|  |
| --- |
|  |
| Scorekeeper Scheduling System Test Plan |
| MSSE 696 Software Practicum II, Regis University |
|  |
| **Carri Martin** |
| **8/23/2018** |

|  |
| --- |
|  |

**Table of Contents**

[**1.0** **Introduction** 3](#_Toc522875074)

[**1.1** **Nature of the Project** 3](#_Toc522875075)

[**1.2** **System Test Objectives and Scope** 3](#_Toc522875076)

[**1.2.1** **Test Objectives** 3](#_Toc522875077)

[**1.2.2** **Testing Scope** 4](#_Toc522875078)

[**1.3** **Related Documents** 4](#_Toc522875079)

[**2.0** **Items to Be Tested** 4](#_Toc522875080)

[**3.0** **Features** 4](#_Toc522875081)

[**3.1** **Features to Be Tested** 4](#_Toc522875082)

[**3.2** **Features Not to Be Tested** 5](#_Toc522875083)

[**4.0** **Approach** 5](#_Toc522875084)

[**4.1** **Sources of Sample Domain Data and Test Oracles** 5](#_Toc522875085)

[**4.2** **Development Staff** 5](#_Toc522875086)

[**4.3** **Test Status** 5](#_Toc522875087)

[**4.4** **Test Tools** 5](#_Toc522875088)

[**4.5** **Stop-Test Criteria** 6](#_Toc522875089)

[**4.6** **Types of Tests** 6](#_Toc522875090)

[**4.7** **Use Case Outline** 6](#_Toc522875091)

[**5.0** **Test Cases** 7](#_Toc522875092)

[5.1 Add Event 7](#_Toc522875093)

[5.1.1 User Story and Acceptance Criteria 8](#_Toc522875094)

[5.1.2 Input Specification 8](#_Toc522875095)

[5.2 Add Scorekeeper 11](#_Toc522875096)

[5.2.1 User Story and Acceptance Criteria 12](#_Toc522875097)

[5.2.2 Input Specification 12](#_Toc522875098)

[5.3 Remove Event 14](#_Toc522875099)

[5.3.1 User Story and Acceptance Criteria 15](#_Toc522875100)

[5.4 Remove Scorekeeper 15](#_Toc522875101)

[5.4.1 User Story and Acceptance Criteria 16](#_Toc522875102)

[5.5 Current Event List 16](#_Toc522875103)

[5.5.1 User Story and Acceptance Criteria 17](#_Toc522875104)

[5.6 Previous Event List 18](#_Toc522875105)

[5.6.1 User Story and Acceptance Criteria 18](#_Toc522875106)

[5.7 Next Event List 19](#_Toc522875107)

[5.7.1 User Story and Acceptance Criteria 20](#_Toc522875108)

[**6.0** **Pass/Fail Criteria** 20](#_Toc522875109)

[**7.0** **Suspend/Resume Criteria** 20](#_Toc522875110)

[**8.0** **Test Deliverables** 20](#_Toc522875111)

[**9.0** **Testing Tasks** 21](#_Toc522875112)

[**10.0** **Environmental Requirements** 22](#_Toc522875113)

[**10.1** **Lab Setup** 22](#_Toc522875114)

[**10.2** **Publications** 22](#_Toc522875115)

[**10.3** **Tools** 22](#_Toc522875116)

[**11.0** **Responsibilities** 22](#_Toc522875117)

[**12.0** **Resources** 22](#_Toc522875118)

[**13.0** **Scheduling** 23](#_Toc522875119)

[**14.0** **Risks and Contingencies** 23](#_Toc522875120)

[**15.0** **Software Configuration Management** 23](#_Toc522875121)

[**Appendix A. Defect Flow** 24](#_Toc522875122)

[**Appendix B. Peer Review Process** 25](#_Toc522875123)

[**Appendix C. Internal Testing Strategies** 26](#_Toc522875124)

[**C.1** **Test Automation** 26](#_Toc522875125)

[**Scope** 26](#_Toc522875126)

[**Test Management** 26](#_Toc522875127)

[**Test Lifecycle** 26](#_Toc522875128)

[**C.2** **Continuous Integration Strategy** 26](#_Toc522875129)

[**Scope** 26](#_Toc522875130)

[**Tools** 26](#_Toc522875131)

1. **Introduction**

This section introduces the software system under test providing some background, an overview, test objectives and scope.

