | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC51-2 Test Income Source Creation | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC52-2 Test Deleting Income Source | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC53-2 Test Income Source Update | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TC54-2 Test Income Source View | Sprint 4 | 12/03/2012 | Acceptance | TBD |
| TCD01-02 Test for creation of correct times for time fields | Sprint 3 | 27/02/2012 | Acceptance | Pass |

## ****Unit Testing****

### ****Iteration 4 Unit Test Results****

**In Iteration 4 we used PHPUnit to create a unit test suite comprised of 34 unit tests. Figure 3.3.2 illustrates the report generated by PHPUnit; As of March 10th2012, all 34 unit tests passed successfully in 14.059 seconds. To test overall unit test coverage we used XDebug, which measures coverage by counting the statements executed, versus all statements, in a project. Figure 3.3.1 illustrates the report generated by XDebug, which indicates the percent statement coverage of our unit tests to be 66.06%, surpassing our goal of 60%.**

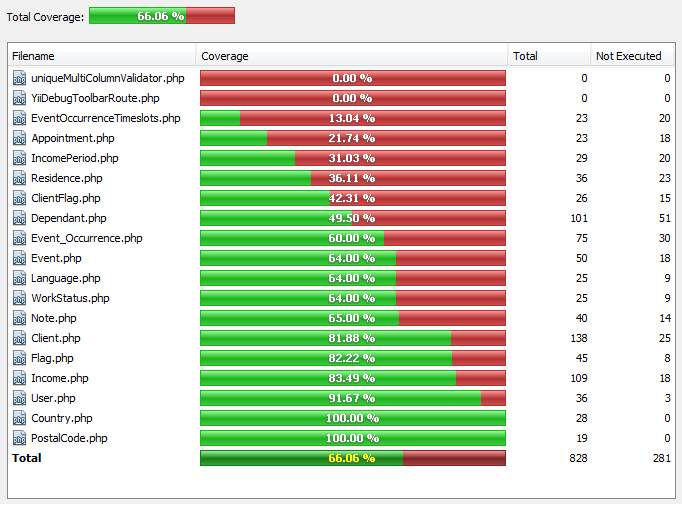


Figure 3.3.1 Iteration 4 Unit Test Coverage

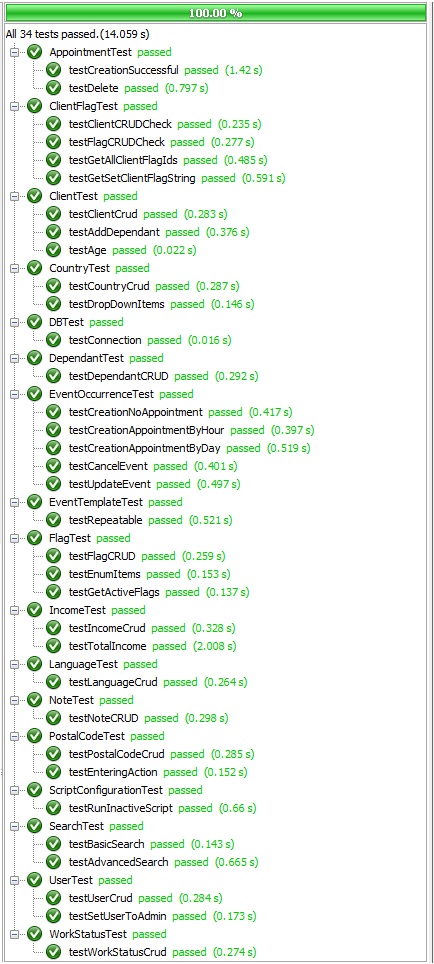


Figure 3.3.2 Iteration 4 Unit Test Results

### ****Unit Test Table & Status****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Title** | **Associated Method** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-1 Unit Test Create User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC02-1 Unit Test Update User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC03-1 Unit Test Delete User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC04-1 Unit Test Set User to Administrator | UserTest::testUserToAdmin() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC10-2 Unit Test Appointment Creation - Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC11-2 Unit Test Appointment Update – Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC12-2 Unit Test Appointment Cancellation – Success Scenario | AppointmentTest::testDelete() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC13-2 Unit Test Appointment Validation – Capacity Exceeded | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC18-2 Unit Test Set Client File to Inactive | ClientTest::testActiveSetting() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC18-3 Unit Test Set Client File to Active | ClientTest::testActiveSetting() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC19-2 Unit Test Set Inactive Script Functionality | ScriptConfigurationTest::testRunInactiveScript() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC20-2 Unit Test Archive Client Files Functionality | ScriptConfigurationTest::testRunArchiveScript() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC24-3 Unit Test Client Flag (Part 1/2) | TestFlag::testFlagCRUD() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC25-2 Unit Test Client Flag (Part 2/2) | TestFlag::testFlagCRUD() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC26-3 Unit Test Note - Add Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC27-2 Unit Test Note - Delete Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC28-2 Unit Test Note - Update Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC29-4 Unit Test Create Dependent | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC29-5 Unit Test Add Dependent to Client | ClientTest::testAddDependants() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC30-2 Unit Test Delete Dependents | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC31-2 Unit Test Update Dependant | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC32-1 Unit Test Create-Update-Delete Postal Code | PostalTestCode::testPostalCodeCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC32-2 Unit Test Ajax Request Passes to the Client Controller | PostalTestCode::testEnteringAction() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-1 Unit Test for drop down lists | CountryTest::testDropDownItems() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-2 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-3 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC34-1 Unit Test Update Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC34-2 Unit Test Update Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC35-1 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC35-2 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC37-1 Unit Test Search by Client ID | SearchTest::testBasicSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-1 Unit Test Search by Name | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-2 Unit Test Search by Medicare | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-3 Unit Test Search by Address | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-4 Unit Test Search by Street | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-5 Unit Test Search by Apartment | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-6 Unit Test Search by City | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-7 Unit Test Search by Province | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-8 Unit Test Search by Postal Code | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC43-1 Unit Test Event Occurrence Creation - No Appointment | EventOccurrenceTest::testCreationNoAppointment | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC43-2 Unit Test Event Occurrence Creation - Appointment By Hour | EventOccurrenceTest::testCreationAppointmentByHour() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC43-3 Unit Test Event Occurrence Creation - Appointment By Day | EventOccurrenceTest::testCreationAppointmentByDay() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC44-1 Unit Test Event Occurrence Cancellation | EventOccurrenceTest::testCancelEvent() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC45-1 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC51-1 Unit Test Income Source Creation | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC52-1 Unit Test Income Source Deletion | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC53-1 Unit Test Income Source Update | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC54-1 Unit Test Income Source View | IncomeTest::testTotalIncome() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TCD01-01 Unit test for Medicare algorithm | ClientTest::testAge() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TCD07-1 Unit Test for Work Status | WorkStatusTest::testWorkStatusCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |

## ****Integration Testing****

See Master Test Plan 4.2.3 Integration Testing

## ****System Testing****

### ****Iteration 4System Test Results****

**In Iteration 4 we used Selenium to create a system test that covered 6 out of 10 feature groups for the F.S.T.S, essentially the feature groups that have been executed up until March 12th 2012. Figure 2.5.1. illustrates the Selenium system test.**

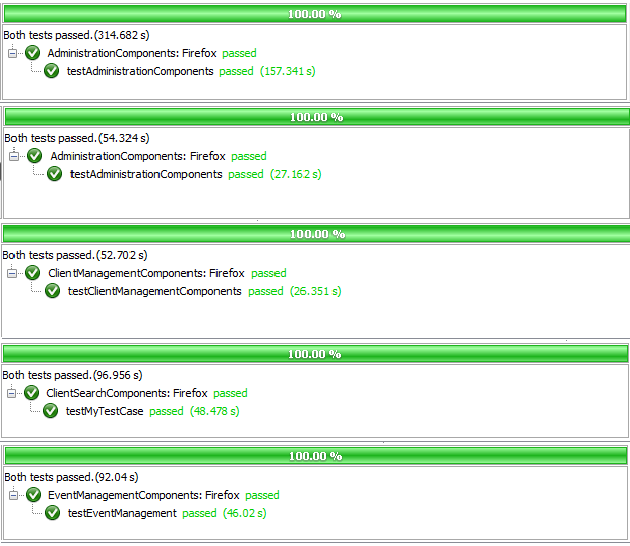
**

Figure 3.5.1 Iteration 4System Test Results

### ****System Test Table & Status****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Title** | **Associated Method** | **Path** | **Sprint** | **Change Date** | **Type** | **Status** |
| STC01 | System Test Administrative Components | AdministrationComponents:: testAdministrationComponents() | 1, 2, 3, 2, 4, 2, 5, 2, 6, 2, 7, 2, 8, 2, 9,  2, 10, 2, 11, 2, 12 ,2, 13,  2, 14, 2, 16, 27 | Sprint 4 | 03/10/2012 | System | Pass |
| STC02 | System Test Client File Management | ClientManagementComponents::testClientManagement Components() | 1, 18.1, 18.3,  19.1, 19.3, 19.2,  20.1, 20.3, 20.2,  21.1, 21.2, 22.1,  22.2, 18.2, 27 | Sprint 4 | 03/10/2012 | System | Pass |
| STC03 | System Test Event Management | EventManagementComponents::testEventManagement() | 1, 2, 15, 24.1,  24.3, 24.2, 27 | Sprint 4 | 03/10/2012 | System | Pass |
| STC04 | System Test Appointment Management | AppointmentManagement  Components::testAppointment  Components() | 1, 17, 23, 25.1,  25.2, 27 | Sprint 4 | 03/10/2012 | System | Pass |
| STC05 | System Test Search Components | ClientSearchComponents::  testClientSearch() | 1, 17, 27 | Sprint 4 | 03/10/2012 | System | Pass |

# ****Iteration 5 Test Results Report****

## ****Introduction****

**This test result report contains the testing results for the F.S.T.S. It will be comprised of 6 sections; Acceptance, Unit, Integration, System, UI and Defects, which are outlined in the Test Plan (Included in the main deliverable).**

## ****Acceptance Testing****

### ****Iteration 5 Acceptance Test Results****

**Iteration 5 acceptance tests are to be conducted on March 26th 2012, during our class demo. Up to 11 acceptance tests will be conducted, time permitting. Accompanying acceptance test cases can be found in the Test Cases section of the iteration testing document.**

### ****Acceptance Test Table & Status****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID & Title** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-2 Test Create User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC02-2 Test Update User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC03-2 Test Delete User | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC04-2 Test Set User to Administrator | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC05-1 Test View Manage Users | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC06-1 Test User Login | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-2 Test User Login – Omitted Password Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-3 Test User Login – Invalid Password | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-4 Test User Login – Invalid Username | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC06-5 Test User Login – Omitted Username Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC07-1 Test View Check-in For All Clients Coming to an Event - Successful | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC08-1 Test View All clients Coming to an Event - Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC09-1 Test Closing an Event – Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC10-1 Test Appointment Creation – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC11-1 Test Appointment Update – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC12-1 Test Appointment Cancellation – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC13-1 Test Appointment Validation – Capacity Exceeded | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC14-1 Test View Summary of Clients with Appointments to Events | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC16-1 Create a user file with minimal information | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-2 Create complete user file | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-3 Missing Require Fields | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC16-4 Attempt to enter an existing address | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC17-1 Check that edited record is properly updated in the database | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC18-1 Test Client File Set To Active | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC19-1 Test Set Inactive Client Files Script Execution – Success Scenario | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC20-1 Test Set Archive Client Files Script Execution – Success Scenario \*Clients scoped out this functionality | Sprint 4 | 12/03/2012 | Acceptance | FAIL |
| TC21-1: Add New Flag - Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC21-2: Add New Flag - Failure Scenario 1 | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC21-3: Add New Flag – Failure Scenario 2 | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC22-1: Delete Flag – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-1: Update Flag – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-2: Update Flag – Failure Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-3: Update Flag to Inactive – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC23-4: Update Flag to Active – Success Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC24-1 Test Set Flag on New Client File - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC24-2: Test Set Flag on Existing Client File - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC25-1: Test Remove Flag on Existing Client File – Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC26-1 Test Set Note Text on New Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC26-2 Test Set Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC27-1 Test Remove Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC28-1 Test Update Note Text on Existing Client File – Success Scenario | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Association | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - All Fields | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - Missing Fields | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC30-1 Test Dependent Deletion | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC31-1 Test Dependent Update | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC32-3 Test Postal Code/City Association – with Region Mapped Valid Postal Code | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC32-4 Test Postal Code/City Association – with Invalid Postal Code | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC33-4 Test case for creating a new country - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC33-5 Test case for creating a new language - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC34-3 Test case for editing a row that was misspelled - Success Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC34-4 Test case for changing a row - Failure Scenario | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC35-3 Test Deleting Country from drop down lists | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC36-1 Test View Client File Foreign Key Mapping | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TC36-2 Test View Client File User Acceptance of General Layout | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TC37-2 Test Search Client Files by Client ID | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC38-10 Test Search Client Files by Last Name and Medicare Number | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC39-1 Test Case to Test Event Template Creation – No Appointment and No Repetition | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-2 Test Case to Test Event Template Creation – Appointment Needed and No Repetition | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-3 Test Case to Test Event Template Creation – Appointment Needed and Event Repeats | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-4 Test Case to Test Event Template Creation – No Appointment Needed and Event | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC39-5 Test Case to Test Event Template Creation – Mandatory Field Omitted | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC40-1 Test Case to Test Event Template Deletion | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC41-1 Test Case to Test Updating Event Templates – Successful Scenario | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC41-2 Test Case to Test Updating Event Templates – Missing Mandatory Field | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC42-1 Test Case to Test Display of a List of Event Templates – Successful Creation | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC42-2 Test Case to Test Display of Event Template Details– Successful Creation | Sprint 1 | 30/01/2012 | Acceptance | Pass |
| TC43-5 Test Event Occurrence Creation– No Appointment | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-6 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-7 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC43-8 Test Event Occurrence Creation–Mandatory Field | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC44-2 Test Event Occurrence Cancellation | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC45-3 Test Event Occurrence Update | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC45-4 Test Event Occurrence Update – Mandatory Field Omitted | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC46-1 Test Display List of Active Event Occurrences | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC47-1 Test Display of Event Occurrence Details | Sprint 2 | 13/02/2012 | Acceptance | Pass |
| TC48-1 Test Generate Operational Reports – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TC50-1 Test Generate Statistical Reports – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TC51-2 Test Income Source Creation | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC52-2 Test Deleting Income Source | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC53-2 Test Income Source Update | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC54-2 Test Income Source View | Sprint 4 | 12/03/2012 | Acceptance | Pass |
| TC55-1 Test Add Client to Waiting List – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TC55-2 Test Remove Client from Waiting List – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TCD01-02 Test for creation of correct times for time fields | Sprint 3 | 27/02/2012 | Acceptance | Pass |
| TCD20-1 Test Deleting an Event with No Associated Appointments – Successful | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TCD33-1 Test Create Appointment per Household Member – Successful | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TCD34-1 Test Set Event Inactive – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TCD34-2 Test Set Event Amount – Success Scenario | Sprint 5 | 26/03/2012 | Acceptance | TBD |
| TCD35-1 Test Closing an Event Sets the Last Visited Date for All Attendee | Sprint 5 | 26/03/2012 | Acceptance | TBD |

