System Test Plan

Cowboys of Code

Kyle Fowler, Nick Whitney, Francis Kurevija, Francisco Granados

Table of Contents

[Testing Goals 3](#_Toc351713837)

[Key Areas 4](#_Toc351713838)

[Testing Deliverables 5](#_Toc351713839)

[Testing Methods 6](#_Toc351713840)

[Types of Tests 7](#_Toc351713841)

[Roles and Responsibilities 10](#_Toc351713842)

[Prerequisites to begin tests 11](#_Toc351713843)

[Test Environment 12](#_Toc351713844)

[Assumptions 13](#_Toc351713845)

[On Success 14](#_Toc351713846)

[On Failure 15](#_Toc351713847)

[Glossary 16](#_Toc351713848)

[Appendix A 17](#_Toc351713849)

# Testing Goals

Goals achieved by this test plan are as follows

* to reach 100% correct code
* to identify the test methodologies for System Testing
* to provide a procedure for System Testing
* to ensure all Functional and Design Requirements are implemented as identified
* to identify the documentation process for System Testing

# Key Areas

The key areas include the GUI, Database and core-logic. The GUI must be easy to navigate and functional. The database must be responsive and store the correct data. The core-logic must produce the correct results.

# Testing Deliverables

The following testing deliverables are produced:

1. Bug Trace-ability Matrix
2. System Tests
3. Test results
4. Test Summary Report
5. System Test Plan

# Testing Methods

## GUI

The GUI will be tested manually. This will be done by manually loading up the system and testing each page individually.

### Reports

Reports will be tested by manually navigating to the report section of the webpage and running the type of report that is being tested. The tester will then compare the output results to what they expect the report should look like and contain.

### Employee Creation

Employee creation will be manually tested by entering the employee creation page. More than one of each employee type will be added then manually tested by searching for each employee that was added.

### Employee Searching

Employee searching will tested during the testing of employee creation. After employees are created searching will be the way we look to see it they were actually created.

### User Creation

User creation will be tested manually by logging in as an admin then adding in new users. The newly created users will be tested by logging out as admin then trying to log in as the newly created user.

### Audit Table

The table can be tested by simply performing one of each action that should be logged then navigating to the audit table to see if they were in fact logged.

### Editing Employees

Employee editing will be tested by logging in as an admin then searching for an employee to edit. The tester will then make some changes to the employee and check to see if they changes were in fact made.

### Time Card Entry

Time card entry will be tested by logging in as a general user then navigating to the time card entry page. The tester will then make some time card entries for the employee, save the changes and leave the page. The tester will then return to the time card entry page to make sure the entries were saved that were entered.

## Business Logic

### Input Validation

The tester shall navigate to the pages that take in user input. The tester will then try to enter some invalid data and see how the system responds. The types of tests being covered is covered in detail in the next section of this document (Types of Tests).

# Types of Tests

## Database

### Reliability

#### How is this being tested

The reliability of the database is being tested by adding information to the database then checking if the information was actually added.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Authorization

#### How is this being tested

Authorization of the getting data from the database is tested by logging in as an admin then making sure only the functionality an admin can use is exposed. The same test will be done for general users.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Access Control

#### How is this being tested

Access control is being tested much the same as authorization.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Correctness

#### How is this being tested

Correctness of the database is being tested by checking that information added to the database is actually stored correctly.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Performance

#### How is this being tested

Performance of the database will be tested by querying using the numerous search functions built into the website and seeing that the information can be retrieved in a reasonable amount of time.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

## Core Logic

### Reliability

#### How is this being tested

Throughout numerous tests the reliability of our core logic will be tested. By running tests over and over we will see that the same results will come from our testing proving that our core logic is reliable.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Authorization

#### How is this being tested

Authorization of the being able to only use the parts of the core logic that an admin should be able to use will be tested. This will be tested by logging in as admin then making sure only those parts of the core logic an admin can use are exposed. A similar test will be done for general users.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Audit Trail

#### How is this being tested

The audit trail will be tested by performing some actions on the system then viewing the audit table to make sure all these actions have been recorded correctly.