## **1.1 Nature of the Project**

The Scorekeeper Scheduling System is a simple cloud-based application that automates the scheduling tasks for an athletic program coordinator. This system will allow the coordinator to schedule softball games for various leagues as well as the scorekeepers assigned to keep official track of the score and monitor the game. Additionally, the system will allow the coordinator to track the scorekeepers as well as add new staff members for future scheduling.

## **1.2 System Test Objectives and Scope**

This test plan describes the environment, activities, tasks, tools, and schedules for system testing the Scorekeeper Scheduling System application.

### **1.2.1 Test Objectives**

The objectives for system testing are to validate the following functional areas:

1. Event scheduling – maintain and update scheduled sporting events
2. Staff management – unique staff information is maintained and updated
3. Reporting – generated correctly on demand

The test objectives are:

1. The application meets its performance requirements
2. The application is reliable and robust. It can handle failures and recovery.
3. The application is simple and easy to use.

### **1.2.2 Testing Scope**

As this will be the initial creation of the Scorekeeper Scheduling System application this project will be a comprehensive test of the application’s functionality. As the system is still in it’s prototype phase not all functionality is expected to work at this stage.

## **1.3 Related Documents**

Valid program coordinator credentials and user name input specification

1. **Items to Be Tested**
2. Event scheduling
3. Staff management
4. Reporting capabilities
5. **Features**

## **3.1 Features to Be Tested**

#### Event scheduling

1. Adding new events to the schedule
2. Deleting events from the schedule
3. Viewing the event schedule

#### Staff

1. Adding/creating new staff
2. Deleting staff

#### Reporting

1. Validating the automatically generated event list
2. Validating the automatically generated lists of previous and future events

## **3.2 Features Not to Be Tested**

1. Future enhancements
2. **Approach**

The Scorekeeper Scheduling System development team does not have a dedicated testing group. As this is a simple application testing will be performed as development progresses. The necessary resources and test environment are present to support their testing efforts. All items will be tested thoroughly to ensure the software will function as a complete system. All test cases, specifications and scripts will be saved and used for future testing situations.

## **4.1 Sources of Sample Domain Data and Test Oracles**

Initial testing will utilize fake events and staff that have been randomly generated.

## **4.2 Development Staff**

The two main developers for the Scorekeeper Scheduling System application will be responsible for test case design, test execution, and the recording of test results.

## **4.3 Test Status**

The automated test results are available in real time for the product owner and team to observe. The status of the current test event will be reported and discussed at the team’s daily stand up meetings.

## **4.4 Test Tools**

The following tools will be used to support the system testing effort. The tools are the following:

* JUnit
* Selenium

## **4.5 Stop-Test Criteria**

The decision to stop the testing will be based on how many defects are discovered verses how many requirements are covered. The test will be completed when all automated test cases are run and all identified features are tested.

## **4.6 Types of Tests**

Functional tests will be utilized to test end-to-end functionality in several important areas of the system. The areas are related to staff creation, event management and reporting. The tests will include the valid input domain and verify invalid data is rejected.

## **4.7 Use Case Outline**

The following diagram depicts the system’s use cases as they relate to the primary actor.

Previous Event List 
Current Event List 
Next Event List 
Add Events 
Remove Events 
Program Coordinator 
Add Scorekeepers 
Remove Scorekeepers 

Figure 1: Use Case Diagram for the Scorekeeper Scheduling System

1. **Test Cases**

## 5.1 Add Event

Feature Under Test: Schedule a new event

Use Case: A program coordinator wishes to schedule a new event. This process will require the coordinator to enter event information such as the scheduled date, field assignment, league type, and scorekeeper first and last name.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he starts the add event process
2. The system prompts the coordinator to enter information about the new event such as the scheduled date, field assignment, league type, and scorekeeper first and last name

ALT FLOW 2.A1 Missing/empty/ invalid required fields

* 1. If any of the required fields are not entered the system will display a prompt to enter or correct the appropriate fields.