## ****Unit Testing****

### ****Iteration 5 Unit Test Results****

**In Iteration 5 we used PHPUnit to create a unit test suite comprised of 41 unit tests. Figure 4.3.2 illustrates the report generated by PHPUnit; As of March 22nd2012, all 41 unit tests passed successfully in 15.04 seconds. To test overall unit test coverage we used XDebug, which measures coverage by counting the statements executed, versus all statements, in a project. Figure 4.3.1 illustrates the report generated by XDebug, which indicates the percent statement coverage of our unit tests to be 60.69%, surpassing our goal of 60%.**

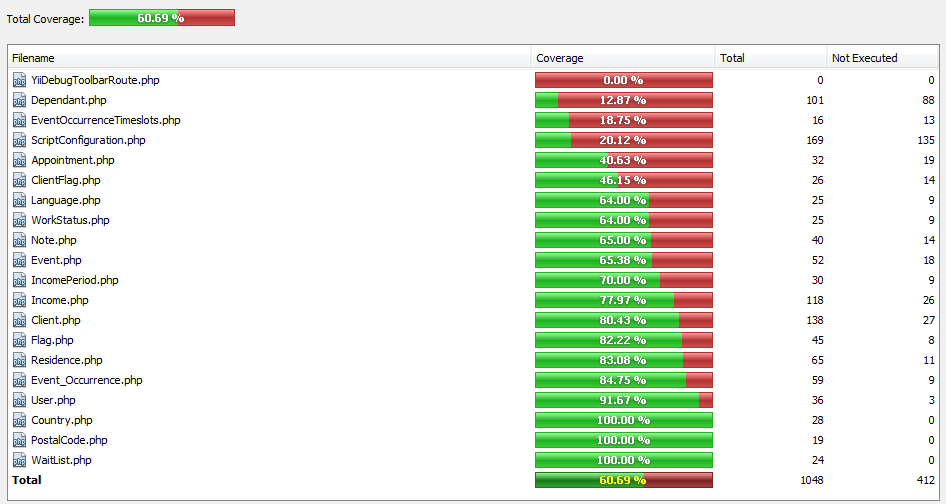


Figure 4.3.1 Iteration 5 Unit Test Coverage

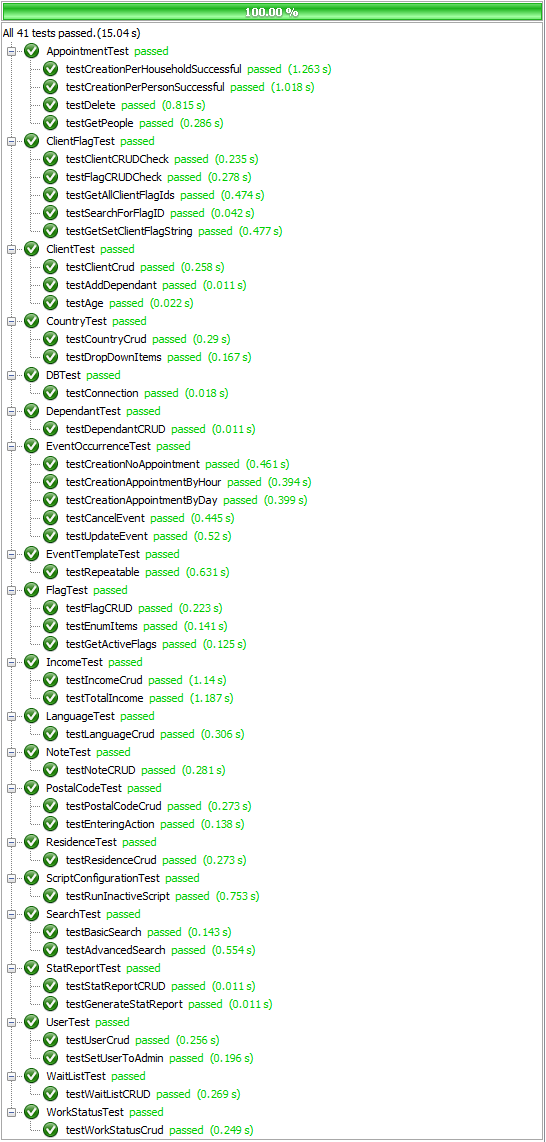


Figure 4.3.2 Iteration 5 Unit Test Results

### ****Unit Test Table & Status****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Title** | **Associated Method** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-1 Unit Test Create User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC02-1 Unit Test Update User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC03-1 Unit Test Delete User | UserTest::testUserCrud() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC04-1 Unit Test Set User to Administrator | UserTest::testUserToAdmin() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC10-2 Unit Test Appointment Creation - Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC11-2 Unit Test Appointment Update – Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC12-2 Unit Test Appointment Cancellation – Success Scenario | AppointmentTest::testDelete() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC13-2 Unit Test Appointment Validation – Capacity Exceeded | AppointmentTest::testCreationSuccessful() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC18-2 Unit Test Set Client File to Inactive | ClientTest::testActiveSetting() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC18-3 Unit Test Set Client File to Active | ClientTest::testActiveSetting() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC19-2 Unit Test Set Inactive Script Functionality | ScriptConfigurationTest::testRunInactiveScript() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC20-2 Unit Test Archive Client Files Functionality | ScriptConfigurationTest::testRunArchiveScript() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC24-3 Unit Test Client Flag (Part 1/2) | TestFlag::testFlagCRUD | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC25-2 Unit Test Client Flag (Part 2/2) | TestFlag::testFlagCRUD | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC26-3 Unit Test Note - Add Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC27-2 Unit Test Note - Delete Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC28-2 Unit Test Note - Update Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TC29-4 Unit Test Create Dependent | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC29-5 Unit Test Add Dependent to Client | ClientTest::testAddDependants() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC30-2 Unit Test Delete Dependents | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC31-2 Unit Test Update Dependant | DependantTest::testDependantCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC32-1 Unit Test Create-Update-Delete Postal Code | PostalTestCode::testPostalCodeCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC32-2 Unit Test Ajax Request Passes to the Client Controller | PostalTestCode::testEnteringAction() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-1 Unit Test for drop down lists | CountryTest::testDropDownItems() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-2 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC33-3 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC34-1 Unit Test Update Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC34-2 Unit Test Update Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC35-1 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC35-2 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC37-1 Unit Test Search by Client ID | SearchTest::testBasicSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-1 Unit Test Search by Name | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-2 Unit Test Search by Medicare | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-3 Unit Test Search by Address | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-4 Unit Test Search by Street | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-5 Unit Test Search by Apartment | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-6 Unit Test Search by City | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-7 Unit Test Search by Province | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC38-8 Unit Test Search by Postal Code | SearchTest::testAdvancedSearch() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC43-1 Unit Test Event Occurrence Creation - No Appointment | EventOccurrenceTest::testCreationNoAppointment() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC43-2 Unit Test Event Occurrence Creation - Appointment By Hour | EventOccurrenceTest::testCreationAppointmentByHour() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC43-3 Unit Test Event Occurrence Creation - Appointment By Day | EventOccurrenceTest::testCreationAppointmentByDay() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC44-1 Unit Test Event Occurrence Cancellation | EventOccurrenceTest::testCancelEvent | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC45-1 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 11/03/2012 | Unit | Pass |
| TC51-1 Unit Test Income Source Creation | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC52-1 Unit Test Income Source Deletion | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC53-1 Unit Test Income Source Update | IncomeTest::testIncomeCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC54-1 Unit Test Income Source View | IncomeTest::testTotalIncome() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TCD01-01 Unit test for Medicare algorithm | ClientTest::testAge() | Sprint 3 | 11/03/2012 | Unit | Pass |
| TCD07-1 Unit Test for Work Status | WorkStatusTest::testWorkStatusCrud() | Sprint 4 | 11/03/2012 | Unit | Pass |
| TC48-2 Unit Test Generate Operation Reports | StatReportTest::testStatReportCRUD() | Sprint 5 | 22/03/2012 | Unit | Pass |
| TC50-2 Unit Test Generate Statistical Reports | StatReportTest::testGenerateStatReport() | Sprint 5 | 22/03/2012 | Unit | Pass |
| TC55-2 Unit Test Waiting List | WaitListTest::testWaitListCRUD() | Sprint 5 | 22/03/2012 | Unit | Pass |
| TCD33-2 Unit Test Appointment Creation by Household | Appointment::testCreationPerHouselholdSuccessful() | Sprint 5 | 22/03/2012 | Unit | Pass |
| TCD33-3 Unit Test get people | Appointment::testGetPeople() | Sprint 5 | 22/03/2012 | Unit | Pass |

## ****Integration Testing****

See Master Test Plan 4.2.3 Integration Testing

## ****System Testing****

### ****Iteration 4System Test Results****

**In Iteration 5 we used Selenium to create system tests that covered 8 out of 10 feature groups for the F.S.T.S, essentially the feature groups that have been executed up until March 22nd 2012. Figure 4.5.1. illustrates the Selenium system test.**

**

Figure 4.5.1 Iteration 5System Test Results

### ****System Test Table & Status****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Title** | **Associated Method** | **Path** | **Sprint** | **Change Date** | **Type** | **Status** |
| STC01 | System Test Administrative Components | AdministrationComponents:: testAdministration Components() | 1, 2, 3, 2, 4, 2, 5, 2, 6, 2, 7, 2, 8, 2, 9,  2, 10, 2, 11, 2, 12 ,2, 13,  2, 14, 2, 16, 27 | Sprint 4 | 03/10/2012 | System | Pass |
| STC02 | System Test Client File Management | ClientManagement Components::testClient ManagementComponents() | 1, 18.1, 18.3,  19.1, 19.3, 19.2,  20.1, 20.3, 20.2,  21.1, 21.2, 22.1,  22.2, 18.2, 27 | Sprint 4, Sprint 5 | 03/10/2012 | System | Pass |
| STC03 | System Test Event Management | EventManagement Components::testEvent Management() | 1, 2, 15, 24.1,  24.3, 24.2, 27 | Sprint 4, Sprint 5 | 03/10/2012 | System | Pass |
| STC04 | System Test Appointment Management | AppointmentManagement Components::testAppointment Components() | 1, 17, 23, 25.1,  25.2, 27 | Sprint 4, Sprint 5 | 03/10/2012 | System | Pass |
| STC05 | System Test Search Components | ClientSearchComponents:: testClientSearch() | 1, 17, 27 | Sprint 4, Sprint 5 | 03/10/2012 | System | Pass |
| STC06 | System Test Statistical Reporting | StatisticalReporting:: testStatReports | 1, 26, 27 | Sprint 5 | 03/22/2012 | System | Pass |

