Test Plan and Cases (TPC)

Farmworkers Safety App

Team 09

TEAM MEMBER NAME	ROLES
Shobhit Agarwal	Project Manager Life Cycle Planner System Architect
Akshay Aggarwal	System Architect Prototype Developer Feasibility Evidence Analyst
Viraj Sahai	Prototype Developer Feasibility Evidence Analyst
Vahagen Sinanian	Operational Concept Developer NDI Analysis Personas
Juan Andrade	Requirements Engineer Prototype Developer Operational Concept Developer
Basir Navab	Life Cycle Planner Project Manager
Marko Djuliarso	Independent Verification and Validation Quality Focal Point

12/05/2016

Version History

Date	Author	Version	Changes made	Rationale
12/05/16	Shobhit Agarwal	1.0	• Compiled the entire doc	• Complete doc for TPC

Table of Contents

Table of Tables

1. Introduction

This document describes the test cases and the acceptance criteria of the Farmworker safety app system. The purpose of doing is to identify the draft acceptance test for specific win-win requirements. The most critical requirements are tested in this document based on the use cases from SSAD document. The testing is focused at software level testing to test key features of the system. The test cases were documented using the equivalence partitioning technique by dividing all potential input data into partitions based on the expected output of our system.

2. Test Strategy and Preparation

Based on the requirements and the use cases, we define a series of acceptance test cases for the key functionalities.

2.1 Hardware preparation

Hardware needed:

- A computer equipped with a web browser and an internet connection
- A smartphone with a browser and an internet connection
- Any mobile phone with texting capabilities
- A cloud based web server to host the application

2.2 Software preparation

Software needed:

- Any modern web browser
- Android / iOS or other mobile operating system
- Server side scripts deployed on web server for sending automated text and retrieving weather information
- PHPUnit, MySQL and NUnitASP testing software

2.3 Other pre-test preparations

None

2.4 Requirements Traceability

Table 1: Requirements Traceability Matrix

OCD	Requirements	Test Case
OC1	As an admin, I should see all of the framework's information.	TC-01 Send SMS notifications
OC1	As a user, I should be able to report that I am feeling ill or dizzy in order to identify signs of heat-illness.	TC-01 Send SMS notifications
OC1	As a user, I should be able to create emergency contact list, and also be able to contact 911 and CalOsha.	TC-01 Send SMS notifications
OC1	As an admin, I should see all of the information of farm.	TC-02 Fetch temperature based on user's location
OC1	As an admin I should be able to access all temperature and health data collected from the farmers	TC-02 Fetch temperature based on user's location
OC1	As a user, I should be able to share my location with emergency services.	TC-07 Update farmworker location via SMS TC-08 Farmworker self-update of location
OC1	As a farm manager/contractor I should be able to search for farmers available to work.	TC-09 Contractor/Farmer update of farmworker location
OC1	As an admin, I should be able to define roles for App Access (App Security)	TC-10 Create user profile

OCD	Requirements	Test Case
OC2	As a user, I should know how to work with application	TC- 11 Edit/Update user profile, TC-12 Delete user profile
OC2	As an admin, I should be able to add, and delete the educational content.	TC-03 Upload new educational content
OC2	As an Admin I should be able to upload videos.	TC-03 Upload new educational content
OC2	As a user, I should be able to access the educational material without internet connection.	TC-04 View educational content
OC2	As a user, I should be able to watch educational videos in the application at any time.	TC-04 View educational content
OC2	As a user, I should be able to watch videos and educational data.	TC-04 View educational content
OC2	As an Admin I should be able to create quizzes.	TC-05 Create a quiz
OC2	As a user, I should be able to do quizzes.	TC-06 Take a quiz on educational content
OC3	As a user, I should be able to use most features with minimal mobile data usage.	TC-04 View educational content TC-06 Take a quiz on educational content

3. Test Identification

Testing Suites:
Capability Requirement CR-1: Temperature-based notification system

1.1 Test Identifier
TC-01 Send SMS notifications that farmworkers as users receive different SMS notifications about weather conditions.

1.2 Test Level
Unit testing

1.3 Test Class
Erroneous input testing

1.4 Test Completion Criteria
Send SMS notifications will be completed when:

- The user receives message successfully.
- The user receives the correct information about weather information.
- The user receives message in the correct format and language.

1.5 Test Cases
Table 1: TC-01-01 The system sends the message successfully.

Description The main goal of this test case is to assess the reliability of our SMS notification system, as this system can prevent our users from harm. It's really important because if the user does not receive the message successfully. They can't understand when the temperature goes high to save their lives.

Test Case Number	TC-01-01 The system sends the message successfully
Test Item	Tested the system sends messages successfully.
Test Priority	M (Must Have)

Pre-conditions	The number of the user should be in the system when he/she registered
Post-conditions	The user should receive the correct message from the system successfully.
Input Specifications	- User go to the registration page to register. - User enter his/her information including the cellphone number. - click on the register bottom.
Expected Output Specifications	The user should receive a correct message from the system.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The system sends the message notification correctly. • The user receives the correct message successfully. Fail criteria: Other than pass criteria.
Assumptions and Constraints	N/A
Dependencies	N/A
Traceability	WC_4156: As a user, I should be able to receive Temperature based notification via text when it exceeds a threshold

Table 2: TC-01-02 The system doesn't send the message.