#### Test Cases that touch on this area

EMS-ST8

### Access Control

#### How is this being tested

Access control is being tested much the same as authorization.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Correctness

#### How is this being tested

The correctness of the core logic will be tested through functional testing. A list of requirements will be tested for by performing numerous system tests.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Maintainable

#### How is this being tested

The maintainability of the core logic will be tested by manually inspecting the code and making sure everything is coded to be as maintainable as possible. Also that all parts of system have been well documented.

#### Test Cases that touch on this area

none

### Coupling

#### How is this being tested

The coupling of the core logic will be tested by manually inspecting the code and making sure everything is coded to be as loosely coupled as possible.

#### Test Cases that touch on this area

none

### Performance

#### How is this being tested

Performance of the core of the core logic will be tested by running numerous tests on the system to make sure functions are executed in a reasonable amount of time.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

## GUI

### Correctness

#### How is this being tested

Correctness of the GUI will be manually tested by browsing from page to page and making sure everything is being displayed correctly.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Ease of Use

#### How is this being tested

Ease of use of the GUI will be tested throughout the entire testing process. We will pay careful attention to making sure all operation the system must be able perform are easy to find and execute for the user.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

### Ease of Operation

#### How is this being tested

Ease of operation will be tested in the same way as Ease of Use.

#### Test Cases that touch on this area

EMS-ST1, EMS-ST2, EMS-ST3, EMS-ST4, EMS-ST5, EMS-ST6, EMS-ST7, EMS-ST8, EMS-ST9, EMS-ST10

# Roles and Responsibilities

|  |  |
| --- | --- |
| Person | Responsibilities |
| Kyle Fowler | Executing tests |
| Nick Whitney | Setting up and maintaining test environment |
| Francisco Granados | Preparing test data |
| Francis Kurevija | Analysing test results |

# Prerequisites to begin tests

1. All components must be completed to begin testing
2. All components must be unit tested
3. All components must pass unit testing
4. All components must be successfully integrated
5. All components must be integration tested
6. All subsystems must pass integration testing
7. An environment similar to the production environment must be created for testing

# Test Environment

* Windows 7 or later (any distribution)
* Visual Studio 2012 (any version)
  + EMS-PSS is being built in VS 2012, testing must be done in this IDE
* Microsoft SQL Server Express CE
  + SQL Server Express CE is the database engine built into VS 2012 ASP.NET MVC 3 projects
* Google Chrome
  + We will be adding HTML5 features that are not supported by other web browsers

# Assumptions

1. The system will be fully coded by the time the system tests are being performed.
2. All team members will be taking part in testing.
3. Unit and integration testing will be already be done by the time these test are being performed.
4. Each person testing will be using the same version of visual studio and browser to perform the testing.
5. All bugs found will be recorded and made aware to all team mates.
6. All bugs will be fixed.
7. A thorough job of testing will be performed by each team member.

# On Success

Test Cases that meet our success criteria will be signed off to proceed to the next level of testing if there is no more tests to be administered the PM will sign off on that type of test and an acceptance test will be the next step there will be 8 processes to determine if the test has passed or failed.

1. Test certain requirements
   1. Gather requirements that will define if the test succeeds
2. Identify the type of test to be used
   1. Which part of the system will this test execute
3. Ensure test data are adequate to verify test cases
   1. Double check test data will verify test
4. What result is expected and Document excepted results
   1. Fill out a test case document
5. Perform test
6. Document the result
7. Sign off
   1. Get PM’s approval
8. Acceptance test
   1. User will verify the system requirements are met

# On Failure

Test cases that do not meet our pass criteria will be documented as failures with detailed documentation of why the test failed and what measures will be taken to ensure the test case does not fail again. Test cases that fail to meet the requirements will have to follow a test case fail process this process will consist of 6 steps.