## ****UI Test Table & Results****

UI testing is twofold, assuring proper navigation and functionality is covered by the Selenium system tests and the following acceptance tests assure that the clients approve of the user interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Title** | **Description** | **Sprint** | **Change Date** | **Type** | **Status** |
| TCUI-01 | UI Test Login/Logout | The login functionality, navigability and appearance. The logout functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 02 | UI Test Client File | The main client file functionality, navigability and appearance. The appointments functionality, navigability and appearance. The dependents functionality, navigability and appearance. The flags functionality, navigability and appearance. The notes functionality, navigability and appearance. The income functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 03 | UI Test Search | The search functionality, navigability and appearance. The result functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 04 | UI Test Create Event | The event creation form functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 05 | UI Test View Events | The events list functionality, navigability and appearance. The check-in list functionality, navigability and appearance. The wait list functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 06 | UI Test Admin Panel | Admin panel categories functionality, navigability and appearance. Admin panel side menu. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 07 | UI Test Help | Appearance of help file and it's coverage. | Sprint 5 | 03/26/2012 | UI Acceptance | TBD |

## Defects

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Title** | **Priority** | **Criticality** | **Platform** | **Browser** | **Sprint** | **Reported By** | **Corrected By** | **Effort (ph)** | **Date** | **Status** |
| D01 Client File Missing Fields | High | Critical | Windows 7 | MS IE 9.0 | Sprint 3 | Client | Mikhail Levkovsky & Patrick Modafferi | 17.25 | 2/20/2012 | Closed |
| D02 Client File Missing Error Msg | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 4 | Katrina Anderson | Katrina Anderson | 0.10 | 2/20/2012 | Closed |
| D03 Flags Need Fixing | High | Significant | Windows 7 | MS IE 9.0 | Sprint 3 | Katrina Anderson | Mikhail Levkovsky | 3.00 | 2/20/2012 | Closed |
| D04 Dependants fix the flow | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 4 | Katrina Anderson | Mikhail Levkovsky & Patrick Modafferi | 2.00 | 2/20/2012 | Closed |
| D05 Dependant Missing Fields | High | Critical | Windows 7 | MS IE 9.0 | Sprint 3 | Client | Patrick Modafferi | 9.00 | 2/22/2012 | Closed |
| D06 Events Mandate how many people per day/per hour | High | Critical | Windows 7 | MS IE 9.0 | Sprint 4 | Client | Cynthia Donato | 10.00 | 2/20/2012 | Closed |
| D07 Client File Management separate citizenship status from work status. | High | Critical | Windows 7 | MS IE 9.0 | Sprint 4 | Client | Mikhail Levkovsky & Patrick Modafferi | 2.00 | 2/29/2012 | Closed |
| D08 Client File Missing Create Date and "Created By" initials should not appear to be modifiable. | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D09 Client File missing "Update Date" and "Updated By" initials should not appear to be modifiable.. | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D10 Household missing "Create Date" and "Created By" initials should not appear to be modifiable. | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D11 Household missing "Update Date" and "Updated By" initials should not appear to be modifiable | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D12 Flags missing "Creation Date" and initials should not appear to be modifiable. | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D13 Income: The additional incomes should be added below current incomes, to be more cohesive with notes and household. | Low | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Josh Hum | 1.00 | 3/15/2012 | Closed |
| D14 Client File: Different postal code field inputs should be modified to fit DB criteria. | High | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Mikhail Levkovsky | Mikhail Levkovsky | 0.1 | 3/15/2012 | Closed |
| D15 Client File: Different telephone field inputs should be modified to fit DB criteria. | High | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Mikhail Levkovsky | Mikhail Levkovsky | 0.1 | 3/15/2012 | Closed |
| D16 Client File: Indicate if client files share the same address. | High | Significant | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Mikhail Levkovsky | 9 | 3/15/2012 | Closed |
| D17 Client File: Indicate if client and dependant Medicare numbers exist elsewhere. | High | Significant | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Mikhail Levkovsky | 3 | 3/15/2012 | Closed |
| D18 Client File: Prevent users from adding an appointment if the file is set to inactive. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Mikhail Levkovsky | 0.1 | 3/15/2012 | Closed |
| D19 Client File: Cascading delete button that appears on a client file up until they have been registered for an appointment. | High | Significant | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Josh Hum | 6 | 3/15/2012 | Closed |
| D20 Event Management: Cascading delete button that appears on an event as long so no one has booked an appointment | High | Significant | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Cynthia Donato | 3.5 | 3/15/2012 | Closed |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D21 Missing error msg for deleting country types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 2 | 3/15/2012 | Closed |
| D22 Missing error msg for deleting citizenship status types | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D23 Missing error msg for deleting flag types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D24 Missing error msg for deleting income types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D25 Missing error msg for deleting income periods. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D26 Missing error msg for deleting language types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D27 Missing error msg for deleting marital status types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D28 Missing error msg for deleting postal code types | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D29 Missing error msg for deleting referral types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D30 Missing error msg for deleting relationship types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D31 Missing error msg for deleting work status types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D32 Missing error msg for deleting event template types. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.1 | 3/15/2012 | Closed |
| D33 Allow Events to be created per person as well as per household | High | Critical | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Cynthia Donato | 7 | 3/15/2012 | Closed |
| D34 Missing active and amount field for event occurrences | High | Critical | Windows 7 | MS IE 9.0 | Sprint 5 | Client | Cynthia Donato | 3 | 3/15/2012 | Open |
| D35 Closing an event does not modify the last visited date on the client file. | High | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Cynthia Donato | 2 | 3/15/2012 | Closed |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D36 Events. Should not be able to book events in the past. | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Cynthia Donato | 3.5 | 3/15/2012 | Open |
| D37 Missing error msg for deleting users (can't delete yourself). | Medium | Minor | Windows 7 | MS IE 9.0 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D38 Client File Income field duration in weeks must be of type double. | High | Minor | Windows 7 | MS IE 9.0 | Sprint 6 | Mikhail Levkovsky | Mikhail Levkovsky | TBD | 3/15/2012 | Closed |

# ****Iteration 6 Test Results Report****

## ****Introduction****

**This test result report contains the testing results for the F.S.T.S. It will be comprised of 8 sections; Acceptance tests, unit tests, integration tests, system tests, UI tests, volume tests, usability tests and Defects, which are outlined in the Test Plan.**

## ****Acceptance Testing****

### ****Iteration 6 Acceptance Test Results****

**Iteration 6 acceptance tests are to be conducted on April 11th 2012, during our class demo. Accompanying acceptance test cases can be found in the Test Cases section of the iteration testing document.**

### ****Acceptance Test Table & Status****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID & Title** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-2 Test Create User | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC02-2 Test Update User | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC03-2 Test Delete User | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC04-2 Test Set User to Administrator | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC05-1 Test View Manage Users | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC06-1 Test User Login | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-2 Test User Login – Omitted Password Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-3 Test User Login – Invalid Password | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-4 Test User Login – Invalid Username | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC06-5 Test User Login – Omitted Username Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC07-1 Test View Check-in For All Clients Coming to an Event - Successful | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC08-1 Test View All clients Coming to an Event - Success Scenario | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC09-1 Test Closing an Event – Success Scenario | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC10-1 Test Appointment Creation – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC11-1 Test Appointment Update – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC12-1 Test Appointment Cancellation – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC13-1 Test Appointment Validation – Capacity Exceeded | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC14-1 Test View Summary of Clients with Appointments to Events | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC16-1 Create a user file with minimal information | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-2 Create complete user file | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-3 Missing Require Fields | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC16-4 Attempt to enter an existing address | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC17-1 Check that edited record is properly updated in the database | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC18-1 Test Client File Set To Active | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC19-1 Test Set Inactive Client Files Script Execution – Success Scenario | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC20-1 Test Set Archive Client Files Script Execution – Success Scenario \*Clients scoped out this functionality | Sprint 4 | 3/12/2012 | Acceptance | FAIL |
| TC21-1: Add New Flag - Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-2: Add New Flag - Failure Scenario 1 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC21-3: Add New Flag – Failure Scenario 2 | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC22-1: Delete Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-1: Update Flag – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-2: Update Flag – Failure Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-3: Update Flag to Inactive – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC23-4: Update Flag to Active – Success Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC24-1 Test Set Flag on New Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC24-2: Test Set Flag on Existing Client File - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC25-1: Test Remove Flag on Existing Client File – Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC26-1 Test Set Note Text on New Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC26-2 Test Set Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC27-1 Test Remove Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC28-1 Test Update Note Text on Existing Client File – Success Scenario | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Association | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - All Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC29-1 Test Client File Dependent Creation - Missing Fields | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC30-1 Test Dependent Deletion | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC31-1 Test Dependent Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-3 Test Postal Code/City Association – with Region Mapped Valid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC32-4 Test Postal Code/City Association – with Invalid Postal Code | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-4 Test case for creating a new country - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC33-5 Test case for creating a new language - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-3 Test case for editing a row that was misspelled - Success Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC34-4 Test case for changing a row - Failure Scenario | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC35-3 Test Deleting Country from drop down lists | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC36-1 Test View Client File Foreign Key Mapping | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TC36-2 Test View Client File User Acceptance of General Layout | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TC37-2 Test Search Client Files by Client ID | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC38-10 Test Search Client Files by Last Name and Medicare Number | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC38-9 Test Search Client Files by Last Name | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC39-1 Test Case to Test Event Template Creation – No Appointment and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-2 Test Case to Test Event Template Creation – Appointment Needed and No Repetition | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-3 Test Case to Test Event Template Creation – Appointment Needed and Event Repeats | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-4 Test Case to Test Event Template Creation – No Appointment Needed and Event | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC39-5 Test Case to Test Event Template Creation – Mandatory Field Omitted | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC40-1 Test Case to Test Event Template Deletion | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-1 Test Case to Test Updating Event Templates – Successful Scenario | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC41-2 Test Case to Test Updating Event Templates – Missing Mandatory Field | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-1 Test Case to Test Display of a List of Event Templates – Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC42-2 Test Case to Test Display of Event Template Details– Successful Creation | Sprint 1 | 1/30/2012 | Acceptance | Pass |
| TC43-5 Test Event Occurrence Creation– No Appointment | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-6 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-7 Test Event Occurrence Creation–Appointment by Hour | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC43-8 Test Event Occurrence Creation–Mandatory Field | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC44-2 Test Event Occurrence Cancellation | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-3 Test Event Occurrence Update | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC45-4 Test Event Occurrence Update – Mandatory Field Omitted | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC46-1 Test Display List of Active Event Occurrences | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC47-1 Test Display of Event Occurrence Details | Sprint 2 | 2/13/2012 | Acceptance | Pass |
| TC48-1 Test Generate Operational Reports – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TC50-1 Test Generate Statistical Reports – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TC51-2 Test Income Source Creation | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC52-2 Test Deleting Income Source | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC53-2 Test Income Source Update | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC54-2 Test Income Source View | Sprint 4 | 3/12/2012 | Acceptance | Pass |
| TC55-1 Test Add Client to Waiting List – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TC55-2 Test Remove Client from Waiting List – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD01-02 Test for creation of correct times for time fields | Sprint 3 | 2/27/2012 | Acceptance | Pass |
| TCD20-1 Test Deleting an Event with No Associated Appointments – Successful | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD33-1 Test Create Appointment per Household Member – Successful | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD34-1 Test Set Event Inactive – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD34-2 Test Set Event Amount – Success Scenario | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD35-1 Test Closing an Event Sets the Last Visited Date for All Attendee | Sprint 5 | 3/26/2012 | Acceptance | Pass |
| TCD56-1 Test Generate Yearly Report - Successful | Sprint 6 | 4/10/2012 | Acceptance | TBD |

## ****Unit Testing****

### ****Iteration 6 Unit Test Results****

**In Iteration 6 we used PHPUnit to create a unit test suite comprised of 45 unit tests. Figure 5.3.2 illustrates the report generated by PHPUnit; As of April 10th, all 45 unit tests passed successfully in 20.185 seconds. To test overall unit test coverage we used XDebug, which measures coverage by counting the statements executed, versus all statements, in a project. Figure 5.3.1 illustrates the report generated by XDebug, which indicates the percent statement coverage of our unit tests to be 64.17%, surpassing our goal of 60%.**