Description: Again same as the last description, the main goal of this test case is to assess the reliability of our SMS notification system, as this system can prevent our users from harm. It's really important because if the user does not receive the message successfully. They can't understand when the temperature goes high to save their lives.

Test Case Number	TC-01-02 The system doesn't send the message
Test Item	Tested the system does not send messages successfully.
Test Priority	M (Must Have)
Pre-conditions	The number of the user should be in the system when he/she registered
Post-conditions	The user does not receive the message and the system recognized it.
Input Specifications	- User go to the registration page to register. - User enter his/her information including the cellphone number. - Click on the register bottom.
Expected Output Specifications	The user does not receive the message successfully. The system sends an error notification to the admin of the system.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following condition. • The error message is shown for the admin correctly.

	Fail criteria: Other than pass criteria.
Assumptions and Constraints	N/A
Dependencies	N/A
Traceability	WC_4156: As a user, I should be able to receive Temperature based notification via text when it exceeds a threshold

Table 3: TC-01-03 The system sends SMS in incorrect language.

Description: The main goal of this test case is to assess the reliability of our SMS notification system, as this system can prevent our users from harm. It's really important because if the user does not receive the message in the correct format and language he/she cannot understand the important information that we are sending them about the weather condition and the temperate.

Test Case Number	TC-01-03 The system sends SMS in incorrect language
Test Item	Tested the system does not send messages in correct language.
Test Priority	M (Must Have)
Pre-conditions	The number of the user should be in the system when he/she registered
Post-conditions	The user does not receive the message in the correct language.
Input Specifications	- User go to the registration page to register. - User enter his/her information including the cellphone number. - click on the register bottom.
Expected Output Specifications	The user does not receive the message in the correct language. The system sends an error notification to the admin of the system.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following condition. • The error message is shown for the admin correctly. Fail criteria: Other than pass criteria.
Assumptions and Constraints	N/A
Dependencies	N/A
Traceability	WC_4156: As a user, I should be able to receive Temperature based notification via text when it exceeds a threshold

Table 4: TC-01-4 The system sends SMS in correct language.

Description: again same as last description. The main goal of this test case is to assess the reliability of our SMS notification system, as this system can prevent our users from harm. It's really important because if the user receive the message in the correct format

and language he/she can understand the important information that we are sending them about the weather condition and the temperate.

Test Case Number	TC-01-4 The system sends SMS in correct language.
Test Item	Tested the system sends messages in correct language.
Test Priority	M (Must Have)
Pre-conditions	The number of the user should be in the system when he/she registered
Post-conditions	The user receives the message in the correct language.
Input Specifications	- User go to the registration page to register. - User enter his/her information including the cellphone number. - click on the register bottom.
Expected Output Specifications	The user receives the message in the correct language.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The system should send the message in the correct language. • The user receives a message about the weather information and temperate in the correct language. Fail criteria: Other than pass criteria.
Assumptions and Constraints	N/A
Dependencies	N/A
Traceability	WC_4156: As a user, I should be able to receive Temperature based notification via text when it exceeds a threshold

Table 5: TC-01-5 The system sends the message to a correct phone number.
Description: The main goal of this test case is to assess the reliability of our SMS notification system, as this system can prevent our users from harm. It's really important because if the system does not send text to the right number, users cannot receive the information about the weather and temperature.

Test Case Number	TC-01-5 the system sends the message to a correct phone number.
Test Item	Tested the system sends messages to the correct number.
Test Priority	M (Must Have)
Pre-conditions	The number of the user should be in the system when he/she registered.
Post-conditions	The system should to send the message to the correct number.
Input Specifications	- User go to the registration page to register. - User enter his/her information including the cellphone number. - click on the register bottom.

Expected Output Specifications	The system sends notification text to the right number.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following condition. • The system sends the text to the right number. • The user receives a message about the weather information and temperate. Fail criteria: Other than pass criteria.
Assumptions and Constraints	N/A
Dependencies	N/A
Traceability	WC_4156: As a user, I should be able to receive Temperature based notification via text when it exceeds a threshold

2.1 Test Identifier

TC-02 Fetch Temperature based on user's location.

2.2 Test Level

Unit Testing.

2.3 Test Class

- Correct Input Testing.
- Erroneous Input Testing.

2.4 Test Completion Criteria

The test for the Fetch Temperature based on user's Location functionality, will be completed when:

- The system fetches the temperature information successfully from the weather API.
- The system shows an appropriate error message, when a problem is encountered while fetching the temperature information from the weather API.
- The system fetches the weather forecast information successfully from the weather API.
- The system shows an appropriate error message, when a problem is encountered while fetching the weather forecast information from the weather API.

- The system correctly determines if the temperature value goes above the specified threshold or not.

2.5 Test Cases

Table 1: TC-02-01 Request Weather Information from Weather API based on user's location.

