American Heart Screening Management System

Test Plan

Carlo Campanini ccampanini2018@my.fit.edu

Chris Newberry cnewberry2018@my.fit.edu

Jack Dewey jdewey2018@my.fit.edu

Noah Wilson wilsonn2018@my.fit.edu

Led By:

Drew Dunkelberger ddunkelberge2018@my.fit.edu

Sponsored By:

Dr. Eraldo Ribeiro eribeiro@fit.edu

Client:

Evan Ernst, CEO - Who We Play For

Klynton Holmes, Tech Advisor - Who We Play For

Table of Contents

1. Introduction

1.1. Overview

This project involves creating a web application that will manage heart screening events across the United States. There is currently a system in place that is localized to areas of Florida, however, the organization behind these events wants to upgrade their system due to limitations in the current design. Our task is to create an event management system that will scale to a national scope, while keeping or decreasing current speeds of operations in the application.

1.2. Purpose

This test plan will outline features to be tested for the event management system and explain how the features will be tested. The goal of these tests will be to verify functionality and eliminate the presence of as many bugs as possible. Additionally, development of the event management system will be done in a test driven development (TDD) format, so the test plan will be followed closely for each feature before implementation can begin.

1.3. Approach

The development team will be using Mocha and Chai for unit testing all of the code. Besides this, we will also be using testing tools provided by AWS. Finally, each item will be tested on multiple platforms including Google Chrome, Mozilla Firefox, and Microsoft Edge.

2. Testing Plan

2.1. Feature 1: Private Event Link Generation

2.1.1. Case 1

Description	Test if a link is generated when a director creates a private event.
Purpose	The feature requires a link to be created when a director creates a private event. This test ensures that the link is created.
Input	Button Click - Submission of event creation form

Expected Output	A link is presented to the director.
Procedure	1. Initial unit testing of code with Mocha and Chai 2. Create a private event with a director account 3. Check that link is presented to the director

2.1.2. Case 2

Description	Test if only the director who creates a private event can see the event's link.
Purpose	Links for private events must only be viewable by directors who make the events. If anyone else can see the links there is a bug.
Input	None
Expected Output	Separate directors will not be able to see the created private event or its link.
Procedure	1. Initial unit testing of code with Mocha and Chai 2. Create a private event with a director account 3. Log into a separate director's account and attempt to view the created private event and its link.

2.1.3. Case 3

Description	Test that non-private events still do not produce a link.
Purpose	This feature is only supposed to affect private events, so it should be assured that public events are not affected by changes made to the system.

Input	Button Click - Submission of event creation form
Expected Output	New event created with no link outputted to director.
Procedure	1. Initial unit testing of code with Mocha and Chai 2. Create a private event with a director account 3. Check that link is presented to the director

2.2. Feature 2: QR Code Generation

2.2.1. Case 1

Description	Test if a QR code is generated when a director clicks the proper button.
Purpose	To ensure the feature functions as desired.
Input	Button Click - QR generation button for an event
Expected Output	A QR is displayed that can be saved.
Procedure	Click button for generation and verify that it is displayed

2.2.2. Case 2

Description	Test that QR pulls up the proper registration form when scanned.
Purpose	The QRs must pull up the proper registration form otherwise the function is useless.
Input	QR Scan
Expected Output	The scanner should be transferred to the

	event's registration page.
Procedure	1. Scan QR code 2. Verify QR code actually scans 3. Verify destination is the correct path

2.3. Feature 3: Display Available Time Slots for a Private Event

2.3.1. Case 1

Description	Test that all available time slots on a private event with no registered participants are presented.
Purpose	This test will ensure the base functionality of this feature.
Input	A newly created private event
Expected Output	Available time slots are displayed automatically
Procedure	1. Create a new private event with X number of time slots 2. Verify that X number of timeslots are displayed 3. Verify that the correct times are available

2.3.2. Case 2

Description	Test that all available time slots on a private event (that have *some* but not *all* time slots registered) are presented.
Purpose	This test will ensure that the feature is able to filter out unavailable time slots.
Input	A newly created event with X participants registered
Expected Output	Only the time slots not registered for are displayed

Procedure	1. Create a new private event 2. Register X amount of participants for different time slots 3. Begin registration of a new participant 4. Verify that registered time slots are not available for a new registration