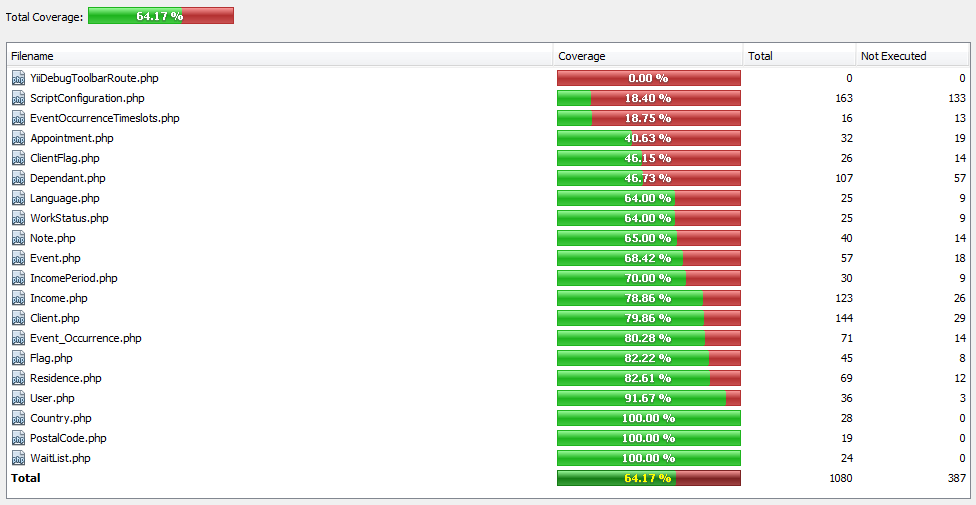
****

Figure 5.3.1 Iteration 5 Unit Test Coverage

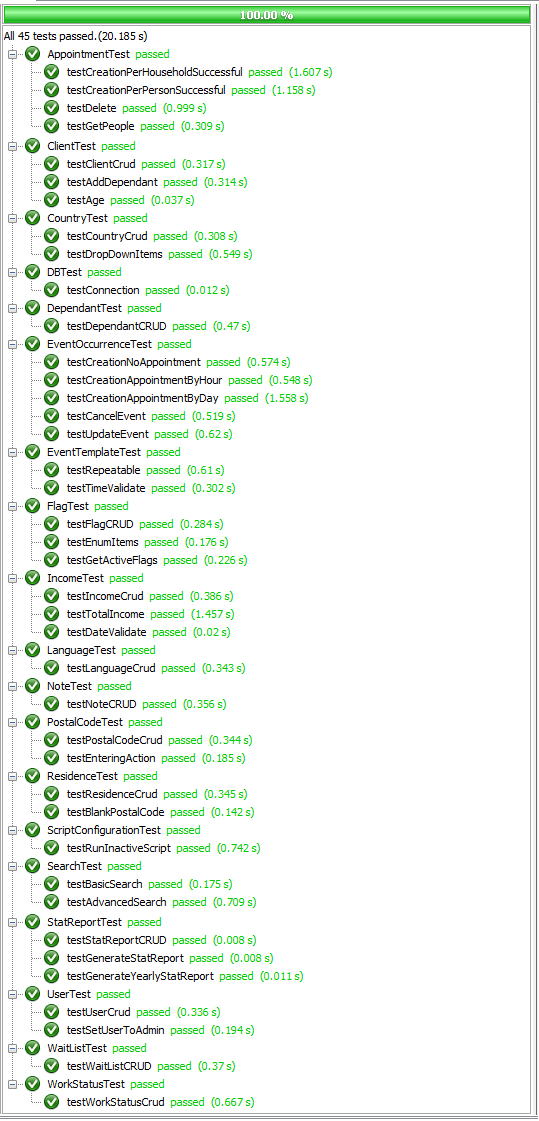


Figure 5.3.2 Iteration 5 Unit Test Results

### ****Unit Test Table & Status****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Title** | **Associated Method** | **Sprint** | **Change Date** | **Type** | **Status** |
| TC01-1 Unit Test Create User | UserTest::testUserCrud() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC02-1 Unit Test Update User | UserTest::testUserCrud() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC03-1 Unit Test Delete User | UserTest::testUserCrud() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC04-1 Unit Test Set User to Administrator | UserTest::testUserToAdmin() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC10-2 Unit Test Appointment Creation - Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC11-2 Unit Test Appointment Update – Success Scenario | AppointmentTest::testCreationSuccessful() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC12-2 Unit Test Appointment Cancellation – Success Scenario | AppointmentTest::testDelete() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC13-2 Unit Test Appointment Validation – Capacity Exceeded | AppointmentTest::testCreationSuccessful() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC18-2 Unit Test Set Client File to Inactive | ClientTest::testActiveSetting() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC18-3 Unit Test Set Client File to Active | ClientTest::testActiveSetting() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC19-2 Unit Test Set Inactive Script Functionality | ScriptConfigurationTest::testRunInactiveScript() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC20-2 Unit Test Archive Client Files Functionality \*Clients scoped out this functionality | ScriptConfigurationTest::testRunArchiveScript() | Sprint 4 | 3/25/2012 | Unit | FAIL |
| TC24-3 Unit Test Client Flag (Part 1/2) | TestFlag::testFlagCRUD | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC25-2 Unit Test Client Flag (Part 2/2) | TestFlag::testFlagCRUD | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC26-3 Unit Test Note - Add Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC27-2 Unit Test Note - Delete Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC28-2 Unit Test Note - Update Note | ClientFlagTest::testFlagCRUDCheck() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TC29-4 Unit Test Create Dependent | DependantTest::testDependantCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC29-5 Unit Test Add Dependent to Client | ClientTest::testAddDependants() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC30-2 Unit Test Delete Dependents | DependantTest::testDependantCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC31-2 Unit Test Update Dependent | DependantTest::testDependantCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC32-1 Unit Test Create-Update-Delete Postal Code | PostalTestCode::testPostalCodeCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC32-2 Unit Test Ajax Request Passes to the Client Controller | PostalTestCode::testEnteringAction() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC33-1 Unit Test for drop down lists | CountryTest::testDropDownItems() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC33-2 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC33-3 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC34-1 Unit Test Update Country | CountryTest::testCountryCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC34-2 Unit Test Update Language | LanguageTest::testLanguageCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC35-1 Unit Test Create Country | CountryTest::testCountryCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC35-2 Unit Test Create Language | LanguageTest::testLanguageCrud() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC37-1 Unit Test Search by Client ID | SearchTest::testBasicSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-1 Unit Test Search by Name | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-2 Unit Test Search by Medicare | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-3 Unit Test Search by Address | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-4 Unit Test Search by Street | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-5 Unit Test Search by Apartment | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-6 Unit Test Search by City | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-7 Unit Test Search by Province | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC38-8 Unit Test Search by Postal Code | SearchTest::testAdvancedSearch() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC43-1 Unit Test Event Occurrence Creation - No Appointment | EventOccurrenceTest::testCreationNoAppointment() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC43-2 Unit Test Event Occurrence Creation - Appointment By Hour | EventOccurrenceTest::testCreationAppointmentByHour() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC43-3 Unit Test Event Occurrence Creation - Appointment By Day | EventOccurrenceTest::testCreationAppointmentByDay() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC44-1 Unit Test Event Occurrence Cancellation | EventOccurrenceTest::testCancelEvent | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC45-1 Unit Test Event Occurrence Update | EventOccurrenceTest::testUpdateEvent() | Sprint 2 | 4/10/2012 | Unit | Pass |
| TC51-1 Unit Test Income Source Creation | IncomeTest::testIncomeCrud() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC52-1 Unit Test Income Source Deletion | IncomeTest::testIncomeCrud() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC53-1 Unit Test Income Source Update | IncomeTest::testIncomeCrud() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TC54-1 Unit Test Income Source View | IncomeTest::testTotalIncome() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TCD01-01 Unit test for Medicare algorithm | ClientTest::testAge() | Sprint 3 | 4/10/2012 | Unit | Pass |
| TCD07-1 Unit Test for Work Status | WorkStatusTest::testWorkStatusCrud() | Sprint 4 | 4/10/2012 | Unit | Pass |
| TCD47-01 Unit Test blank postal codes | ResidenceTest::testBlankPostalCode() | Sprint 6 | 4/10/2012 | Unit | Pass |
| TC48-2 Unit Test Generate Operation Reports | StatReportTest::testStatReportCRUD() | Sprint 5 | 4/10/2012 | Unit | Pass |
| TC50-2 Unit Test Generate Statistical Reports | StatReportTest::testGenerateStatReport() | Sprint 5 | 4/10/2012 | Unit | Pass |
| TC55-2 Unit Test Waiting List | WaitListTest::testWaitListCRUD() | Sprint 5 | 4/10/2012 | Unit | Pass |
| TCD33-2 Unit Test Appointment Creation by Household | Appointment::testCreationPerHouselholdSuccessful() | Sprint 5 | 4/10/2012 | Unit | Pass |
| TCD33-3 Unit Test get people | Appointment::testGetPeople() | Sprint 5 | 4/10/2012 | Unit | Pass |
| TCD70-1 Unit Test Validate Date | IncomeTest::testDateValidate() | Sprint 6 | 4/10/2012 | Unit | Pass |
| TCD71-1 Unit Test Time Validation | EventTemplateTest::testTimeValidate() | Sprint 6 | 4/10/2012 | Unit | Pass |

## ****Integration Testing****

See Master Test Plan 4.2.3 Integration Testing

## ****System Testing****

### ****Iteration 6 System Test Results****

**In Iteration 6 we used Selenium to create system tests that covered 8 out of 8 feature groups for the F.S.T.S, essentially the feature groups that have been executed up until April 10th 2012. Figure 5.5.1. illustrates the Selenium system test.**



Figure 5.5.1 Iteration 6System Test Results

### ****System Test Table & Status****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Title** | **Associated Method** | **Path** | **Sprint** | **Change Date** | **Type** | **Status** |
| STC01 | System Test Administrative Components | AdministrationComponents:: testAdministrationComponents() | 1, 2, 3, 2, 4, 2, 5, 2, 6, 2, 7, 2, 8, 2, 9,  2, 10, 2, 11, 2, 12 ,2, 13,  2, 14, 2, 16, 27 | Sprint 4 | 3/22/2012 | System | Pass |
| STC02 | System Test Client File Management | ClientManagementComponents ::testClientManagement Components() | 1, 18.1, 18.3,  19.1, 19.3, 19.2,  20.1, 20.3, 20.2,  21.1, 21.2, 22.1,  22.2, 18.2, 27 | Sprint 4, Sprint 5 | 3/22/2012 | System | Pass |
| STC03 | System Test Event Management | EventManagementComponents ::testEventManagement() | 1, 2, 15, 24.1,  24.3, 24.2, 27 | Sprint 4, Sprint 5 | 3/22/2012 | System | Pass |
| STC04 | System Test Appointment Management | AppointmentManagement Components::testAppointment Components() | 1, 17, 23, 25.1,  25.2, 27 | Sprint 4, Sprint 5 | 3/22/2012 | System | Pass |
| STC05 | System Test Search Components | ClientSearchComponents:: testClientSearch() | 1, 17, 27 | Sprint 4, Sprint 5 | 3/22/2012 | System | Pass |
| STC06 | System Test Statistical Reporting | StatisticalReporting:: testStatReports | 1, 26, 27 | Sprint 5, Sprint 6 | 4/10/2012 | System | Pass |

## ****UI Test Table & Results****

UI testing is twofold, assuring proper navigation and functionality is covered by the Selenium system tests and the following acceptance tests assure that the clients approve of the user interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **System Test ID** | **Title** | **Description** | **Sprint** | **Change Date** | **Type** | **Status** |
| TCUI01 | UI Test Login/Logout | The login functionality, navigability and appearance. The logout functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | Pass |
| TCUI 02 | UI Test Client File | The main client file functionality, navigability and appearance. The appointments functionality, navigability and appearance. The dependents functionality, navigability and appearance. The flags functionality, navigability and appearance. The notes functionality, navigability and appearance. The income functionality, navigability and appearance. Error and success messages. | Sprint 6 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 03 | UI Test Search | The search functionality, navigability and appearance. The result functionality, navigability and appearance. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | Pass |
| TCUI 04 | UI Test Create Event | The event creation form functionality, navigability and appearance. Error and success messages. | Sprint 6 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 05 | UI Test View Events | The events list functionality, navigability and appearance. The check-in list functionality, navigability and appearance. The wait list functionality, navigability and appearance. Error and success messages. | Sprint 6 | 03/26/2012 | UI Acceptance | TBD |
| TCUI 06 | UI Test Admin Panel | Admin panel categories functionality, navigability and appearance. Admin panel side menu. Error and success messages. | Sprint 5 | 03/26/2012 | UI Acceptance | Pass |
| TCUI 07 | UI Test Help | Appearance of help file and it's coverage. | Sprint 5 | 03/26/2012 | UI Acceptance | Pass |

## Volume Test

**The following testing results have been acquired through the technique of volume testing discussed in the test plan document. It was conducted to provide an idea of how the system responds to a given amount of volume (data) in the system.**

### ****Initial results and Solutions to Perceived Problems****

**Initially, the system did not have a limit on how many results were returned. On initial testing through 20000 records and searching for ‘0’, which is contained in nearly every client’s id number, the program failed as shown in the screenshot below.(The same results were obtained:**