Description: This test case, is a logical test to perform, in accordance to make a complete and appropriate testing of the Test Suite: TC-02 Fetch Temperature based on user's location. It involves a sunny day scenario in regard of weather information request from the API, which is of truly importance and it corresponds to one of the core functionalities of the Farmworker's Safety Application, therefore it is of high priority that it works as it should.

Test Case Number	TC-02-01 Request Weather Information from Weather API based on user's location.
Test Item	Request Weather Information Functionality, providing a location.
Test Priority	M (Must Have)
Pre-conditions	Fetching of User's Location.
Post-conditions	The weather information is requested successfully and accurately from the Weather API.
Input Specifications	1. Correct and appropriate, user's location is fetched from the data base. 2. Execute the function of weather information retrieval.
Expected Output Specifications	The system shows successfully accurate weather information of the user's location.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions: • The weather information is shown on the screen correctly and without any errors. • The weather information of the location is stored on the database. Fail criteria: • Any other criteria that the one described on the pass criteria.
Assumptions and Constraints	• The location of the user selected for the test is correct and accurate, without any errors or invalid characters.

	• Weather API, must be online and working properly. • The database must be online and available for the location fetching.
Dependencies	N/A
Traceability	As a user, I should be able to view current temperature data so that I know if conditions are unsafe

Table 2: TC-02-02 Request Weather Information from Weather API using invalid or missing location value.

Description: This test case, is a logical test to perform, in accordance to make a complete and appropriate testing of the Test Suite: TC-02 Fetch Temperature based on user's location. It involves a rainy-day scenario in regard of weather information request from the API, which is of truly importance and it corresponds to one of the core functionalities of the Farmworker's Safety Application, therefore it is of high priority that it works as it should.

Also, more specifically to this test, if the location of the user is missing or is corrupted/ invalid, the user should be warned and the system should re-fetch the user's location at a later stage as soon as possible. Therefore, the realization of this tests reassures the correct execution of this important segment of the application.

Test Case Number	TC-02-02 Request Weather Information from Weather API using corrupted or invalid location values.
Test Item	Request Weather Information Functionality, providing a location in an invalid format or blank.
Test Priority	M (Must Have)
Pre-conditions	Fetching of User's Location.
Post-conditions	The system identifies that the user's location value is invalid and/or is not yet available.
Input Specifications	1. User location fetched from the database is in invalid format and/or empty. 2. Execute the function of weather information retrieval.
Expected Output Specifications	The system shows an error message indicating that the location from which the weather is trying to be fetch is not valid or blank.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The error message is shown on the screen accordingly.

	• No request to the weather API is made. Fail criteria: • Any other criteria that the one described on the pass criteria.
Assumptions and Constraints	• The location entered must invalid or blank. • The database must be online and available for the location fetching.
Dependencies	N/A
Traceability	As a user, I should be able to view current temperature data so that I know if conditions are unsafe

Table 3: TC-02-03 Request daily weather forecast from Weather API at a previously specified schedule.

Description: This test case, is a logical test to perform, in accordance to make a complete and appropriate testing of the Test Suite: TC-02 Fetch Temperature based on user's location. Because it involves a sunny-day scenario in regard of weather forecast information request from the API, which is of truly importance and it corresponds to one of the core functionalities of the Farmworker's Safety Application, therefore it is of high priority that it works as it should.

Also, more specifically to this test, since the plan that is going to be conveyed with the company that provides the Weather API has a cap on API requests, it is very important that the Farmworker Safety App application makes as fewer requests as possible, therefore a schedule is set for these API calls, that will both satisfy the end user and the client.

Additionally, due to the difference on time zones in the country, this test will also demonstrate that this functionality will perform accordingly in different time and schedules constraints.

Test Case Number	TC-02-03 Request daily weather forecast from Weather API at a previously specified schedule.
Test Item	Request Weather Forecast Functionality, providing a location.
Test Priority	M (Must Have)
Pre-conditions	• Fetching of User's Location. • The System is set to time and date that matches the specified schedule.

Post-conditions	The weather forecast information is requested successfully and accurately and at the correct schedule from the Weather API.
Input Specifications	1. Correct and appropriate, user's location is fetched from the data base. 2. Execute the function of weather forecast information retrieval.
Expected Output Specifications	The system shows successfully accurate weather forecast information of the user's location.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The weather forecast information is shown on the screen correctly and without any errors. • The weather forecast information of the location is stored on the database. Fail criteria: • Any other criteria that the one described on the pass criteria.
Assumptions and Constraints	• The location of the user selected for the test is correct and accurate, without any errors or invalid characters. • Weather API, must be online and working properly. • The database must be online and available for the location fetching.
Dependencies	N/A
Traceability	As a user, I should be able to view current temperature data so that I know if conditions are unsafe

Table 4: TC-02-04 Request daily weather forecast at a schedule different that the one previously specified.

Description: This test case, is a logical test to perform, in accordance to make a complete and appropriate testing of the Test Suite: TC-02 Fetch Temperature based on user's location. Because it involves a rainy-day scenario in regard of weather forecast information request from the API, which is of truly importance and it corresponds to one of the core functionalities of the Farmworker's Safety Application, therefore it is of high priority that it works as it should.