1. Test did not succeed
   1. If the test did not pass a specific test this process will be implemented
2. Document why the tests did not succeed
   1. Fill out test case documenting why the test did not succeed
3. Document changes to make to the source code
4. Write a defect report
5. Flag where the test failed
6. Come up with how to fix issue
   1. Document in your report what can be done to resolve the bug
7. Regression test entire system
   1. Run every test from the beginning to ensure the fixed issues do not affect other parts of the system

# Glossary

EMS-STx – Employee Management System-System Test x (x = the number uniquely identifying the test)

Browser – software used to view a website such as internet explorer

Bugs – an error or problem found in the system

The system – the software project being test, Employee management system

PM – Project Manager

# Appendix A

**Test Case ID** EMS-ST1

**Test Case Description** Running Seniority Report

For this test the functionality of being able to run a seniority report is being tested. This will be testing the GUI, the Database and the core logic of the system.

**How the test will be performed**

1. The tester will perform this test by adding some employees to the system.
2. They will then log in as an admin or general user, navigate to the report page and run the report.
3. The tester will then manually inspect that the report being displayed matches what they expected output was correct.

**Input Data**

1. Employees of each type added through either a database query or through the GUI of the system
2. Login credentials
3. Report page navigated to by clicking the report button the navigation bar
4. Run seniority report button clicked

**Family of Test**

Functional

**Types of Tests**

1. Requirements
2. Manual Support
3. Intersystem
4. Error Handling
5. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The report displays all employee that exist in the correct order and with all the information that should be displayed by this report.

**Actual Result**

*Test Passed -* All employees for the selected company are displayed in the correct order with the correct information.

**Test Case ID** EMS-ST2

**Test Case Description** Running Weekly Hours Worked Report

For this test the functionality of being able to run a weekly hours worked report is being tested. This will be testing the GUI, the Database and the core logic of the system.

**How the test will be performed**

1. The tester will perform this test by adding some employees to the system.
2. They will then log in as a general user and navigate to the time card entry page.
3. From this page the tester will select numerous users one at a time and enter time card information.
4. They will then log in as an admin or general user, navigate to the report page and run the report.
5. The tester will then manually inspect that the report being displayed matches what they expected output was correct.

**Input Data**

1. Employees of each type added through either a database query or through the GUI of the system
2. Login credentials
3. Time card entry page navigated to by clicking the time card entry button on the navigation bar
4. Users selected using the time card entry page
5. Time card information entered using the time card entry page for specific employees
6. Report page navigated to by clicking the report button the navigation bar
7. Run weekly hours worked report button clicked

**Family of Test**

Functional

**Types of Tests**

1. Requirements
2. Manual Support
3. Intersystem
4. Error Handling
5. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The report displays all employee that exist and with all the information that should be displayed by this report.

**Actual Result**

*Test Passed -* All employees for the selected company are displayed with their correct weekly hours.

**Test Case ID** EMS-ST3

**Test Case Description** Correcting incomplete employees – Admin

The purpose of this test is to demonstrate that an admin user can finalize incomplete full time, part time and seasonal employees. This case will test the ability to logon as an admin user, verify the information of a full time, part time and seasonal employee and complete the process of adding a valid employee to the database. The functional area this test is intended for is the core logic, database and GUI.

**How the test will be performed**

1. The tester will generate a database script to add a part time, full time and seasonal employee who’s status is incomplete
2. The tester will log in as an administrative user
3. The tester will select a task to be completed using the admin home page and complete the employees data
4. The tester will visually inspect the database to ensure the employee has be corrected

**Input Data**

1. Generate a SQL script to add incomplete employees
2. Login as a admin user
3. Select an employee to be completed using the task list
4. Verify employee data
5. Select complete employee
6. Completed employee is added to the database

**Family of Test**

Structural, Functional

**Types of Tests**

1. Compliance
2. Security
3. Requirements
4. Manual Support
5. Intersystem
6. Error Handling
7. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The completed employee will be added to the database with a status of complete instead of incomplete, the admin user will have a populated task list with employees whose status is incomplete when they first login.