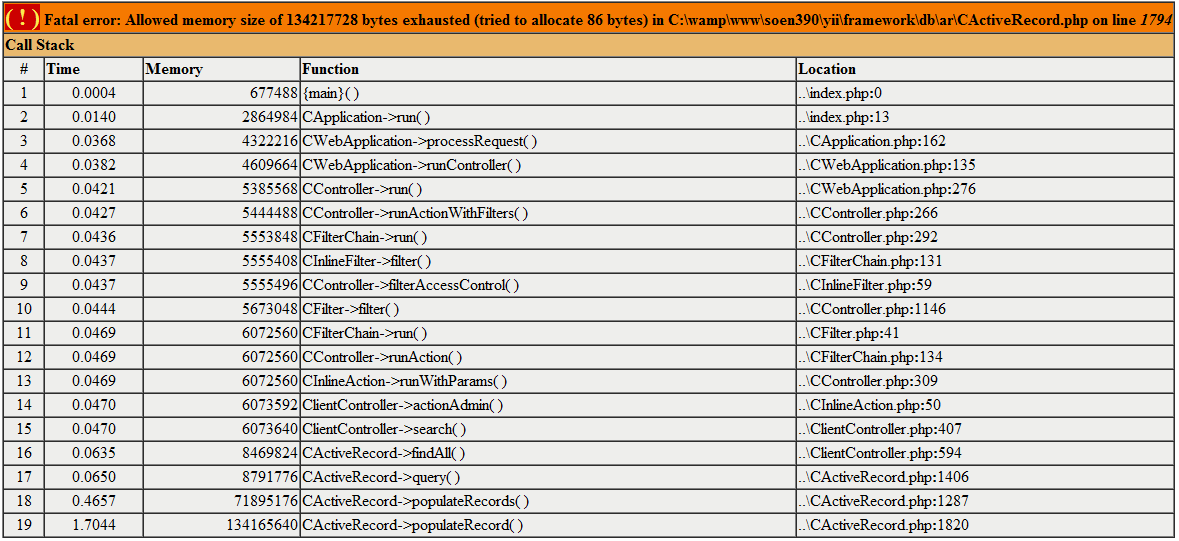


Figure 5.7.1 Memory exhausted message

We concluded that the default 128MB is a reasonable limit to set for a basic web application, however, the FSTS is business grade software running solely on internal servers. This means that much more RAM will be available and it is fair to test the system under different settings. We used the following php functions to change the settings for our environment to better simulate the context at Welcome Hall Mission:

set\_time\_limit ( 250 ); //in seconds

ini\_set('memory\_limit', '1024'); //in MB

### ****Client File Search****

This section represents volume tests performed on the client search functionality of the system. In this test, the testing simply loaded the database to the desired volume, then modified a piece of code which determined the amount of results the system should return to all desired values before starting over with a new database volume and repeating the exercise.

#### Volume Test Table & Status 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Searching Speeds** | | | | | | | |
| **ID** | **Client Files** | **Returned Results** | **Time (s)** | **Iteration** | **Change Date** | **Type** | **Status** |
| TCV-03 | 10000 | 100 | 0.077 | Sprint 6 | 04/05/2012 | Volume | Acceptable |
| 1000 | 0.202 |
| 10000 | 1.2939 |
| 15000 | 100 | 0.0498 | Sprint 6 | 04/07/2012 | Volume | Acceptable |
| 1000 | 0.186 |
| 10000 | 2.157 |
| 20000 | 100 | 0.053 | Sprint 6 | 04/08/2012 | Volume | Acceptable |
| 1000 | 0.2785 |
| 10000 | 3.449 |
| 25000 | 100 | 0.0377 | Sprint 6 | 04/08/2012 | Volume | Acceptable |
| 1000 | 0.208 |
| 10000 | 4.021 |
| 30000 | 100 | 0.0377 | Sprint 6 | 04/10/2012 | Volume | Acceptable |
| 1000 | 0.2635 |
| 10000 | 3.21 |

Here is a sample output of the test run. The result is show on the top left corner:

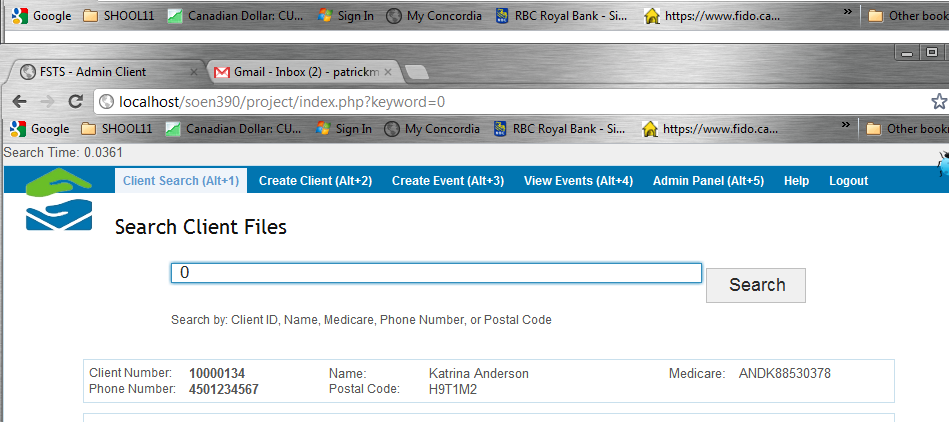


Figure 5.7.2 Memory exhausted message

#### Conclusion

In this test we are not concerned with the strings input into the search since upon thorough attempts with different queries, we noted that times do not vary significantly with multiple strings or longer strings. We have deemed these results acceptable based on our SRS document’s specification. Furthermore, we have decided to limit the results to 100. Based on experience with search engines, a user does not wish to browse more than 100 returned results and would prefer to simply narrow the search and not due to performance issues.

Refer to SRS section 5.1

### Statistical Report Generation

The following are the results for the testing on the statistical report component. After doing the search test, rather than reloading the database with more entries and completing all of the search tests, the tester immediately moves on and tests this component under the given setup. In these series of tests, both PDF and Excel formats were tested. The difference here is that the PDF format is a single report for a given time interval and the Excel one must generate 13 reports in one file; one for each month plus one for total. For this reason, the test results tracks its progress throughout the generation of each of these nested reports and gathers the results for each month and then the total.

#### Initial Results

Refer to Search Client Files – Initial Results

#### Volume Test Table & Status 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stress Test ID** | **Setup** | | **Results** | | | **Iteration** | **Change Date** | **Type** | **Status** |
| **Clients** | **Appt.** | **Date Interval** | **Time(s)** | **Mem Usage (MB)** |
| TCV-01 (Excel) | **10000** | **10000** | 2012-01-02-2013-01-02 | 38.7098 | 197.5 | Sprint 6 | 04/06/2012 | Stress | Pass |
| 2012-01-02-2012-02-02 | 6.1587 | 199.5 |
| 2012-02-02-2012-03-02 | 8.3052 | 199.5 |
| 2012-03-02-2012-04-02 | 13.709 | 199.5 |
| 2012-04-02-2012-05-02 | 6.7062 | 199.5 |
| 2012-05-02-2012-06-02 | 6.7646 | 199.5 |
| 2012-06-02-2012-07-02 | 14.9907 | 225 |
| 2012-07-02-2012-08-02 | 7.2647 | 232.5 |
| 2012-08-02-2012-09-02 | 6.7973 | 244.5 |
| 2012-09-02-2012-10-02 | 14.1824 | 275.25 |
| 2012-10-02-2012-11-02 | 7.4828 | 282.5 |
| 2012-11-02-2012-12-02 | 7.438 | 295.25 |
| 2012-12-02-2013-01-02 | 9.2357 | 320.5 |
| Total | 148.3424 | 317.5 |
| TCV-02 (PDF) | **10000** | **10000** | 2011-04-01-2013-04-25 | 37.5828 | 194.25 | Sprint 6 | 04/06/2012 | Stress | Pass |
|  |  |  | 2012-03-01-2012-04-03 | 12.6336 | 49.5 |  |  |  |  |

#### Volume Test Table & Status 2 (Continued)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stress Test ID** | **Setup** | | **Results** | | | **Iteration** | **Change Date** | **Type** | **Status** |
| **Clients** | **Appt.** | **Date Interval** | **Time(s)** | **Mem Usage (MB)** |
| TCV-01 (Excel) | **15000** | **15000** | 2012-01-02-2013-01-02 | 49.5725 | 291 | Sprint 6 | 04/08/2012 | Stress | Pass |
| 2012-01-02-2012-02-02 | 7.5894 | 292 |
| 2012-02-02-2012-03-02 | 11.3995 | 292 |
| 2012-03-02-2012-04-02 | 20.4402 | 293 |
| 2012-04-02-2012-05-02 | 11.0977 | 293 |
| 2012-05-02-2012-06-02 | 11.2176 | 293 |
| 2012-06-02-2012-07-02 | 20.8283 | 297.5 |
| 2012-07-02-2012-08-02 | 10.5873 | 302.25 |
| 2012-08-02-2012-09-02 | 12.1811 | 323.75 |
| 2012-09-02-2012-10-02 | 22.0783 | 372 |
| 2012-10-02-2012-11-02 | 12.9909 | 390 |
| 2012-11-02-2012-12-02 | 10.7051 | 399.25 |
| 2012-12-02-2013-01-02 | 13.62 | 424.25 |
| Total | 214.8072 | 422.5 |
| TCV-02 (PDF) | **15000** | **15000** | 2011-04-01-2013-04-25 | 49.0574 | 285.75 | Sprint 6 | 04/08/2012 | Stress | Pass |
|  |  |  | 2012-03-01-2012-04-03 | 16.3207 | 62.5 |  |  |  |  |
| TCV-01 (Excel) | **\*20000** | **20000** | 2012-01-02-2013-01-02 | 64.0881 | 219.5 | Sprint 6 | 04/08/2012 | Stress | Pass |
| 2012-01-02-2012-02-02 | 10.1961 | 221 |
| 2012-02-02-2012-03-02 | 14.347 | 221 |
| 2012-03-02-2012-04-02 | 28.0163 | 241.5 |
| 2012-04-02-2012-05-02 | 12.097 | 246 |
| 2012-05-02-2012-06-02 | 13.3465 | 270.25 |
| 2012-06-02-2012-07-02 | 27.4668 | 323.25 |
| 2012-07-02-2012-08-02 | 14.8799 | 341.5 |
| 2012-08-02-2012-09-02 | 14.2876 | 366.25 |
| 2012-09-02-2012-10-02 | 27.3387 | 405.75 |
| 2012-10-02-2012-11-02 | 15.4738 | 428.5 |
| 2012-11-02-2012-12-02 | 15.3143 | 452.5 |
| 2012-12-02-2013-01-02 | 15.8955 | 476.5 |
| Total | 273.1525 | 476.5 |
| TCV-02 (PDF) | **\*20000** | **20000** | 2011-04-01-2013-04-25 | 77.3335 | 378 | Sprint 6 | 04/08/2012 | Stress | Pass |
|  |  |  | 2012-03-01-2012-04-03 | 22.3946 | 80.5 |  |  |  |  |

\*After 20000 records we can observe that the PDF format is performing well relatively to the Excel one. Therefore, we will continue to push the Excel format and can stop testing the PDF version.

#### Volume Test Table & Status 2 (Continued)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stress Test ID** | **Setup** | | **Results** | | | **Iteration** | **Change Date** | **Type** | **Status** |
| **Clients** | **Appt.** | **Date Interval** | **Time(s)** | **Mem Usage (MB)** |
| TCV-01 (Excel) | **25000** | **25000** | 2012-01-02-2013-01-02 | 96.5414 | 270 | Sprint 6 | 04/09/2012 | Stress | Pass |
| 2012-01-02-2012-02-02 | 13.1641 | 272.5 |
| 2012-02-02-2012-03-02 | 20.1912 | 272.5 |
| 2012-03-02-2012-04-02 | 39.3316 | 306 |
| 2012-04-02-2012-05-02 | 17.3091 | 319.5 |
| 2012-05-02-2012-06-02 | 19.4459 | 359 |
| 2012-06-02-2012-07-02 | 36.7832 | 411 |
| 2012-07-02-2012-08-02 | 21.2118 | 443.75 |
| 2012-08-02-2012-09-02 | 18.3759 | 466 |
| 2012-09-02-2012-10-02 | 38.8749 | 516 |
| 2012-10-02-2012-11-02 | 22.8566 | 549.25 |
| 2012-11-02-2012-12-02 | 20.1934 | 573.75 |
| 2012-12-02-2013-01-02 | 23.5189 | 610.25 |
| Total | 388.266 | 610.25 |
| TCV-01 (Excel) | **30000** | **30000** | 2012-01-02-2013-01-02 | 140.1851 | 327.75 | Sprint 6 | 04/10/2012 | Stress | Pass |
| 2012-01-02-2012-02-02 | 17.9279 | 329.25 |
| 2012-02-02-2012-03-02 | 24.9415 | 329.25 |
| 2012-03-02-2012-04-02 | 52.9623 | 374.25 |
| 2012-04-02-2012-05-02 | 23.4958 | 402 |
| 2012-05-02-2012-06-02 | 24.2133 | 438.25 |
| 2012-06-02-2012-07-02 | 48.4591 | 503.5 |
| 2012-07-02-2012-08-02 | 25.873 | 532.25 |
| 2012-08-02-2012-09-02 | 25.9795 | 568.5 |
| 2012-09-02-2012-10-02 | 55.9704 | 652.25 |
| 2012-10-02-2012-11-02 | 28.6159 | 683.25 |
| 2012-11-02-2012-12-02 | 29.1781 | 719 |
| 2012-12-02-2013-01-02 | 29.5225 | 755 |
| Total | 527.8666 | 751.25 |