Also, more specifically to this test, since the plan that is going to be conveyed with the company that provides the Weather API, has a cap on API calls, it is very important that the Farmworker Safety App application makes as fewer requests as

possible, therefore a schedule is set for these API calls, that will both satisfy the user and the client.

Additionally, due to the difference on time zones in the country, this test will also demonstrate that this functionality will perform accordingly in different time and schedules constraints, as it constitutes a boundary case scenario.

Test Case Number	TC-02-04 Request daily weather forecast at a time of day different that the one previously specified.
Test Item	Attempt to request weather forecast functionality, providing a location, at a different time schedule that the one previously specified.
Test Priority	M (Must Have)
Pre-conditions	• Fetching of User's Location. • The System is set to time and date that doesn't match the specified schedule.
Post-conditions	• The weather forecast information is not requested. • The system must provide an appropriate warning accordingly.
Input Specifications	1. Correct and appropriate, user's location is fetched from the data base. 2. Execute the function of weather forecast information retrieval.
Expected Output Specifications	The system does not make a forecast request from the weather API, and shows a signal indicating this outcome.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The error message is shown on the screen accordingly. • No request to the weather API is made. Fail criteria: • Any other criteria that the one described on the pass criteria.
Assumptions and Constraints	• The location of the user selected for the test is correct and accurate, without any errors or invalid characters. • The database must be online and available for the location fetching.
Dependencies	N/A

Traceability	As a user, I should be able to view current temperature data so that I know if conditions are unsafe

Table 5: TC-02-05 Determine if the user's location weather conditions (temperature) go above the specified threshold.

Description: This test case, is a logical test to perform, in accordance to make a complete and appropriate testing of the Test Suite: TC-02 Fetch Temperature based on user's location. Because it involves a standard scenario in regard of retrieving the temperature information from the API and determining if its value is (or is not) within the predefined threshold, as it corresponds to one of the core functionalities of the Farmworker's Safety Application, it is of high priority that it works properly.

If the temperature is above the threshold the system should provide a warning (that would later on be utilized in other segments of the system to notify the end user of this temperature conditions) if it is below the threshold no warning is shown.

Test Case Number	TC-02-05 Determine if the user's location weather conditions (temperature) go above the specified threshold.
Test Item	Determine if the temperature information of the user's location is above the predefined temperature threshold.
Test Priority	M (Must Have)
Pre-conditions	• Fetching of User's Location. • Fetching of User's Location Weather Information (Temperature).
Post-conditions	• If the location's temperature is above the threshold, a warning is shown. • If the location's temperature is below the threshold, no warning is shown.
Input Specifications	1. Correct and appropriate, user's location is fetched from the data base. 2. Correct and appropriate temperature value is fetched from the location's weather information. 2. Execute the function of temperature raise warning.
Expected Output Specifications	The system determines correctly and accurately if the temperature value of a location is within the predefined temperature threshold or not.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The warning message is shown on the screen at the appropriate scenario (Temperature above threshold). • No warning message is shown at the appropriate

	scenario (Temperature below threshold). Fail criteria: • Any other criteria that the one described on the pass criteria.
Assumptions and Constraints	• The location of the user selected for the test is correct and accurate, without any errors or invalid characters. • Correct Temperature value is always provided. • The database must be online and available for the location fetching.
Dependencies	N/A
Traceability	As a user, I should be able to view current temperature data so that I know if conditions are unsafe

Capability Requirement CR-2: Educational content system

1. Test Identifier

TC-03 Upload new educational content
Farmers Safety app contains an educational section which contains education videos and documents for farmworkers. These videos will train and educate the farmers about how to prevent and reduce the risk of heat illnesses.
In this test suit, I will test the functions that allow the admin of the system to upload the videos and documents.

2. Test Level

Testing the TC-03: Upload new educational material feature of farmworkers safety app is in ***unit testing level*** because a small unit of the entire educational feature of our system is being tested.

3. Test Class

Erroneous input testing

4. Test Completion Criteria

The test for uploading the new educational material will be completed when:

- Display easy to use and well organized file selection window for admin
- Dynamically test the size and format of the file and display warning to the admin
- The admin of the system be able to upload education videos and documents.
- Display appropriate success or failure message after upload for admin

Test Cases

Table 1: TC-03-01 Upload a file with correct size and format.

Description: I used equivalence partitioning technique to design this test case. This test case is important because we always have to test the base condition to make sure that everything works in base case before moving to other cases.

Test Case Number	TC-03-01 Upload a file with correct size and format.
Test Item	The file being uploaded by the admin matches all the requirements of the system.
Test Priority	M (Must have)
Pre-conditions	The admin chooses the file that she wants to upload from the file selection window. Then uploads the file to the server
Post-conditions	The admin receives success message.
Input Specifications	a. Click on choose file button b. Select the target file c. Click on open button of file selection window d. Click on upload button
Expected Output Specifications	Receive success message
Pass/Fail Criteria	This test will pass if: a) The expected output occurs b) The file gets stored on server. c) The video or document become available to the user to watch and review. Otherwise it fails.