2.3.3. Case 3

Description	Test that *no* time slots are presented on a private event that has all time slots registered.
Purpose	This is to test for cracks in the filtering process of unavailable slots.
Input	A newly created private event with all time slots filled
Expected Output	No available time slots are displayed
Procedure	1. Create a new private event 2. Register full amount of participants for different time slots 3. Begin registration of a new participant 4. Verify that no time slots are available for a new registration

2.4. Feature 4: Reminder System for Registered Participants

2.4.1. Case 1

Description	Test system to ensure it reminds participants of an event at 5pm the day before the event.
Purpose	This test is to have an initial test on our cron job implementation and ensure that the timing job is performed accurately.
Input	None

Expected Output	Each participant of an event will get an email notification at 5pm the day before an event.
Procedure	1. Create an event with a director account 2. Register for event with a participant account 3. At 5pm the before an event ensure that the participant gets an email reminder for the event

2.4.2. Case 2

Description	Test system to ensure it reminds many participants of an event at 5pm the day before the event.
Purpose	This test is to give a larger scale to the cron job testing.
Input	None
Expected Output	Each participant of an event will get an email notification at 5pm the day before an event.
Procedure	1. Create an event 2. Register for the event with many participant accounts 3. Test if the participants are given a notification at the proper time.

2.5. Feature 5: Close Online Event Registration Day Before Event

2.5.1. Case 1

Description	Test if an event closes its registration at 5pm the day before the event.
Purpose	The system is required to close events at 5pm the day before the event takes place. This test ensures that the events close

	properly and at the correct time.
Input	None
Expected Output	Event registration is unavailable to participants
Procedure	1. Create an event with a director account 2. Attempt to sign up for event with participant account after 5pm 3. Ensure participants can't sign up after cut off time

2.6. Feature 6: Searching for Payments

2.6.1. Case 1

Description	Test if accountant/director accounts can search for customer payments in a given date range.
Purpose	The system is required to allow these users to search for payments in various methods, one being by a date range.
Input	Date range
Expected Output	All payments made from those given dates
Procedure	1. Use accountant account to request a spreadsheet of a payment history 2. Choose to filter by date range 3. Select upper and lower bound of date range 4. Check if the resulting spreadsheet contains information from correct date range

2.6.2. Case 2

Description	Test if accountant/director accounts can search for customer payments on a given

	date.
Purpose	The system is required to allow these users to search for payments in various methods, one being by a given date.
Input	A date
Expected Output	All customer payments on that given date
Procedure	1. Use accountant account to request a spreadsheet of a payment history 2. Choose to filter by specific date 3. Select date 4. Check if the resulting spreadsheet contains information from correct date

2.6.3. Case 3

Description	Test if accountant/director accounts can search for customer payments from a given event.
Purpose	The system is required to allow these users to search for payments in various methods, one being by searching all payments for a given event.
Input	The event object, containing all event details
Expected Output	All payments made from participant's towards that event
Procedure	1. Use accountant account to request a spreadsheet of a payment history 2. Choose to filter by specific event 3. Select the event 4. Check if the resulting spreadsheet contains information from correct date range

2.6.4. Case 4

Description	Test if accountant/director accounts can search for customer payments for a given participant.
Purpose	The system is required to allow these users to search for payments in various methods, one being by searching all payments for a given participant.
Input	The participant's information
Expected Output	The participant's confirmation of payment
Procedure	1. Use accountant account to request a spreadsheet of a payment history 2. Choose to filter by date range 3. Select upper and lower bound of date range 4. Check if the resulting spreadsheet contains information from correct date range

2.7. Feature 7: System Changes Wording Based on Participant Age

2.7.1. Case 1

Description	Test if accounts with an age of 17 or younger get displayed the correct text.
Purpose	The text presented to participants will vary depending on if they are older than 17 or younger than 18.
Input	Use app with an account with age set for <18
Expected Output	Based on the age, the appropriate text for registration is displayed. (i.e. Parent/Guardian signature)
Procedure	1. Using an account that is <18 browse the application 2. Ensure each text shows the correct wording for the account age

2.7.2. Case 2

Description	Test if accounts that are 18 or older get displayed the correct text.
Purpose	The text presented to participants will vary depending on if they are older than 17 or younger than 18.
Input	An account with age set for 18+
Expected Output	Based on the age, the appropriate text for registration is displayed. (i.e. participant's own signature)
Procedure	1. Using an account that is 18+ browse the application 2. Ensure each text shows the correct wording for the account age