**Actual Result**

*Test Passed -* The status changed in the database for each employee type on the specified field from false to true, adding employees from as a general user set the status to inactive when logging in as an admin a list of tasks that needed to be completed appeared with a link to allow an admin to complete. When an admin saved all required fields the status changed from incomplete to complete

**Test Case ID** EMS-ST4

**Test Case Description** Adding Employees to a new Company

The purpose of this test is to verify that you can add a new company and that company can have multiple employees associated with the company. This test touches the core logic, database and the GUI this test will verify that a company can be associated with multiple employees.

**How the test will be performed**

1. The tester will log in as an administrative user
2. The tester will navigate to the manage employee page and add the name of the company
3. These test will run a script to populate the employees database with a part time, full time, seasonal and contract employee
4. The tester will manually inspect the company’s employee listing to ensure the employees have been added

**Input Data**

1. Login as admin user
2. Navigate to manage employees page
3. Add a company
4. Run a script to add employees
5. Associate employee to the company

**Family of Test**

Functional

**Types of Tests**

1. Requirements
2. Manual Support
3. Intersystem
4. Error Handling
5. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The employees will add that company to their company field the company can have multiple types of employees associated with it.

**Actual Result**

*Test Passed -* Created a company and associated that company to a part time, full time, seasonal and contact employee I then proceeded to run a seniority report to ensure these employee were associated with the new company

**Test Case ID** EMS-ST5

**Test Case Description** Creating new users

The purpose of this test is to verify that an admin user can create more users (both admin and general). We plan to test is by using an admin will to another admin which will create a general user. This test will touch the core logic, database and GUI functional areas of the system.

**How the test will be performed**

1. The tester will login and navigate to manage users page
2. The tester will create a new admin user which is added to the database then log off
3. The tester will login as the new admin user and navigate to manage user page
4. The tester will create a new general employee then log off
5. The tester will login as the new general employee
6. Through these tests the tester will ensure that all users are properly created, with appropriate privileges

**Input Data**

1. Logon as admin user
2. Navigate to manage user tab
3. Create admin user
4. Logon as second admin user
5. Navigate to manage user tab
6. Create general user

**Family of Test**

Structural, Functional

**Types of Tests**

1. Compliance
2. Security
3. Requirements
4. Manual Support
5. Intersystem
6. Error Handling
7. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The admin created users will be created and added to the database they will be able to login to the system with the correct permissions.

**Actual Result**

*Test Passed -* A new admin user was created and added to the database I then used that user to login with my password and double checked that all the data was correct I created a general user with my new admin user and logged off to login as the new user and checked security level everything ran as expected.

**Test Case ID** EMS-ST6

**Test Case Description** Searching for created employees of each type

For this test, the functionality of searching for each type of created employee within the system is being tested. This will be testing the GUI, the Database and the core logic of the system.

**How the Test Will Be Performed**

1. The tester will perform this test by logging in as either a general or administrative user
2. The tester will fill in the search form and submit the form
3. The tester will then manually inspect that the employees returned from the search match the search criteria and that all employees in the database matching the criteria have been returned

**Input Data**

1. Login credentials
2. User fills out fields of search form
3. User submits search form
4. User is redirected to the Search Results page
5. List of employees returned from the search displayed in a table

**Family of Test**

Functional

**Types of Tests**

1. Requirements
2. Manual Support
3. Intersystem
4. Error Handling
5. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The Search Results display all employees that match the user’s search criteria.

**Actual Result**

*Test Passed -* All employees in the database matching the search criteria were returned and no others.

**Test Case ID** EMS-ST7

**Test Case Description** Running Active Employees Report

For this test, the functionality of being able to run a report of active employees is being tested. This will be testing the GUI, the Database and the core logic of the system.

**How the Test Will Be Performed**

1. The tester will perform this test by adding some active and inactive employees to the system
2. The tester will then log in as an administrative user and navigate to the Reports page
3. From the Reports page the tester will select the “Active Employee Report” and click the “Run Report” button
4. The tester will then manually inspect that employees being displayed in the report match all known active employees

**Input Data**

1. Active and Inactive employees of each type added through either a database query or through the GUI of the system
2. Login credentials
3. Reports page navigated to by clicking the Run Reports button on the navigation bar
4. User selects “Active Employees” report
5. Run Report button clicked

**Family of Test**

Structural, Functional

**Types of Tests**

1. Compliance
2. Security
3. Requirements
4. Manual Support
5. Intersystem
6. Error Handling
7. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The report displays all active employees contained within the system and no inactive employees.