1. After the fields have been verified the new event is created in the database. The system displays a confirmation message that the action has been successful.

Expected Outcome Success:

New event is added to the event schedule.

* A confirmation message is displayed in the application confirming the success of the operation.
* Event information is populated to the database correctly.

Expected Outcome Failure:

New event is not added to the event schedule.

* An error message is displayed in the application.
* Event information is not populated to the database correctly.

### 5.1.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to schedule a new event to reserve the field on the date games are scheduled to play.

Acceptance Criteria: Enter the scheduled date, field the game is scheduled to play on, the game type and the first and last name of the scorekeeper assigned. Make sure this information shows up correctly on the event list.

Acceptance Criteria: Enter the necessary information for a second event. Make sure that information also shows up correctly in the event list.

### 5.1.2 Input Specification

Valid Input Specification

|  |  |
| --- | --- |
| Input Field | Valid Input |
| Scheduled Date | Date Format YYYY-MM-DD |
| Field | Alphanumeric. Min 2, max 10 |
| Game Type | Only Alpha characters. Min 2, max 50 |
| Scorekeeper First Name | Only Alpha characters. Min 2, max 255 |
| Scorekeeper Last Name | Only Alpha characters. Min 2, max 255 |

Table 1 - Valid Form Input

Test Approach

The valid input domain for this feature is too large to manually test all potential values. Because the input is straight data entry no decision or state changes will occur as a result of user selections. The functional testing will be designed using equivalence class partitioning and boundary value analysis. The following table lists the equivalence classes defined after equivalence class partitioning analysis. Table 3 contains the boundary values defined for this feature.

|  |  |  |
| --- | --- | --- |
| EC | Description | Valid/Invalid |
| 1 | Scheduled Date contains all numeric values | Valid |
| 2 | Scheduled Date contains all alpha values | Invalid |
| 3 | Scheduled Date contains numeric values in a non-date format | Invalid |
| 4 | Field contains all numeric values | Valid |
| 5 | Field contains all alpha values | Invalid |
| 6 | Field contains more than 10 numeric values | Invalid |
| 7 | Game Type contains all alpha values | Valid |
| 8 | Game Type contains all numeric values | Invalid |
| 9 | Game Type contains more than 50 alpha characters | Invalid |
| 10 | Scorekeeper First Name contains all alpha values | Valid |
| 11 | Scorekeeper First Name contains only non-alpha values | Invalid |
| 12 | Scorekeeper First Name has between 2 and 255 alpha characters | Valid |
| 13 | Scorekeeper First Name has less than 2 alpha characters | Invalid |
| 14 | Scorekeeper First Name has more than 255 alpha characters | Invalid |
| 15 | Scorekeeper First Name contains all numeric values | Invalid |
| 16 | Scorekeeper Last Name contains all alpha values and | Valid |
| 17 | Scorekeeper Last Name contains only non-alpha values | Invalid |
| 18 | Scorekeeper Last Name has between 2 and 255 alpha characters | Valid |
| 19 | Scorekeeper Last Name has less than 2 alpha characters | Invalid |
| 20 | Scorekeeper Last Name has more than 255 alpha characters | Invalid |
| 21 | Scorekeeper Last Name contains all numeric values | Invalid |

