**Software Test Plan**



Attendance System

Table of Contents

Table of Contents 3

1 Introduction 4

2 Goals and exit criteria 5

2.1 Quality goals that needs to be met for the test phase to exit 5

2.2 Robustness goals of the product 5

2.3 Schedule goals of the project 5

2.4 Performance and efficiency goals of the product 5

3 Items to be tested/ inspected 6

3.1 Executables such as modules and components 6

3.2 Nonexecutable such as requirements specification or design specification 6

4 Test process/ methodologies 7

4.1 Unit test/ functional test/ acceptance test/ regression test/ and so on, methodologies. 7

4.2 Inspection/ reviews methodologies 7

4.3 Black-box testing (e.g., input domain test, boundary value testing) 7

4.4 White-box testing (e.g., control path testing, data flow testing ) 7

4.5 Test metrics(e.g., code coverage, branch coverage, number of problem by severity) 7

4.6 Test-bug report-fix-retest process 7

5 resources 8

5.1 People(number of, skills , etc) 8

5.2 Tools (for measurement, defect management, etc.) 8

5.3 Systems (test execution platform, test case development, etc.) 8

6 Schedule 9

6.1 Test-case development 9

6.2 Test execution 9

6.3 Problem reporting and fixing 9

7 Risks 10

7.1 Missing goals 10

7.2 Back-up resources needed 10

8 Major test scenarios and test cases 11

8.1 Boundary value and input domain test cases 11

8.2 Control path and dataflow test cases 11

8.3 Integration and intermodular test cases 11

# Introduction

Acquiring evidence that a product is of sufficient quality with a reasonable level of confidence is the primary objective of the testing process.

This test plan will describe the necessary test procedures that must be undertaken and passed in order for the project to meet this demand. A selection of tests will be chosen to target specific aspects of the projects functionality. Each test will be documented in an auditable fashion such that it may be used to assess the quality of the product. Once all tests have achieved a satisfactory pass mark the results will provide a level of assurance to the customer and developer that the product is fit for purpose.

In order for the tests to be considered valid, they must be performed with certain constraints. The tests must be performed under conditions that reflect those similar to an end user. In this instance, that will mean a clean operating system devoid of developer tools and libraries. To achieve this goal, the testing will be performed from inside a virtual machine environment with a clean install of the specified operating system.

The specific functional and non-functional requirements that should be targeted by the tests are described in the Software Development Plan and the Software Requirements and Design Document.

# Goals and exit criteria

## Quality goals that needs to be met for the test phase to exit

All test goals must be fulfilled under the constraints before the test phase may be exited.

## Robustness goals of the product

## Schedule goals of the project

## Performance and efficiency goals of the product

# Items to be tested/ inspected

## Executables such as modules and components

The items to be tested are the Checkpoint web application software, the Checkpoint native application software and the associated Reading-terminal hardware.

The specific features to be tested are as follows:

|  |  |
| --- | --- |
| Checkpoint Web App  Feature / Functionality | SRD Design Requirement Reference  Use Case Documents |
| Client Registration | Chapter 6.2.1 Register |
| Log In | Chapter 6.2.2 Log In |
| Attendee Registration | Chapter 6.2.3 Become Attendee |
| Manage Profile | Chapter 6.2.4 Manage Profile |
| Create Appointment | Chapter 6.2.5 Create Appointment |
| Manage Appointment | Chapter 6.2.6 Manage Appointment |
| Create Course | Chapter 6.2.7 Create Course |
| Manage Course | Chapter 6.2.8 Manage Course |
| Create Report | Chapter 6.2.10 Create Report |

|  |  |
| --- | --- |
| Checkpoint Web App  Performance | SRD Design Requirement Reference  Technical Requirements |
| Browser Support Chrome, Firefox, Opera | Chapter 3.1.1 Functional Requirements |
| Support Access by Multiple Simultaneous Users | Chapter 3.1.1 Functional Requirements |

|  |  |
| --- | --- |
| Checkpoint Native App  & Hardware Reading-terminal  Feature / Functionality | SRD Design Requirement Reference  Use Case Documents |
| Attend Appointment | Chapter 6.2.9 Attend Appointment |

|  |  |
| --- | --- |
| Checkpoint Native App  & Hardware Reading-terminal  Performance | SRD Design Requirement Reference |
| Attendance Registration Time (within 4 secs) | Chapter 3.2 Non-Functional Requirements |