**Actual Result***Test Passed -* All known active employees for the selected companies are displayed and no others.

**Test Case ID** EMS-ST8

**Test Case Description** Verifying audit table integrity

For this test, the integrity of the audit table is being tested. This will be testing the GUI, the Database and the core logic of the system

**How the Test Will Be Performed**

1. The tester will perform this test by adding an employee to the system
2. The tester will then log in as an administrative user and navigate to the Edit Employee page
3. From the Edit Employee page the tester will edit the test employee’s information so that data will be added to the Audit Table
4. The tester will then navigate to the Audit Table page
5. On the Audit Table page the tester will select a date frame to display audits for and press the View button
6. The tester will then manually inspect that the change he made to the employee appear in the audit table at the correct time

**Input Data**

1. A test employee added through either a database query or through the GUI of the system
2. Login credentials
3. Edit Employee page navigated to by clicking the Manage Employees button on the navigation bar
4. Edit one or more fields of the existing test employee and save changes
5. Navigate to the Audit Table page by clicking the Audit Table button on the navigation bar
6. Enter the date frame for when the change was make and click View

**Family of Test**

Structural, Functional

**Types of Tests**

1. Operations
2. Compliance
3. Security
4. Requirements
5. Manual Support
6. Intersystem
7. Error Handling
8. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The Audit Table displays the changes made to the test employee with all the correct details

**Actual Result**

*Test Passed -* All changes made to different types of employees appear in the audit table.

**Test Case ID** EMS-ST9

**Test Case Description** Running Payroll Report

For this test the functionality of being able to run a payroll report is being tested. This will be testing the GUI, the Database and the core logic of the system.

**How the Test Will Be Performed**

1. The tester will perform this test by adding some employees to the system
2. The tester will then log in as a general user and navigate to the time card entry page
3. From the time card entry page the tester will select numerous users one at a time and enter time card information
4. The tester will then log in as an admin or general user, navigate to the report page and run the report
5. The tester will then manually inspect that the report being displayed matches what they expected output was correct

**Input Data**

1. Employees of each type added through either a database query or through the GUI of the system
2. Login credentials
3. Time card entry page navigated to by clicking the time card entry button on the navigation bar
4. Users selected using the time card entry page
5. Time card information entered using the time card entry page for specific employees
6. Report page navigated to by clicking the report button the navigation bar
7. Run payroll report button clicked

**Family of Test**

Structural, Functional

**Types of Tests**

1. Operations
2. Compliance
3. Security
4. Requirements
5. Manual Support
6. Intersystem
7. Error Handling
8. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The report displays all employee that exist and with all the information that should be displayed by this report.

**Actual Result**

*Test Passed -* All Employees shown with correct notes and pay.

**Test Case ID** EMS-ST10

**Test Case Description** Editing an employee that was just created or that already existed

For this test the functionality of being able to run a payroll report is being tested. This will be testing the GUI, the Database and the core logic of the system

**How the Test Will Be Performed**

1. The tester will perform this test by adding some employees to the system
2. The tester will search up the employees, as well as some preexisting employees and then edit them
3. The tester will then ensure the updated info is existing

**Input Data**

1. Create SQL script to create multiple employees in the database.
2. Log in
3. Create multiple employees
4. Search up employees in database and newly created
5. edit those employees
6. Search and verify changes held

**Family of Test**

Functional

**Types of Tests**

1. Security
2. Requirements
3. Manual Support
4. Intersystem
5. Error Handling
6. Control

**Functional Areas**

1. GUI
2. Database
3. Core Logic

**Expected Result**

The employees should show the changed data.

**Actual Result**

*Test Passed -* Employees show the change data