Table 2 – Equivalence Classes

|  |  |
| --- | --- |
| Boundary Group | Boundary Value |
| Length of scheduled date. Value just below the lower bound (BLB) | 0 |
| Length of scheduled date. Value on the lower boundary (LB) | 1 |
| Length of scheduled date. Value just above of lower boundary (ALB) | 2 |
| Length of scheduled date. Value just below the upper bound (BUB) | 7 |
| Length of scheduled date. Value on the upper bound (UB) | 8 |
| Length of scheduled date. Value just above the upper bound (AUB) | 9 |
| Length of field. Value just below the lower bound (BLB) | 0 |
| Length of field. Value on the lower boundary (LB) | 1 |
| Length of field. Value just above of lower boundary (ALB) | 2 |
| Length of field. Value just below the upper bound (BUB) | 9 |
| Length of field. Value on the upper bound (UB) | 10 |
| Length of field. Value just above the upper bound (AUB) | 11 |
| Length of game type. Value just below the lower bound (BLB) | 0 |
| Length of game type. Value on the lower boundary (LB) | 1 |
| Length of game type. Value just above of lower boundary (ALB) | 2 |
| Length of game type. Value just below the upper bound (BUB) | 49 |
| Length of game type. Value on the upper bound (UB) | 50 |
| Length of game type. Value just above the upper bound (AUB) | 51 |
| Length of first name. Value just below the lower bound (BLB) | 1 |
| Length of first name. Value on the lower boundary (LB) | 2 |
| Length of first name. Value just above of lower boundary (ALB) | 3 |
| Length of first name. Value just below the upper bound (BUB) | 254 |
| Length of first name. Value on the upper bound (UB) | 255 |
| Length of first name. Value just above the upper bound (AUB) | 256 |
| Length of last name. Value just below the lower bound (BLB) | 1 |
| Length of last name. Value on the lower boundary (LB) | 2 |
| Length of last name. Value just above of lower boundary (ALB) | 3 |
| Length of last name. Value just below the upper bound (BUB) | 254 |
| Length of last name. Value on the upper bound (UB) | 255 |
| Length of last name. Value just above the upper bound (AUB) | 256 |

Table 3 - Boundary Value

## 5.2 Add Scorekeeper

Feature Under Test: Add a new scorekeeper

Use Case: Coordinators will have the ability to add new scorekeepers as needed. This process will require the coordinator to enter information such as the scorekeepers’ first name and last name.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he starts the add staff process
2. The system prompts the coordinator to enter information about the new item such as the scorekeepers’ first name and last name.

ALT FLOW 2.A1 Missing/empty/ invalid required fields

* 1. If any of the required fields are not entered the system will display a prompt to enter or correct the appropriate fields.

1. After the fields have been verified the new staff member is created in the database. The system displays a confirmation message that the action has been successful.

Expected Outcome Success:

New scorekeeper is added to the system.

* A confirmation message is displayed in the application confirming the success of the operation.
* Staff information is populated to the database correctly.

Expected Outcome Failure:

New scorekeeper is not added the system.

* An error message is displayed in the application.
* Staff information is not populated to the database correctly.

### 5.2.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to add a new scorekeeper so that I may assign that person to a future event.

Acceptance Criteria: Enter the new scorekeeper’s first and last name. Make sure this information shows up correctly on the active staff list.

Acceptance Criteria: Enter the necessary information for a second scorekeeper. Make sure that information also shows up correctly in the active staff list.

### 5.2.2 Input Specification

Valid Input Specification

|  |  |
| --- | --- |
| Input Field | Valid Input |
| Scorekeeper First Name | Only Alpha characters. Min 2, max 255 |
| Scorekeeper Last Name | Only Alpha characters. Min 2, max 255 |

Table 4 - Valid Form Input

Test Approach

The valid input domain for this feature is too large to manually test all potential values. Because the input is straight data entry no decision or state changes will occur as a result of user selections. The functional testing will be designed using equivalence class partitioning and boundary value analysis. The following table lists the equivalence classes defined after equivalence class partitioning analysis. Table 6 contains the boundary values defined for this feature.