## Nonexecutable such as requirements specification or design specification

# Test process/ methodologies

## Unit test/ functional test/ acceptance test/ regression test/ and so on, methodologies.

## Inspection/ reviews methodologies

## Black-box testing (e.g., input domain test, boundary value testing)

## White-box testing (e.g., control path testing, data flow testing )

## Test metrics(e.g., code coverage, branch coverage, number of problem by severity)

## Test-bug report-fix-retest process

# resources

## People(number of, skills , etc)

## Tools (for measurement, defect management, etc.)

## Systems (test execution platform, test case development, etc.)

# Schedule

## Test-case development

## Test execution

## Problem reporting and fixing

# Risks

## Missing goals

## Back-up resources needed

# Major test scenarios and test cases

## Boundary value and input domain test cases

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Browser Compatibility | OK | FAIL | Description |
| Functionality in Chrome |  |  |  |
| Functionality in Firefox |  |  |  |
| Functionality in Opera |  |  |  |
| Functionality in IE10 |  |  |  |
| Functionality in Edge |  |  |  |
| GUI is rendered correctly in Chrome |  |  |  |
| GUI is rendered correctly in Firefox |  |  |  |
| GUI is rendered correctly in Opera |  |  |  |
| GUI is rendered correctly in IE10 |  |  |  |
| GUI is rendered correctly in Edge |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Login Function | OK | FAIL | Description |
| Login succeeds with valid data |  |  |  |
| Login fails with invalid data |  |  |  |
| Login recognizes correct client type |  |  |  |
| Login stores client to session |  |  |  |
| Login response to null fields |  |  |  |
| Login hides password |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Register Client Function | OK | FAIL | Description |
| Registration succeeds with valid data |  |  |  |
| Registration fails with invalid password length (< 6 chars) |  |  |  |
| Registration fails with duplicate username |  |  |  |
| Registration fails with invalid postcode  (does not exist) |  |  |  |
| Registration fails with invalid email  format |  |  |  |
| Registration fails with invalid phone number (< 8 digits / non-integer) |  |  |  |
| Registration fails with null fields |  |  |  |
| Registration response to invalid data |  |  |  |
| Registration data stored correctly in database |  |  |  |
| Response if registration fails |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Create Appointment Function | OK | FAIL | Description |
| New Appointment creation with valid data is successful |  |  |  |
| Creation fails with null fields |  |  |  |
| Creation fails with invalid time |  |  |  |
| Creation fails with invalid date |  |  |  |
| Creation fails with invalid postcode  (does not exist / non-integer) |  |  |  |
| Appointment is stored correctly in database |  |  |  |
| Response if creation fails |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Update Appointment Function | OK | FAIL | Description |
| Appointment updated with valid data is successful |  |  |  |
| Update fails with null fields |  |  |  |
| Update fails with invalid time |  |  |  |
| Update fails with invalid date |  |  |  |
| Update fails with invalid postcode  (does not exist / non-integer) |  |  |  |
| Appointment is updated correctly in database |  |  |  |
| Response if update fails |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Delete Appointment Function | OK | FAIL | Description |
| Appointment is deleted |  |  |  |
| Deletion is updated correctly in database |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Create Course Function | OK | FAIL | Description |
| New course creation with valid data is successful |  |  |  |
| Creation fails with null fields |  |  |  |
| Creation fails with invalid time |  |  |  |
| Creation fails with invalid date |  |  |  |
| Creation fails with invalid postcode  (does not exist / non-integer) |  |  |  |
| Course is stored correctly in database |  |  |  |
| Creation fails if no appointment is associated to course |  |  |  |
| Response if creation fails |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Add New Appointment To Course Function | OK | FAIL | Description |
| New appointment added with valid data is successful |  |  |  |
| Adding fails with null fields |  |  |  |
| Adding fails with invalid time |  |  |  |
| Adding fails with invalid date |  |  |  |
| Adding fails with invalid postcode  (does not exist / non-integer) |  |  |  |
| Adding can be abandoned successfully |  |  |  |
| Course is stored with new appointment correctly in database |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Add Existing Appointment To Course Function | OK | FAIL | Description |
| Existing appointment is added successfully |  |  |  |
| Adding fails with null fields |  |  |  |
| Adding fails with invalid time |  |  |  |
| Adding fails with invalid date |  |  |  |
| Adding fails with invalid postcode  (does not exist / non-integer) |  |  |  |
| Adding can be abandoned successfully |  |  |  |
| Course is stored with existing appointment correctly in database |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Host Home Page Function | OK | FAIL | Description |
| Home page table is displayed successfully |  |  |  |
| Sorting appointment data functions correctly |  |  |  |
| Create new appointment button navigates to correct page |  |  |  |
| Manage appointment button navigates to correct page |  |  |  |
| View courses button navigates to correct page |  |  |  |
| Manage attendance button navigates to correct page |  |  |  |
| Create report button navigates to correct page |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Host Courses Page Function | OK | FAIL | Description |
| Courses page table is displayed successfully |  |  |  |
| Sorting course data functions correctly |  |  |  |
| Create new course button navigates to correct page |  |  |  |
| Manage courses button navigates to correct page |  |  |  |
| View appointments button navigates to correct page |  |  |  |
| Manage attendance button navigates to correct page |  |  |  |
| Create report button navigates to correct page |  |  |  |

Tester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| User Home Page Function | OK | FAIL | Description |
| Home page table is displayed successfully |  |  |  |
| Sorting appointment data functions correctly |  |  |  |
| Find appointment button navigates to correct page |  |  |  |
| Find courses button navigates to correct page |  |  |  |
| View courses button navigates to correct page |  |  |  |
| Apply to appointment button navigates to correct page |  |  |  |
| Apply to course button navigates to correct page |  |  |  |

## Control path and dataflow test cases

## Integration and intermodular test cases