Here is a screen shot of how the results are gathered and output during a volume test:

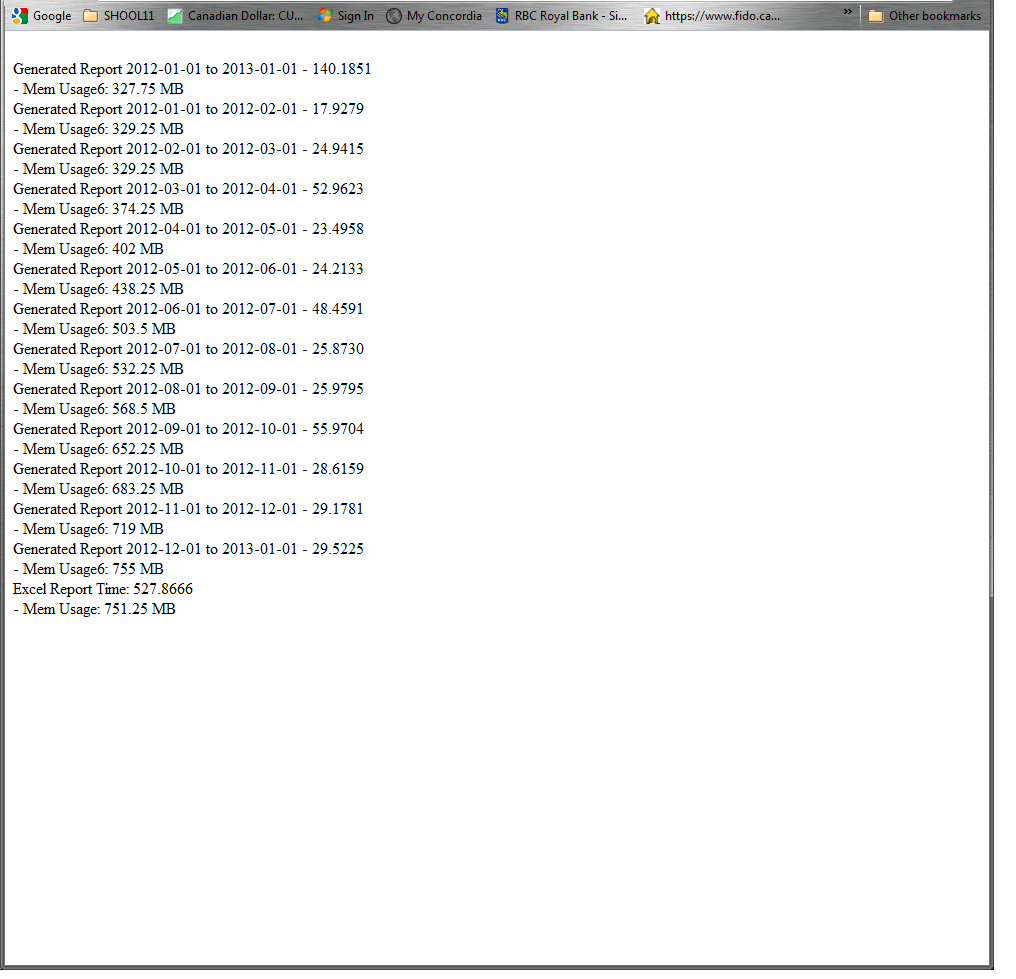


Figure 5.7.3 Memory exhausted message

#### Conclusion

These results have been deemed acceptable for results discussed earlier; the environment at Welcome Hall Mission should allow for this size of memory usage. As for time constraints, the later tests have shown to exceed the previously set limit of 5 minutes set by section 5.1 in the SRS document. However, the input volume for the later tests represents a value that far exceeds the regular environment in which the software is currently working.

Refer to SRS Section 5.1

### Database Size Usage

The database was loaded with over 100,000 client files, 100,000 appointments, over 200 countries and over 200 languages. The result of this test was that the database under an extreme volume took up less than 20MB of disk space which is not even near the amount available on the server of 100GB.