|  |  |  |
| --- | --- | --- |
| EC | Description | Valid/Invalid |
| 1 | Scorekeeper First Name contains all alpha values | Valid |
| 2 | Scorekeeper First Name contains only non-alpha values | Invalid |
| 3 | Scorekeeper First Name has between 2 and 255 alpha characters | Valid |
| 4 | Scorekeeper First Name has less than 2 alpha characters | Invalid |
| 5 | Scorekeeper First Name has more than 255 alpha characters | Invalid |
| 6 | Scorekeeper First Name contains all numeric values | Invalid |
| 7 | Scorekeeper Last Name contains all alpha values and | Valid |
| 8 | Scorekeeper Last Name contains only non-alpha values | Invalid |
| 9 | Scorekeeper Last Name has between 2 and 255 alpha characters | Valid |
| 10 | Scorekeeper Last Name has less than 2 alpha characters | Invalid |
| 11 | Scorekeeper Last Name has more than 255 alpha characters | Invalid |
| 12 | Scorekeeper Last Name contains all numeric values | Invalid |

Table 5 – Equivalence Classes

|  |  |
| --- | --- |
| Boundary Group | Boundary Value |
| Length of first name. Value just below the lower bound (BLB) | 1 |
| Length of first name. Value on the lower boundary (LB) | 2 |
| Length of first name. Value just above of lower boundary (ALB) | 3 |
| Length of first name. Value just below the upper bound (BUB) | 254 |
| Length of first name. Value on the upper bound (UB) | 255 |
| Length of first name. Value just above the upper bound (AUB) | 256 |
| Length of last name. Value just below the lower bound (BLB) | 1 |
| Length of last name. Value on the lower boundary (LB) | 2 |
| Length of last name. Value just above of lower boundary (ALB) | 3 |
| Length of last name. Value just below the upper bound (BUB) | 254 |
| Length of last name. Value on the upper bound (UB) | 255 |
| Length of last name. Value just above the upper bound (AUB) | 256 |

Table 6 - Boundary Value

## 5.3 Remove Event

Feature Under Test: Remove an event from the schedule

Use Case: Coordinators will have the ability to remove events from the schedule as needed. This process will require the coordinator to select a row from the event list that displays the correct event to remove from the system.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he selects the correct event to remove from the schedule.
2. The coordinator initiates the removal process

ALT FLOW 2.A1 Missing/empty/ invalid required fields

* 1. If none of the events are selected the system will display a prompt to select an event.

1. After the fields have been verified the event is removed from the database. The system displays a confirmation message that the action has been successful.

Expected Outcome Success:

The selected event is removed from the schedule.

* A confirmation message is displayed in the application confirming the success of the operation.
* Event information is removed from the database correctly.

Expected Outcome Failure:

The selected event is not removed from the schedule.

* An error message is displayed in the application.
* Event information is not removed from the database correctly.

### 5.3.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to remove an event from the schedule so that I may reserve the field for a different event on the same day.

Acceptance Criteria: Select the event to remove and initiate the removal process. Make sure the selected event is removed from the event list.

Acceptance Criteria: Select an event to remove and initiate the removal process. Make sure that no other events are removed from the event list.

Acceptance Criteria: Select a second event to remove and initiate the removal process. Make sure that information is also removed from the event list.

## 5.4 Remove Scorekeeper

Feature Under Test: Remove a scorekeeper form the active staff list

Use Case: Coordinators will have the ability to remove scorekeepers from the active staff list as needed. This process will require the coordinator to select the staff member from the active staff list to soft delete from the system.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he starts the add remove a scorekeeper process
2. The system provides the active staff list from which the coordinator must select a name.

ALT FLOW 2.A1 Missing/empty/ invalid required fields

* 1. If no staff member name is selected the system prompts for the selection of a staff member name.

1. After the fields have been verified the system will set the selected staff member’s active status to “No” in the database. The system displays a confirmation message that the action has been successful.

Expected Outcome Success:

The selected scorekeeper is removed from the active staff list.

* A confirmation message is displayed in the application confirming the success of the operation.
* Staff information is updated in the database correctly.

Expected Outcome Failure:

The selected scorekeeper is not removed from the active staff list.

* An error message is displayed in the application.
* Staff information is not updated in the database correctly.

### 5.4.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to remove a scorekeeper from the active list because that person is no longer available.