Assumptions and Constraints	a) The file is selected before clicking upload button.
Dependencies	none
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 2: TC-03-02 Upload a file with correct size but incompatible format.

Description: I used equivalence partitioning technique to design this test case. This test case is important because the file that is being uploaded should be compatible with the system so the farmworkers, farmers, and contractors be able to view them.

Test Case Number	TC-03-02
Test Item	The admin tries to upload a file which have a different format than what is requiered by the system.
Test Priority	M(Must have)
Pre-conditions	The admin chooses the file that she wants to upload from the file selection window and clicks on open.
Post-conditions	A warning message shows up to inform the admin that the file has incompatible format. Then the system allows the admin to choose other file.
Input Specifications	a. Click on choose file button b. Select the target file c. Click on open button of file selection window
Expected Output Specifications	Appropriate error message that states the selected file has incompatible format.
Pass/Fail Criteria	This test will pass if: a) The Admin select the file and receive an error message. b) After closing the error message and selection window the

	file will not be selected for upload. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	Depended on the file format requirements defined for the system.
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 3: TC-03-03 upload a file with correct format but larger size

Description: I used equivalence partitioning technique to design this test case. This test case is important for storage management.

Test Case Number	TC-03-03
Test Item	The admin tries to upload a file which size exceeds the limits defined for system to upload a file.
Test Priority	M(Must have)
Pre-conditions	The admin chooses the file that she wants to upload from the file selection window and clicks on open.
Post-conditions	A warning message shows up to inform the admin that the file has larger size and cannot be uploaded. Then the system allows the admin to choose another file.
Input Specifications	a) Click on choose file button b) Select the target file c) Click on open button of file selection window
Expected Output Specifications	Appropriate error message that states the selected file exceeds the size limits defined by system.
	This test will pass if:

Pass/Fail Criteria	c) The Admin select the file and receive an error message. d) After closing the error message and selection window the file will not be selected for upload. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	The file size limits defined for the system.
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 4: TC-03-04 Not enough space on server for the new file

Description: I used equivalence partitioning technique to design this test case. This test case is important to make sure that the admin is being informed about the upload error and she knows how to fix the problem. Also, it will prevent from damaging the existing files by overwriting them.

Test Case Number	TC-03-04
Test Item	There are a lot of educational material existing on server storage that there is no room for new file to be uploaded.
Test Priority	M(Must have)
Pre-conditions	The admin selects the file that she wants to upload, clicks on open to finalize the selection, then clicks upload to upload the file.
Post-conditions	The admin receives error message indicating there is not enough storage space on server to upload the file.
Input Specifications	a. Click on choose file button b. Select the target file

	c. Click on open button of file selection window d. Click on upload button to upload the file.
Expected Output Specifications	Error message that inform the user about storage space of the server.
Pass/Fail Criteria	This test will pass if: a) The user receives not enough space error message. b) The system allows the admin to perform the action again without refreshing the app. Otherwise it will fail.
Assumptions and Constraints	The system doesn't overwrite space and damage other files to complete the upload.
Dependencies	none
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 5: TC-03-05 Upload a file that already exist in directory

Description: I used equivalence partitioning technique to design this test case. This test case is important because existence of two files with same name and format can cause major problems.

Test Case Number	TC-03-05
Test Item	The admin uploads a file that has same name and format as another file that already exists on server.
Test Priority	M(Must have)
Pre-conditions	The admin selects the file that she wants to upload from file selection window and click on open to finalize the selection. The clicks on upload to upload the file.

Post-conditions	receives an error message that another file with same name and format already exist. The error message has two condition: replace and cancel. If the user choose replace the existing file will be overwritten by new file and cancel will cancel the upload.
Input Specifications	a. Click on choose file button b. Select the target file c. Click on open button of file selection window d. Click on upload button to upload the file.
Expected Output Specifications	Error message that indicates a file with same name and format already exists.
Pass/Fail Criteria	This test will pass if: a) It displays the error message that indicates file with same name and format exists. b) Click on replace will replace the existing file with new file. c) If the user decides to cancel the upload, the existing file will not be overwritten. d) Overwriting the file will not damage other files. e) Two file with same name will not exist. Otherwise it will fail.
Assumptions and Constraints	Replacing the file will not damage other files and it will not keep two files with same name and format on server.
Dependencies	none
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

1. **Testing Identifier**
   TC-04 View Educational content

The test suite is to test the functionality of the farmworker being able to successfully view all the educational content when one is available. The system should also show any new content as soon as it becomes available.

2. Test Level

Unit Testing: Since the test suite caters to test a single functionality which is independent of any sequence of steps required under the assumption that the underlying functionalities which are directly used by the test suite are all working as expected.