Refer to SRS Section 5.4

## ****Usability Test Results.****

### Results for Naomi Mpinga

#### User Profile

Name: Naomi Mpinga

Age: 24

Gender: Female

Occupation: Nursing Student

Computer Literacy: Average

#### Test Results

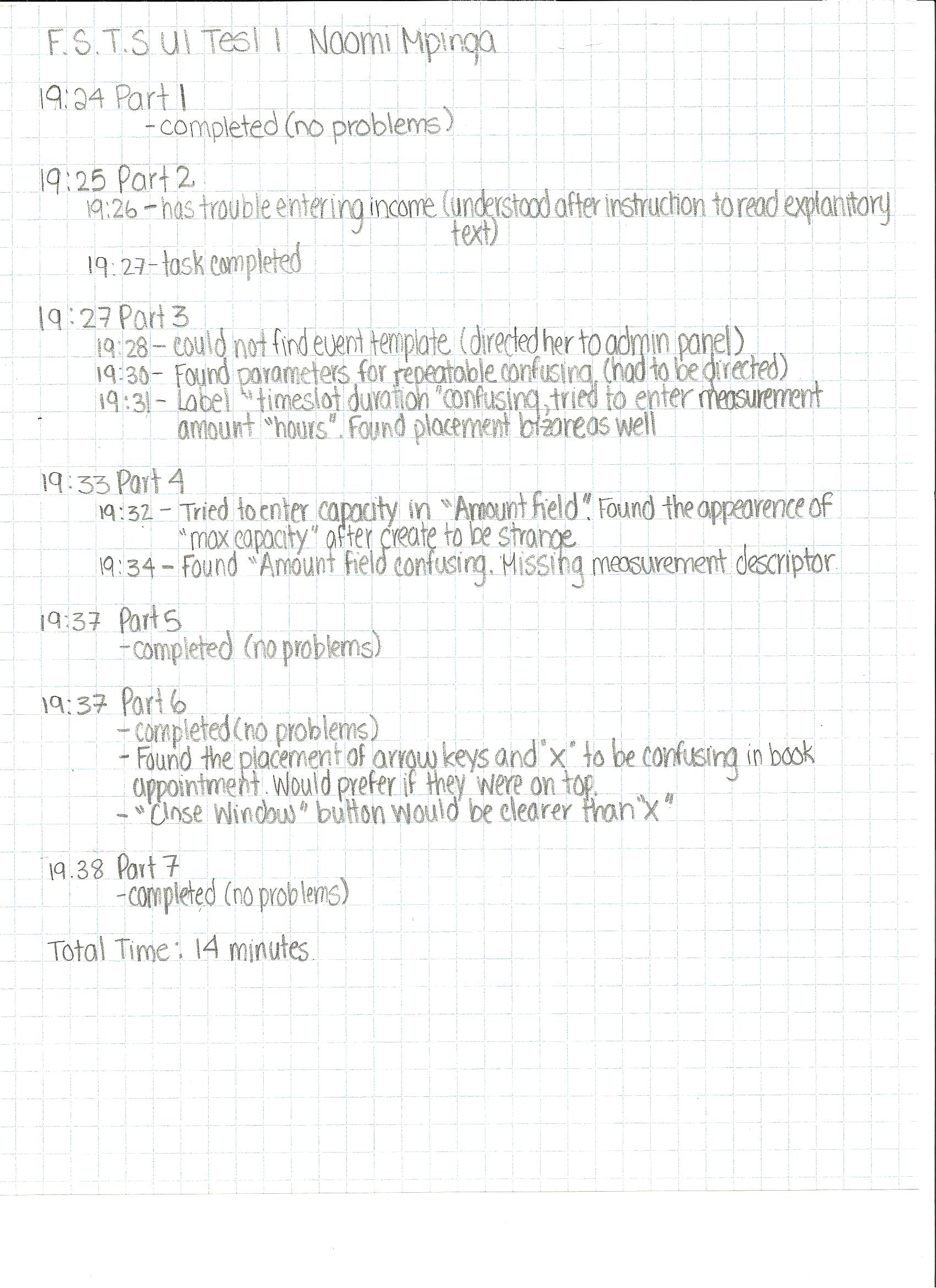


Figure 5.8.1 Memory exhausted message

### Results for Roxanne Labelle

#### User Profile

Name: Roxanne Labelle

Age: 50

Gender: Female

Occupation: Home Daycare Provider

Computer Literacy: Low

#### Test Results

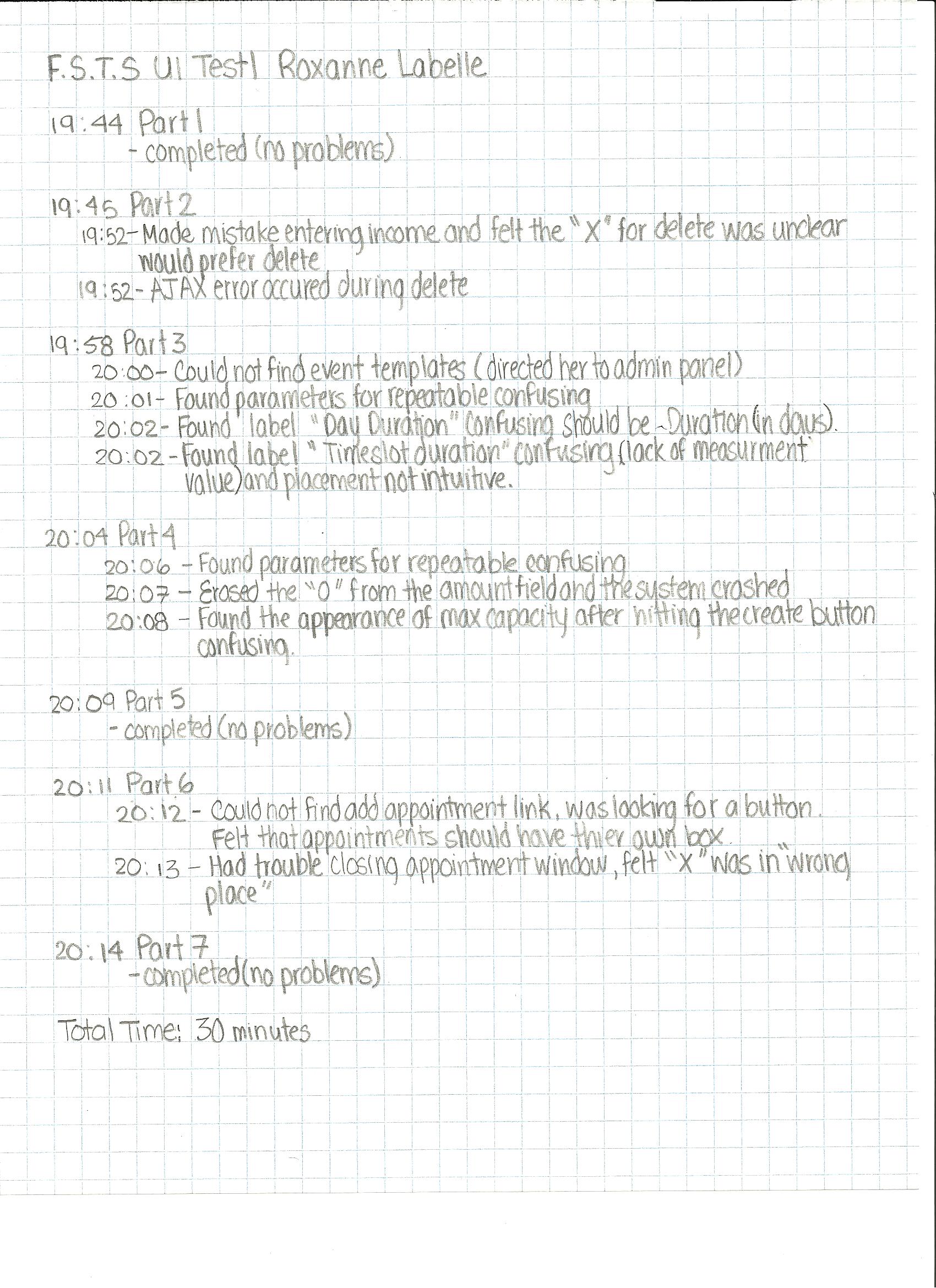


Figure 5.8.2 Memory exhausted message

#### User Profile

Name: Tania Togian & Gordon McPhee

Gender: Female

Occupation: Employee Welcome Hall Mission

Computer Literacy:Medium

#### Test Results

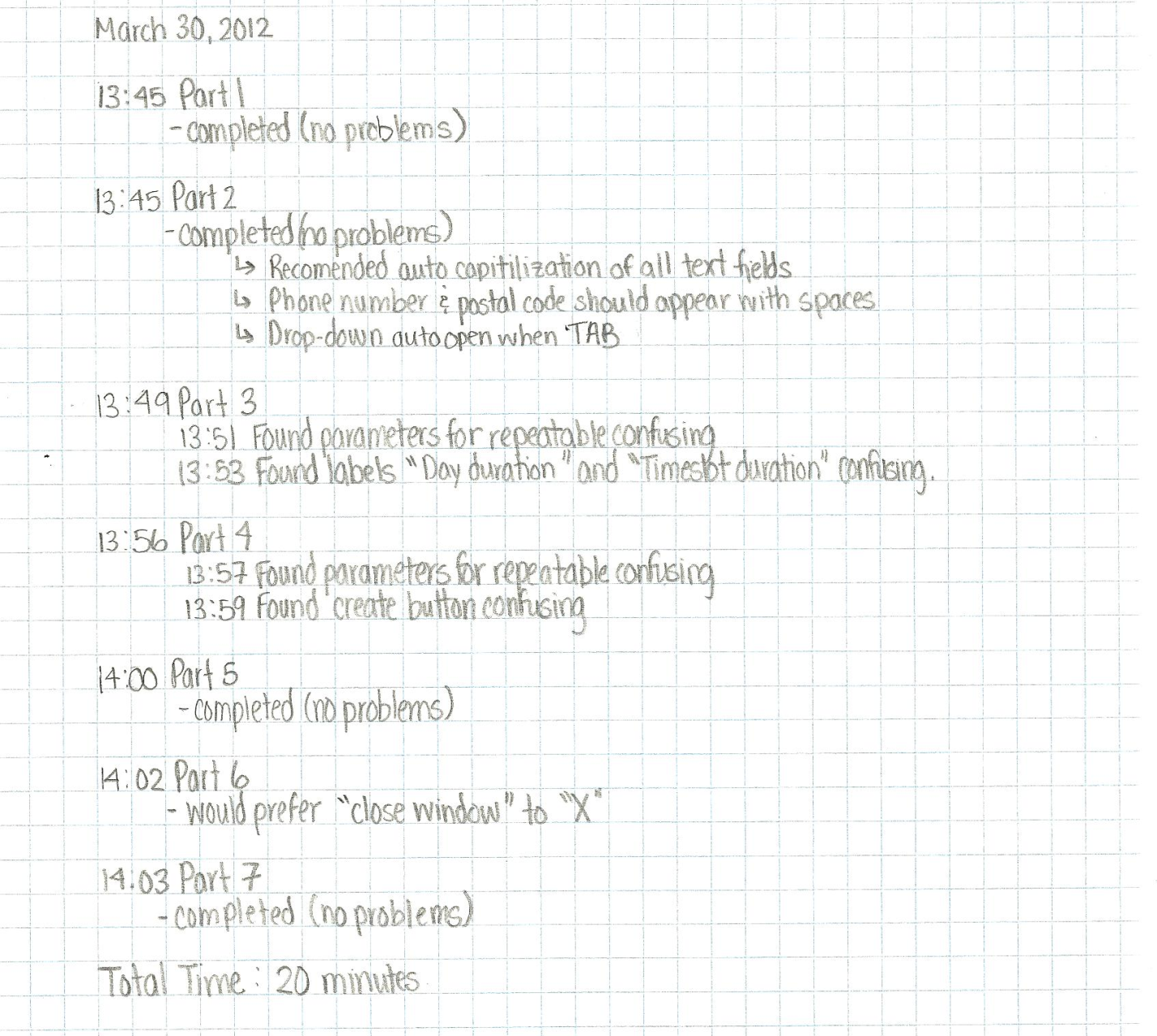


Figure 5.8.3 Memory exhausted message

### Analysis & Solutions to Perceived Problems

Overall the user feedback suggests that the F.S.T.S. was easy to learn and intuitive. In terms of efficiency, filling in fields went fairly quickly, due to the many auto complete features of the system. The following describes areas where the testers expressed confusion or in extreme cases got stuck, along with possible solutions.

#### Client File

Users could not find the add appointment link because they were expecting a button. Proposed solution is to change the link “add appointment” to a button.



Figure 5.8.4 Add Appointment Link

Users Found the placement of arrow keys and close button on the “Appointment Creation” window to be confusing and poorly placed (on bottom of window).Users would have liked the buttons to be on top of window, with descriptions of where the arrow keys lead (Users thought the navigation keys were undo and redo buttons). Also instead of an “X” button, users would prefer a descriptive “close window.”

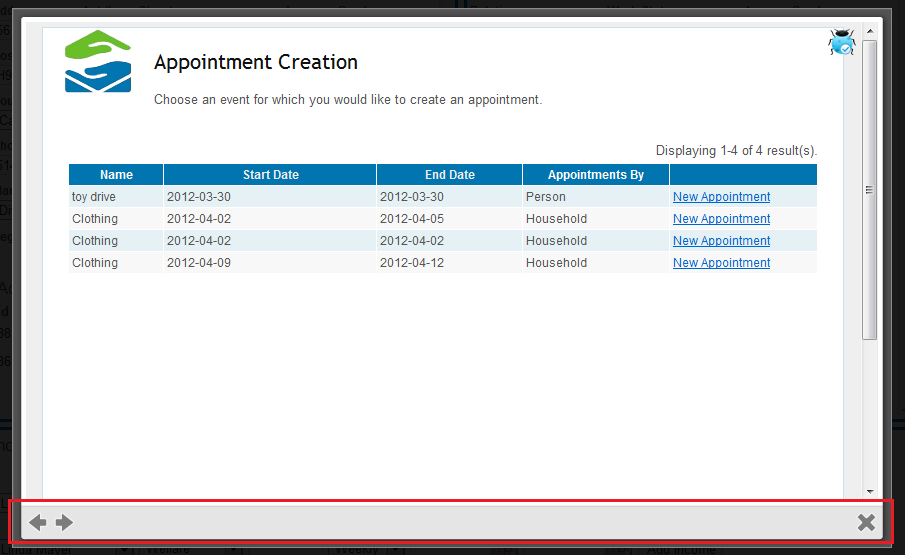


Figure 5.8.5 Appointment Creation Navigation Buttons

#### Event Templates

Users found the parameters for the repeatable events to be confusing. Meaning they felt it was unclear that the parameters belonged to the repeatable section. A possible solution would be for the parameters to only appear if repeatable is set to “yes.”

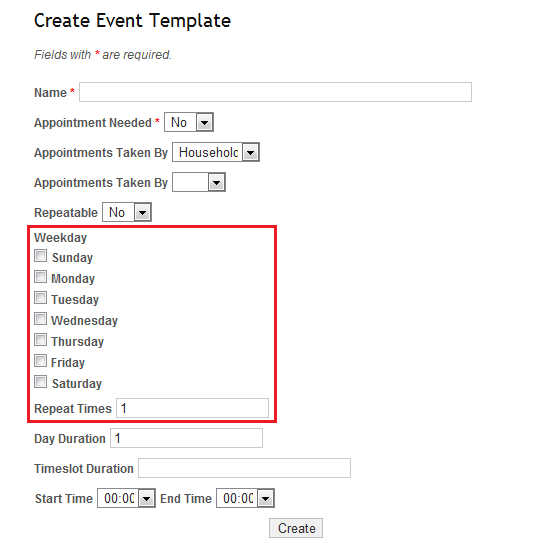


Figure 5.8.6 Event Template Repeatable

Users found label “Day Duration” to be confusing. A solution would be to change the label to “Duration (in days).”



Figure 5.8.7 Event Template Day Duration

Users found label “Timeslot Duration” to be confusing, due to its lack of measurement value and placement. “Timeslot Duration” field should be moved closer to “Appointments Taken By” field to add validity in terms of context.

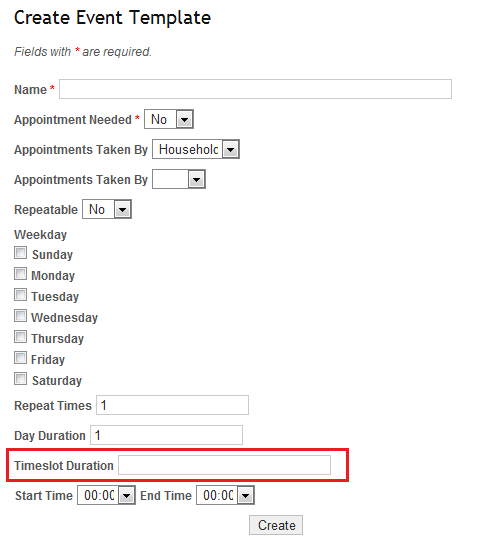


Figure 5.8.8 Event Template Timeslot Duration

#### Event Occurrence

Users found the parameters for the repeatable events to be confusing. Meaning they felt it was unclear that the parameters belonged to the repeatable section. A possible solution would be for the parameters to only appear if repeatable is set to “yes”.

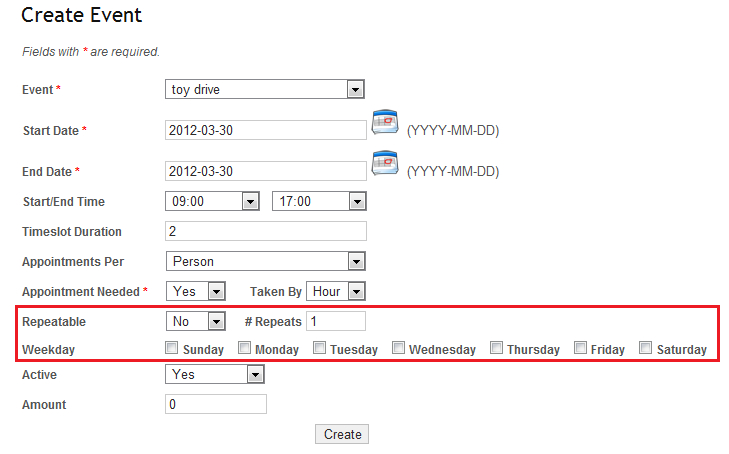


Figure 5.8.9 Create Event Repeatable

Users found it confusing that capacity of timeslots is only defined after the event is created. A possible solution would be to change the label “Create” on the button to say “Next > Define Event Capacity”

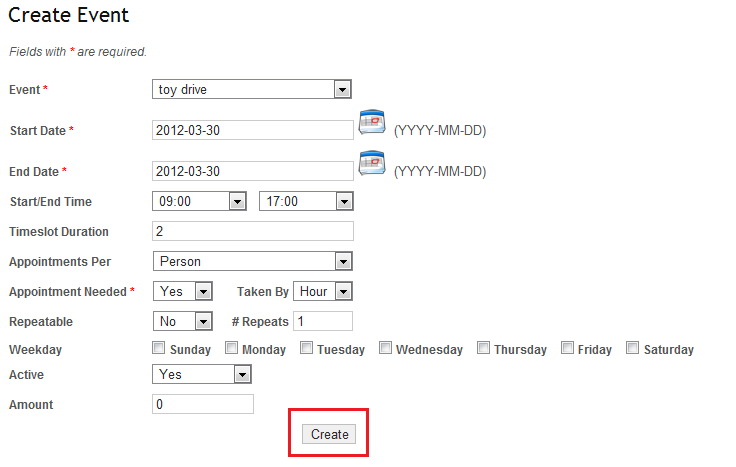


Figure 5.8.10 Create Event Create Button

### Conclusion

Overall, the goals of the usability testing were to evaluate the level of ease towards learning and using the F.S.T.S’s interface, as well as the efficiency of the interface. The results of the usability tests, as well as tester feedback suggest that learning to use the F.S.T.S is relatively easy, the only challenging parts being in the Admin Panel. On average, trained volunteers at the Welcome Hall Mission take 20 minutes to register a client. A user with a low level understanding of computers, who has never seen the F.S.T.S. previously and is unaware of the domain, is able to create an event template, an event occurrence, a client file, an appointment and execute a search in 30 minutes. In other words, it appears that the new F.S.T.S. is far superior in terms of efficiency.