Acceptance Criteria: Select the scorekeeper’s name and initiate the removal process. Make sure the scorekeeper’s name is removed from the active staff list.

Acceptance Criteria: Select a scorekeeper’s name and initiate the removal process. Make sure no other scorekeeper’s names are removed from the active staff list.

Acceptance Criteria: Select the name of a second scorekeeper and initiate the removal process. Make sure the scorekeeper’s name is removed from the active staff list.

## 5.5 Current Event List

Feature Under Test: Automatically generate a list of current events

Use Case: Following a successful login, the program coordinator will be directed to the system’s home page where a list of currently scheduled events will be automatically generated.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he is directed to the home page to view a list of currently scheduled events.
2. The list will begin with the current date and show events up to 7 days in the future.

Expected Outcome Success:

A list of currently scheduled events will be automatically displayed.

* The list will begin with the current date and display events up to 7 days in the future.
* The displayed list will show all events within the expected timeframe.

Expected Outcome Failure:

A list of currently scheduled events will not be automatically displayed.

* An error message is displayed in the application that there are no events to display in the given timeframe.
* The displayed list will not show all events within the expected timeframe.

### 5.5.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to see the current week’s list of events when I arrive on the home page so that I may view the current schedule.

Acceptance Criteria: Log into the system to access the home page. Make sure the event list displays events from the current date up to 7 days in the future.

Acceptance Criteria: Log into the system to access the home page. Make sure the event list displays all of the events scheduled in the current 7-day time period.

## 5.6 Previous Event List

Feature Under Test: Automatically generate a list of the previous week’s events

Use Case: Following a successful login, the program coordinator will be directed to the system’s home page where a list of currently scheduled events will be automatically generated. From there they may elect to view the previous weeks’ worth of scheduled events.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he is directed to the home page to view a list of currently scheduled events.
2. The coordinator elects to view the previous weeks’ worth of scheduled events.
3. The list will end with the current date and display the previous 7 days of scheduled events.

Expected Outcome Success:

A list of the previous weeks’ worth of scheduled events scheduled events will be automatically displayed.

* The list will end with the current date and display the previous 7 days of scheduled events.
* The displayed list will show all events within the expected timeframe.

Expected Outcome Failure:

A list of the previous weeks’ worth of scheduled events scheduled events will not be automatically displayed.

* An error message is displayed in the application that there are no events to display in the given timeframe.
* The displayed list will not show all events within the expected timeframe.

### 5.6.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to see the past week’s list of events so that I may know what events were scheduled.

Acceptance Criteria: Elect to view the prior week’s events from the home page. Make sure the event list displays events from the current date up to 7 days in the past.

Acceptance Criteria: Elect to view the prior week’s events from the home page. Make sure the event list displays all of the events scheduled in the previous 7-day time period.

## 5.7 Next Event List

Feature Under Test: Automatically generate a list of the next week’s events

Use Case: Following a successful login, the program coordinator will be directed to the system’s home page where a list of currently scheduled events will be automatically generated. From there they may elect to view the previous weeks’ worth of scheduled events.

Interactive Reproduction Steps:

1. After the coordinator has successfully entered the system, s/he is directed to the home page to view a list of currently scheduled events.
2. The coordinator elects to view the previous weeks’ worth of scheduled events.
3. The list will end with the current date and display the previous 7 days of scheduled events.

Expected Outcome Success:

A list of the previous weeks’ worth of scheduled events scheduled events will be automatically displayed.

* The list will end with the current date and display the previous 7 days of scheduled events.
* The displayed list will show all events within the expected timeframe.

Expected Outcome Failure:

A list of the previous weeks’ worth of scheduled events scheduled events will not be automatically displayed.

* An error message is displayed in the application that there are no events to display in the given timeframe.
* The displayed list will not show all events within the expected timeframe.

### 5.7.1 User Story and Acceptance Criteria

User Story: As a Program Coordinator I want to see the next week’s list of events so that I may know what events coming up.