3. Test Class

Erroneous input test. The test suite validates the functionality using equivalence partitioning by trying different sets of input to verify that the expected behavior is performed. On entering any incorrect information, the system does not crash and gracefully shows the error message

4. Test Completion Criteria

The test for viewing educational content involves the following conditions to be satisfied:

- The farmworker is able to view the same educational content multiple times
- Removal of any content while being use does not crashes the application, but gracefully shows the error.
- Links for the removed educational content does not show.
- The farmworker is able to search for the educational content he wants to watch/read.

5. Test Cases

Table 2: TC-04-01 The farmworker is able to view the same educational content multiple times

Description: I am testing the equivalence partitioning on when a farmworker tries to view the same educational content multiple times.
Importance: A user might have to stop reading / viewing the content in between and may want to return to it later. A user could also want to refer the content later even when it's completed.

Test Case Number	TC-04-01 The farmworker is able to view the same educational content multiple times
Test Item	Ability to view the educational content multiple times.
Test Priority	M (Must Have)

Pre-conditions	The system displays all the available educational contents.
Post-conditions	The system successfully shows the educational content.
Input Specifications	1. Click Educational Content 2. Click on any available educational content that has been watched at least once earlier
Expected Output Specifications	The system displays the educational content
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The educational content is displayed from the start. • No error is thrown that it has been already watched. • The correct education content that was clicked is shown. Fail criteria: • The system fails to show the educational content • A wrong educational content is shown. • There is any error shown
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • The user have permission to view the educational content.
Dependencies	N/A
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 3: TC-04-02 Removal of any content while being use does not crashes the application, but gracefully shows the error.

Description: I am testing the equivalence partitioning when a content is deleted while some user was using it
Importance: Deletion of a content should not crash the app on a system that was using it

Test Case Number	TC-04-02 Removal of any content while being use does not crashes the application, but gracefully shows the error.
Test Item	Ability to handle dynamic removal of a content
Test Priority	M (Must Have)
Pre-conditions	The system displays all the available educational contents.
Post-conditions	The system shows an error message that the content

	requested was deleted
Input Specifications	1. Click Educational Content 2. Click on any available educational content 3. Admin deletes the content that the user is watching
Expected Output Specifications	The system shows an error saying that the content has been removed
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The educational content is displayed from the start. • No error is thrown until the content is deleted. • An error is thrown when content is deleted and user is taken to the list of all available contents Fail criteria: • The system crashes • A wrong educational content is shown. • There is any error shown
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • The user have permission to view the educational content.
Dependencies	N/A
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 4: TC-04-03 Links for the removed educational content does not show.

Description: I am testing the equivalence partitioning on when an educational content is deleted, the link is also removed.
Importance: Educational content might be deleted while an app is in use

Test Case Number	TC-04-03 Links for the removed educational content does not show.
Test Item	Successfully handle when an educational content is removed while the app is in use
Test Priority	M (Must Have)
Pre-conditions	The system displays all the available educational contents.
Post-conditions	The system shows error when user clicks on removed content and removes that item from the list.
Input Specifications	1. Click Educational Content

	2. Click on any available educational content that has been removed
Expected Output Specifications	The system displays an error message and removes that item from the list
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The educational content is not displayed. • Error is thrown that it has been removed. • The item gets removed from the list. Fail criteria: • The system crashes • A wrong educational content is shown. • There is no error shown
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • The user have permission to view the educational contents
Dependencies	N/A
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

Table 5: TC-04-04 The farmworker is able to search for the educational content he wants to watch/read.

Description: I am testing the equivalence partitioning on when a farmworker tries to search for an education content he/she might be interested in
Importance: With a huge list of educational contents, the user might want to search for a particular content he/she wants to watch

Test Case Number	TC-04-04 The farmworker is able to search for the educational content he wants to watch/read.
Test Item	Ability to perform a search on the educational content.
Test Priority	M (Must Have)
Pre-conditions	The system displays all the available educational contents.
Post-conditions	The system successfully shows the list of educational content for the search query.
Input Specifications	1. Click Educational Content 2. Search for a keyword
Expected Output	The system displays the list of educational contents relevant

Specifications	to the search criteria
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The search result is valid for the keyword. • System does not crashes when there are no search results • User is able to watch the educational content from the search result Fail criteria: • The system fails to show the educational content based on search • A wrong educational content is shown. • System crashes
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to edit the location field.
Dependencies	N/A
Traceability	WC_4159: As a user, I should be able to watch educational videos in the application at any time.

3.1 TC-05 Create a quiz

3.1.1 Test Level

Software item level

3.1.2 Test Class

Erroneous test

3.1.3 Test Completion Criteria

The test for create a quiz will be completed when
- The system create complete and correct quiz.
- The quiz display correctly on the monitor.
- The system show appropriate error message according to the mistake made.

3.1.4 Test Cases

Table 1: TC-05-01 Check create quiz with correct information.