## Defects

**Platform: Windows 7**

**Browser: MS IE 9.0**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Title** | **Priority** | **Criticality** | **Related Test Case** | **Sprint** | **Reported By** | **Corrected By** | **Effort (ph)** | **Date** | **Status** |
| D01 Client File Missing Fields | High | Critical | TC16.1 | Sprint 3 | Client | Mikhail Levkovsky & Patrick Modafferi | 17.25 | 2/20/2012 | Closed |
| D02 Client File Missing Error Msg | Medium | Minor | TC16.1 | Sprint 4 | Katrina Anderson | Katrina Anderson | 0.10 | 2/20/2012 | Closed |
| D03 Flags Need Fixing | High | Significant | TC22.1 | Sprint 3 | Katrina Anderson | Mikhail Levkovsky | 3.00 | 2/20/2012 | Closed |
| D04 Dependents fix the flow | Low | Minor | TC29.2 | Sprint 4 | Katrina Anderson | Mikhail Levkovsky & Patrick Modafferi | 2.00 | 2/20/2012 | Closed |
| D05 Dependent Missing Fields | High | Critical | TC29.2 | Sprint 3 | Client | Patrick Modafferi | 9.00 | 2/22/2012 | Closed |
| D06 Events Mandate how many people per day/per hour | High | Critical | TC43.6 & TC43.7 | Sprint 4 | Client | Cynthia Donato | 10.00 | 2/20/2012 | Closed |
| D07 Client File Management separate citizenship status from work status. | High | Critical | TC16.1 | Sprint 4 | Client | Mikhail Levkovsky & Patrick Modafferi | 2.00 | 2/29/2012 | Closed |
| D08 Client File Missing Create Date and "Created By" initials should not appear to be modifiable. | Low | Minor | TC16.1 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D09 Client File missing "Update Date" and "Updated By" initials should not appear to be modifiable.. | Low | Minor | TC16.1 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D10 Household missing "Create Date" and "Created By" initials should not appear to be modifiable. | Low | Minor | TC31.1 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D11 Household missing "Update Date" and "Updated By" initials should not appear to be modifiable | Low | Minor | TC31.1 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D12 Flags missing "Creation Date" and initials should not appear to be modifiable. | Low | Minor | TC23.1 | Sprint 5 | Client | Adrian Lloyd | 0.10 | 3/15/2012 | Closed |
| D13 Income: The additional incomes should be added below current incomes, to be more cohesive with notes and household. | Low | Minor | TC51.2 | Sprint 5 | Katrina Anderson | Josh Hum | 1.00 | 3/15/2012 | Closed |
| D14 Client File: Different postal code field inputs should be modified to fit DB criteria. | High | Minor | TC16.1 | Sprint 5 | Mikhail Levkovsky | Mikhail Levkovsky | 0.10 | 3/15/2012 | Closed |
| D15 Client File: Different telephone field inputs should be modified to fit DB criteria. | High | Minor | TC16.1 | Sprint 5 | Mikhail Levkovsky | Mikhail Levkovsky | 0.10 | 3/15/2012 | Closed |
| D16 Client File: Indicate if client files share the same address. | High | Significant | TC16.1 | Sprint 5 | Client | Mikhail Levkovsky | 9.00 | 3/15/2012 | Closed |
| D17 Client File: Indicate if client and dependent Medicare numbers exist elsewhere. | High | Significant | TC16.1 | Sprint 5 | Katrina Anderson | Mikhail Levkovsky | 3.00 | 3/15/2012 | Closed |
| D18 Client File: Prevent users from adding an appointment if the file is set to inactive. | Medium | Minor | TC16.1 | Sprint 5 | Client | Mikhail Levkovsky | 0.10 | 3/15/2012 | Closed |
| D19 Client File: Cascading delete button that appears on a client file up until they have been registered for an appointment. | High | Significant | TC12.1 | Sprint 5 | Katrina Anderson | Josh Hum | 6.00 | 3/15/2012 | Closed |
| D20 Event Management: Cascading delete button that appears on an event as long so no one has booked an appointment | High | Significant | TC12.1 | Sprint 5 | Katrina Anderson | Cynthia Donato | 3.50 | 3/15/2012 | Closed |
| D21 Missing error msg for deleting country types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 2.00 | 3/15/2012 | Closed |
| D22 Missing error msg for deleting citizenship status types | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D23 Missing error msg for deleting flag types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D24 Missing error msg for deleting income types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D25 Missing error msg for deleting income periods. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D26 Missing error msg for deleting language types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D27 Missing error msg for deleting marital status types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D28 Missing error msg for deleting postal code types | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D29 Missing error msg for deleting referral types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D30 Missing error msg for deleting relationship types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D31 Missing error msg for deleting work status types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D32 Missing error msg for deleting event template types. | Medium | Minor | STC01 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D33 Allow Events to be created per person as well as per household | High | Critical | TCD33.1 | Sprint 5 | Client | Cynthia Donato | 7.00 | 3/15/2012 | Closed |
| D34 Missing active and amount field for event occurrences | High | Critical | TCD34.1 & TCD34.2 | Sprint 5 | Client | Cynthia Donato | 3.00 | 3/15/2012 | Closed |
| D35 Closing an event does not modify the last visited date on the client file. | High | Minor | TCD35.1 | Sprint 5 | Katrina Anderson | Cynthia Donato | 2.00 | 3/15/2012 | Closed |
| D36 Events. Should not be able to book events in the past. | Medium | Minor | STC03 | Sprint 5 | Katrina Anderson | Cynthia Donato | 3.50 | 3/15/2012 | Closed |
| D37 Missing error msg for deleting users (can't delete yourself). | Medium | Minor | STC03 | Sprint 5 | Katrina Anderson | Patrick Modafferi | 0.10 | 3/15/2012 | Closed |
| D38 Client File Income field duration in weeks must be of type double. | High | Minor | STC02 | Sprint 6 | Mikhail Levkovsky | Mikhail Levkovsky | 0.25 | 3/15/2012 | Closed |
| D39 Fix the built in search functionality in IE. | Medium | Minor | TCU-01 | Sprint 6 | Josh Hum | Mikhail Levkovsky | 0.25 | 4/7/2012 | Closed |
| D40 Filter events by event type | Medium | Minor | TCU-01 | Sprint 6 | Katrina Anderson | Cynthia Donato | 2.00 | 4/8/2012 | Closed |
| D41 Automatically sort event list by date | Medium | Minor | STC03 | Sprint 6 | Katrina Anderson | Cynthia Donato | 1.00 | 4/7/2012 | Closed |
| D42 Error Msg deleting events from management | Medium | Significant | STC03 | Sprint 6 | Katrina Anderson | Cynthia Donato,Patrick Modafferi | 0.25 | 4/6/2012 | Closed |
| D43 Check-In must be sorted in alphabetical order | Medium | Minor | STC04 | Sprint 6 | Katrina Anderson | Cynthia Donato | 0.50 | 4/7/2012 | Closed |
| D44 Fit Event information on check in list into 2 lines | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 1.50 | 4/10/2012 | Closed |
| D45 Search should have paging system | Medium | Minor | TCU-01 | Sprint 6 | Patrick Modafferi | Josh Hum,Cynthia Donato | 2.50 | 4/8/2012 | Closed |
| D46 Search must have a limit on search results (stack-overflow) | High | Significant | TC37.2 | Sprint 6 | Patrick Modafferi | Josh Hum, Patrick Modafferi | 1.00 | 4/7/2012 | Closed |
| D47 Client File conflict on postal codes that are left blank/ blank postal codes should be ignored | High | Minor | STC02 | Sprint 6 | Katrina Anderson | Mikhail Levkovsky | 0.50 | 4/7/2012 | Closed |
| D48 Client File conflict on Medicare card numbers that are left blank/ blank Medicare cards should be ignored. | High | Minor | STC02 | Sprint 6 | Katrina Anderson | Mikhail Levkovsky | 0.50 | 4/6/2012 | Closed |
| D49 Client Name should appear in appointment window | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 0.50 | 4/5/2012 | Closed |
| D50 Auto capitalize first letter in all fields | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 3.50 | 4/7/2012 | Closed |
| D51 Display postal code and phone number with spaces. | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 1.00 | 4/7/2012 | Closed |
| D52 Modify event template view | Medium | Minor | TCU-01 | Sprint 6 | Client | Josh Hum, Katrina Anderson | 2.00 | 4/8/2012 | Closed |
| D53 Modify Event view | Medium | Minor | TCU-01 | Sprint 6 | Client | Josh Hum, Katrina Anderson | 2.00 | 4/8/2012 | Closed |
| D54 Change "X" to close window button in make appointment window. | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 0.25 | 4/5/2012 | Closed |
| D55 When you tab to a dropdown field, the dropdown must open. | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 10.00 | 4/10/2012 | NO FIX AVAILABLE |
| D56 Change formatting of report. Please see scan. | High | Minor | TCD56.1 | Sprint 6 | Client | Mikhail Levkovsky, Patrick Modafferi | 7.50 | 4/7/2012 | Closed |
| D57 Change report file type to an excel capable file. | High | Minor | TCU-01 | Sprint 6 | Client | Mikhail Levkovsky, Patrick Modafferi | 14.00 | 4/7/2012 | Closed |
| D58 Amount Field in Event produces error if left blank. Remove 0. | High | Significant | TCU-01 | Sprint 6 | Client | Cynthia Donato | 0.50 | 4/7/2012 | Closed |
| D59 Client File Add Flag button should be with notes. | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 0.25 | 4/5/2012 | Closed |
| D60 Populate postal code table with the list given by client. | High | Minor | TCU-01 | Sprint 6 | Client | Adrian Lloyd | 0.50 | 4/7/2012 | Closed |
| D61 Populate countries table with world countries | Low | Minor | TCU-01 | Sprint 6 | Client | Adrian Lloyd | 0.50 | 4/7/2012 | Closed |
| D62 AJAX error when deleting income. | High | Significant | STC02 | Sprint 6 | Katrina Anderson | Mikhail Levkovsky | 0.25 | 4/10/2012 | Closed |
| D63 Populate languages table with world languages | Low | Minor | N/A | Sprint 6 | Client | Adrian Lloyd | 0.75 | 4/7/2012 | Closed |
| D64 Opening calendar widget requires mouse. | Low | Minor | STC03 | Sprint 6 | Katrina Anderson | Josh Hum | 3.00 | 4/8/2012 | Closed |
| D65 Event creation form button should say "next define appointment timeslots" instead of "Create" | Medium | Minor | TCU-01 | Sprint 6 | Client | Josh Hum, Katrina Anderson | 0.25 | 4/8/2012 | Closed |
| D67 Client age should not be autoset to 12. | Medium | Minor | TCUI-02 | Sprint 6 | Josh Hum | Mikhail Levkovsky | 0.25 | 4/7/2012 | Closed |
| D68 Conflict on addresses that are left blank/blank fields should be ignored. | Medium | Minor | STC02 | Sprint 6 | Patrick Modafferi | Mikhail Levkovsky | 0.50 | 4/6/2012 | Closed |
| D69 Income and Phone Number have "0" in the field and crash if the field is left blank | High | Significant | TCU-01 | Sprint 6 | Client | Cynthia Donato | 0.50 | 4/7/2012 | Closed |
| D70 Date Validation in Income Section. | Medium | Minor | TCU-01 | Sprint 6 | Client | Patrick Modafferi | 1.00 | 4/5/2012 | Closed |
| D71 Time Validation in Event Templates. | Medium | Minor | STC03 | Sprint 6 | Katrina Anderson | Patrick Modafferi | 1.00 | 4/5/2012 | Closed |
| D72 Date and Time Validation in Event Occurrence | Medium | Minor | TCU-01 | Sprint 6 | Client | Patrick Modafferi | 0.50 | 4/5/2012 | Closed |
| D73 All links should be converted to buttons | Low | Minor | TCU-01 | Sprint 6 | Client | Josh Hum | 0.75 | 4/8/2012 | Closed |
| D74 Date Validation for Reports Generation | Medium | Minor | TCU-01 | Sprint 6 | Client | Patrick Modafferi | 0.50 | 4/5/2012 | Closed |

# ****Appendix A References****

[1] W3C. (2012, January 15). *HTML & CSS Standards*[Online]. Available:<http://www.w3.org/standards/webdesign/htmlcss#specifications>

**[2] The jQuery Project. (2012, January 15). jQuery [Online]. Available:**<http://jquery.com/>

[3] Yii Software LLC. (2012, January 15) *Yii PHP Framework* [Online]. Available: [*http://www.yiiframework.com/*](http://www.yiiframework.com/)

[4] The PHP Group. (2012, January 15) *PHP* [Online]. Available: <http://www.php.net/>

[5] C. Larman. *Applying UML and Patterns*, 3rd edition. Prentice Hall. 2005.

[6] Oracle Corporation. (2012, January 12) *Netbeans IDE 7.1* [Online] Available: http://netbeans.org/index.html.

[7] Yii Software LLC. (2012, January 12) *Yii Framework; Unit Testing.*[Online] Available: http://www.yiiframework.com/doc/guide/1.1/en/test.unit

**[8] D. Rethans.** (2012, January 12) **XDebug Extension For PHP [Online]** Available: http://xdebug.org/

**[9] IEEE.** (2012, January 12) *Test Plan Outline (IEEE 829 Format*) [Online] Available: http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#9

[10] M. Pichler. (2012,January 12) *PHP Depend; Documentation* [Online] Available : http://pdepend.org/documentation/getting-started.html

[11] O. Ormandjieva. (2012, February 24) *Lecture14;Integration Testing* [Online]: Available: http://users.encs.concordia.ca/~s345\_4/

**[12] Selenium Project. (2012, February 23). Selenium Documentation [Online] Available :** http://seleniumhq.org/docs/

[13] CSS-Tricks. (2012, March 23). *A web design community* [Online]. Available: http://css-tricks.com/label-placement-on-forms/