Acceptance Criteria: Elect to view the next week’s events from the home page. Make sure the event list displays events starting 7 days in the future for another 7 days beyond that.

Acceptance Criteria: Elect to view the next week’s events from the home page. Make sure the event list displays all of the events scheduled in the given time period.

1. **Pass/Fail Criteria**

A test failure is defined as any output generated by a test that does not agree with the test cases’ expected outcome when executed in the specified environment. A defect is generated for this failure and it is assigned a severity level. Severity is graded on how impactful the defect is to the customer and the software system’s ability to function.

1. **Suspend/Resume Criteria**

Testing will only occur during regular business hours, Monday through Friday, resuming each morning and ending at the end of each work day. Testing may be suspended if any defect discovered with a high enough severity level makes further testing impossible.

1. **Test Deliverables**

The following items are expected to be delivered by the test team:

1. A system test plan
2. A test design specification, detailing all test specific requirements including needed input data.
3. Detailed test cases, including user stories
4. Detailed test procedures
5. All test logs
6. Test incident reports
7. Defect discovery logs
8. Test summary reports detailing test status and defects found
9. Test data summary including test results
10. **Testing Tasks**

Testing tasks were developed to ensure all functional areas of the Scorekeeper Scheduling System application are covered appropriately. The list of tasks includes:

1. Test plan preparation
2. The creation of the test design specification
3. The preparation of the test case specification
4. The creation of test case/user story reproduction steps
5. The creation of the test case expected outcomes
6. Preparing the production environment, including hardware systems
7. Ensuring all testers have access to necessary documentation and testing tools
8. Execution of functional test and recording results
9. Supervising the test team
10. Preparing status meeting material
11. Attending status meetings
12. Distributing the test status and summary reports
13. **Environmental Requirements**

## **10.1 Lab Setup**

The testing environment will mirror the production environment located within the Scorekeeper Scheduling System Microsoft Azure cloud environment.

## **10.2 Publications**

All test documentation including user stories, acceptance tests, test cases and reproduction steps is stored in the repository and will be made available to all members of the test team.

## **10.3 Tools**

No additional tools are identified for the test team. If any are identified as necessary, they will be provided to the team in a timely manner.

1. **Responsibilities**

The responsibilities will be spread according to skill amongst the members of the test team. The developers and product owner will be responsible for the test design, test cases, procedures, and test status.

1. **Resources**

Testing will be accomplished using the existing hardware that has been configured to mirror the Scorekeeper Scheduling System production environment. No new hardware or software must be purchased for this project.

1. **Risks and Contingencies**

As this is the initial development of the Scorekeeper Scheduling System application and the development team is small there is the risk the project may fall behind due to unforeseen events that further limit the available man hours that may be devoted to the project.

1. **Software Configuration Management**

All software will reside onsite to be managed by the Scorekeeper Scheduling System development team.

**Appendix A. Defect Flow**

The test team is aware that defects will be found during the testing process. A defined defect workflow has been established to advise every one of their roles and responsibilities. As defects are discovered during testing, the tester who discovered the problem is responsible for filling out the appropriate report and providing any necessary information required to locate and help with further testing.

**Appendix B. Peer Review Process**

Peer reviews are a required element of the software development process. As such, each feature added to the application must be peer reviewed.

**Appendix C. Internal Testing Strategies**

## **C.1 Test Automation**

### **Scope**

The Scorekeeper Scheduling System developers understand that 100% test automation is both unrealistic and unwise. Thus, the strategy instead is to automate testing for high priority user interactions.

### **Test Management**

Due to the limitations of the production environment, the test automation tools will be likewise limited.

### **Test Lifecycle**

The automated test suites are primarily used for integration testing. Any new tests created during the development of a new feature will be added to the existing suites upon completion of the feature in question.

## **C.2 Continuous Integration Strategy**

### **Scope**

The Continuous Integration strategy was primarily developed to catch build mistakes and defects early in the software development process.

### **Tools**

The team will be using JUnit tests for code coverage and Selenium testing for ongoing continuous integration testing.