Test Case Number	TC-05-01 Check create quiz with correct information.
Test Item	Create quiz functionality is tested using appropriate information.
Test Priority	M (Must Have)
Pre-conditions	The system display forum page. User has admin privileges
Post-conditions	The created quiz page display correctly.
Input Specifications	1. Click create quiz 2. Put appropriate information in every required fields. 3. Click submit
Expected Output Specifications	The created quiz page is shown.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The quiz is created correctly • Created quiz data is stored in database correctly Fail criteria: • Other than pass criteria.
Assumptions and Constraints	• There are no violated information when the quiz is created • Every required fields is filled. • Only admin has permission to create a quiz.
Dependencies	N/A
Traceability	As an admin, I can create quizzes

Table 6: TC-05-02 Check create quiz with some missing required fields

Test Case Number	TC-05-02 Check create quiz with some missing required fields.
Test Item	Create quiz functionality is tested without some important information.
Test Priority	M (Must Have)
Pre-conditions	The system display the quiz page where user want to create

	a quiz. User has admin privileges
Post-conditions	The system identify the missing fields in the creating-quiz form
Input Specifications	1. Click create quiz 2. Put information in the form with some blank required fields. 3. Click submit
Expected Output Specifications	• The system show red text to identify the missing required fields. • No change for the fields that have already been filled.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The error message is shown correctly. • No change for the other fields. Fail criteria: • Other than pass criteria.
Assumptions and Constraints	• Admin privilege needed to have permission to create a quiz • Hardware and network is working correctly
Dependencies	N/A
Traceability	As an admin, I can create quizzes

Table 3: TC-05-03 Check create post with restricted word.

Test Case Number	TC-05-03 Check create quiz with restricted word.
Test Item	Create quiz functionality is tested with restricted information.
Test Priority	M (Must Have)
Pre-conditions	The system display the quiz page where user want to create a quiz.
Post-conditions	The created quiz display on the monitor without restricted information.
Input Specifications	1. Click create quiz 2. Put information with restricted word in the form 3. Click submit
Expected Output Specifications	The created quiz is shown on the thread page without restricted information.

Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The quiz is created correctly without restricted information • Created quiz data without restricted information is stored in database correctly. • A notification to owner of the quiz is created correctly. Fail criteria: • Other than pass criteria.
Assumptions and Constraints	• The user have permission to create a quiz on the system. • Every required filed is filled.
Dependencies	N/A
Traceability	As an admin, I can create quizzes

Table 4: TC-05-04 Check create quiz by a user without permission.

Test Case Number	TC-05-04 Check delete thread by a user without permission.
Test Item	Create quiz functionality when handle with a user who has no permission.
Test Priority	M (Must Have)
Pre-conditions	The system display forum page where the quiz he wants to create is shown.
Post-conditions	The system display an error message identified that user have no permission to create the quiz.
Input Specifications	Click create quiz
Expected Output Specifications	An error message “Cannot create quiz” is shown.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following condition. • An error message is shown correctly. • User cannot create the quiz. Fail criteria: • Other than pass criteria.
Assumptions and Constraints	The user has no permission to create the quiz

Dependencies	N/A
Traceability	As an admin, I can create quizzes

Table 5: TC-05-05 Check create quiz by a user with permission.

Test Case Number	TC-05-05 Check create quiz by a user with permission.
Test Item	Create quiz functionality is tested by user with permission.
Test Priority	M (Must Have)
Pre-conditions	The system display a page where admin can start to create a quiz
Post-conditions	The system display a page without the created quiz.
Input Specifications	1. Click create quiz 2. Continue to next questions
Expected Output Specifications	The page is shown with created quiz.
Pass/Fail Criteria	Pass criteria: This test will passed when meet all the following conditions • The quiz is created correctly. Fail criteria: • Other than pass criteria.
Assumptions and Constraints	The user has permission to create a quiz
Dependencies	N/A
Traceability	As an admin, I can create quizzes

Capability Requirement CR-2: Educational content system

6. **Testing Identifier**
   TC-06 Take a quiz on Educational content
   The test suite is to test the functionality of the farmworker being able to take a quiz on any educational content if one is available.

7. **Test Level**

   Unit Testing: Since the test suite caters to test a single functionality which is independent of any sequence of steps required under the assumption that the underlying functionalities which are directly used by the test suite are all working as expected.

8. Test Class

Erroneous input test. The test suite validates the functionality using equivalence partitioning by trying different sets of input to verify that the expected behavior is performed. On entering any incorrect information, the system does not crash and gracefully shows the error message

9. Test Completion Criteria

The test for taking educational content quizzes involves the following conditions to be satisfied:

- The farmworker is able to take the quiz multiple times.
- System saves the progress and completion of all quizzes and show the last grade scored by the farmworker
- The farmworker should be able to take a quiz from a mobile device or/and a personal computer
- Deletion of a quiz while in use does not crashes the application, but gracefully exits with an error message

10. Test Cases

Table 7: TC-06-01 The farmworker is able to take the quiz multiple times.

Description: I am testing the equivalence partitioning on when a farmworker tries to take a quiz for more than one time, he should be able to do so.
Importance: A user might want to take a quiz multiple times to ensure he understands the topic well.