2.8. Feature 8: Allow a Director to Delete Their Events

2.8.1. Case 1

Description	Test if an admin account is notified when a director requests to delete one of their events.
Purpose	The event deletion will involve requesting deletion from admins and this test ensures the admins receive the request.
Input	Button click on delete button
Expected Output	Admin account receives a notification that the director requested to delete an event
Procedure	1. Create an event with a director account 2. Attempt to delete the event 3. Login with administrator account and verify that notification of deletion request is present

2.8.2. Case 2

Description	Test that an admin can properly confirm and deny deletion of an event.
Purpose	This test ensures that the confirmation/denial of an event deletion works properly with no side effects.
Input	An event submission
Expected Output	The event successfully denied and not appearing under list of events nor in the database
Procedure	1. Create and request deletion of an event from a director account 2. Use an admin account to confirm the deletion of the event 3. Ensure that all records of the event are deleted

2.9. Feature 9: Require Confirmation for Deletion of Events

2.9.1. Case 1

Description	Test the confirmation for event deletion.
Purpose	To prevent accidental event deletion, a confirmation will be presented to a director on deletion attempt. This test will ensure the functionality of this feature.
Input	Button click on delete button.
Expected Output	A prompt displays asking the director if they are sure that they want to submit a request for deletion.
Procedure	1. Create an event with a director account 2. Attempt to delete event 3. Ensure pop up requesting confirmation for deletion appears

2.10. Feature 10: Allow Participants to Cancel Their Registrations

2.10.1. Case 1

Description	Use a participant account to test canceling a registration.
Purpose	This tests if a participant can properly back out of an event after signing up for it.
Input	Button Click: Participant cancellation of event
Expected Output	The participant's information and registration removed from the event
Procedure	1. Sign up for an event with a participant account 2. Cancel the registration of the account 3. Check for pop up 4. Check that pop up displays correct information

2.10.2. Case 2

Description	Ensure that options for either refunding or crediting for a new event pops up after a participant cancels registration.
Purpose	This ensures that customers will be able to properly get their money back from an event they could not attend.
Input	Button Click: Participant cancellation of event
Expected Output	When the participant cancels registration, a pop up with the options to get a refund or credit money appears.
Procedure	1. Sign up and cancel

2.10.3. Case 3

Description	Test out refund functionality.
Purpose	Since the system works with people's money, the refund system must be tested thoroughly.
Input	Participant cancellation of registration for an event.
Expected Output	A payment from the system's account to the participant's account that requested a refund.
Procedure	1. Create an event with a director account 2. Sign up for the event with a participant account 3. Cancel registration for event 4. Choose refund option when cancelling 5. Check that system properly transferred funds to participant 6. Check that system transferred correct amount of funds

2.10.4. Case 4

Description	Test out crediting functionality.
Purpose	If a customer prefers to credit for another event, the accreditation system must be tested.
Input	Participant cancellation of registration for an event.
Expected Output	System properly stores how much a participant has been credited.
Procedure	1. Create an event with a director account 2. Sign up for the event with a participant account 3. Cancel registration for event

	4. Choose credit option when cancelling 5. Check that system properly stores credit for participant 6. Check that system credits the correct amount of money

2.11. Feature 11: Allow Participant Payments to be Exported to a Local Spreadsheet

2.11.1. Case 1

Description	Test button for converting participant payments to spreadsheet format.
Purpose	This ensures that the button properly triggers spreadsheet creation.
Input	Click on export button
Expected Output	A spreadsheet containing participant payment information
Procedure	1. Log in with accountant account 2. Click button to convert a payment history into a spreadsheet 3. Check that the spreadsheet is created and stored properly

2.11.2. Case 2

Description	Test information conversion from databases to spreadsheets.
Purpose	This test is to ensure that all the information in the participant payment database is properly transferred to spreadsheet format.
Input	Database tables and information from database
Expected Output	A spreadsheet containing information that was in the database

Procedure	1. Log in with accountant account 2. Click button to convert a payment history into a spreadsheet 3. Check if all the cells in the spreadsheet properly match the database information

2.12. Feature 12: Verify a User Before Accessing ECG Information

This feature needs to be further discussed with the client for more information before tests can be written