Test Case Number	TC-06-01 The farmworker is able to take the quiz multiple times.
Test Item	Ability to take a quiz multiple times
Test Priority	M (Must Have)
Pre-conditions	The system displays the quiz and previous score
Post-conditions	The system saves the user's response and allows to take the quiz more than once.
Input Specifications	3. Click Take a Quiz 4. Take the quiz even if he has taken it earlier 5. Click Submit
Expected Output Specifications	The system saves the new response and shows it on the display
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions.

	• The response gets saved in the database. • User can take the quiz again Fail criteria: • System doesn't save the user's response or score • System does not allow the user to take the quiz more than once
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to take quizzes.
Dependencies	N/A
Traceability	WC_4161: As a user, I should be able to take quizzes

Table 8: TC-06-02 System saves the progress and completion of all quizzes and show the last grade scored by the farmworker

Description: I am testing the equivalence partitioning on when a farmworker takes quiz multiple times and is able to see his last score
Importance: A user might want to take quiz multiple times and want to see what his previous score was.

Test Case Number	TC-06-02 System saves the progress and completion of all quizzes and show the last grade scored by the farmworker
Test Item	Save all quiz responses by the user and show when user takes the quiz again
Test Priority	M (Must Have)
Pre-conditions	The system displays the quiz and previous score
Post-conditions	The system saves the user's response and allows to take the quiz more than once.
Input Specifications	1. Click Take a Quiz 2. User gets the previous score shown on the display 3. Take the quiz and click submit
Expected Output Specifications	The system saves the new response and shows it on the display
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The response gets saved in the database. • User can take the quiz again

	• User can see his last score on the quiz Fail criteria: • System doesn't save the user's response or score • System does not allow the user to take the quiz more than once • System does not show the last score
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to take quizzes.
Dependencies	N/A
Traceability	WC_4161: As a user, I should be able to take quizzes

Table 9: TC-06-03 The farmworker should be able to take a quiz from a mobile device or/and a personal computer

Description: I am testing the equivalence partitioning on when a farmworker tries to take a quiz from either computer or mobile device.
Importance: A user might want to take the quiz from the mobile device by opening the web page on his handheld device.

Test Case Number	TC-06-03 The farmworker should be able to take a quiz from a mobile device or/and a personal computer
Test Item	Ability to take a quiz from a mobile device/computer
Test Priority	M (Must Have)
Pre-conditions	The system displays the quiz and previous score
Post-conditions	The system saves the user's response and allows to take the quiz more than once.
Input Specifications	1. Click Take a Quiz 2. Take the quiz even if he has taken it earlier 3. Click Submit
Expected Output Specifications	The system saves the new response and shows it on the display
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The response gets saved in the database. • User can take the quiz again • The user is successfully able to take quiz either on mobile or computer Fail criteria:

	• System doesn't save the user's response or score • System does not allow the user to take the quiz more than once • The user cannot access quiz on mobile device / computer
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to take quizzes.
Dependencies	N/A
Traceability	WC_4161: As a user, I should be able to take quizzes

Table 10: TC-06-04 Deletion of a quiz while in use does not crashes the application, but gracefully exits with an error message

Description: I am testing the equivalence partitioning on when a farmworker tries to take a quiz from either computer or mobile device.
Importance: A user might want to take the quiz from the mobile device by opening the web page on his handheld device.

Test Case Number	TC-06-04 Deletion of a quiz while in use does not crashes the application, but gracefully exits with an error message
Test Item	Deleting a quiz while in use does not crashes the app
Test Priority	M (Must Have)
Pre-conditions	The system displays the quiz and previous score
Post-conditions	The admin removes the quiz while the user is taking it
Input Specifications	1. Click Take a Quiz 2. Take the quiz even if he has taken it earlier 3. Ask admin to remove the quiz
Expected Output Specifications	An error message is shown and the user is taken to the homepage without crashing the app
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The system shows an error message that the quiz was removed • The system does not crash and takes user to the homepage Fail criteria: • The partial response gets saved in the database. • System does not throw any error message • The system crashes on deletion of the quiz

Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to take quizzes.
Dependencies	N/A
Traceability	WC_4161: As a user, I should be able to take quizzes

Capability Requirement CR-3: Accurate farmworker location

11. Testing Identifier

TC-08 Farmworker self-update of location
The test suite is to test the functionality of the farmworker being able to update his/her location themselves by entering either their ZIP code or the location name on their profile. The system should take in the new location and send the confirmation text along with the current temperature of that location

12. Test Level

Unit Testing: Since the test suite caters to test a single functionality which is independent of any sequence of steps required under the assumption that the underlying functionalities which are directly used by the test suite are all working as expected.

13. Test Class

Erroneous input test. The test suite validates the functionality using equivalence partitioning by trying different sets of input to verify that the expected behavior is performed. On entering any incorrect information, the system does not crash and gracefully shows the error message

14. Test Completion Criteria

The test for updating the farmworker location by the farmworker himself/herself from the profile (self-update of the farmworker location using profile). This should involve following steps

- The farmworker is able to update his/her location correctly using a valid ZIP code which gets persisted in the database.
- Invalid ZIP codes should throw error and not persist in the database.
- The farmworker is able to update his/her location correctly using a valid farm name / city name / area name which gets persisted in the database.
- Farmworker should receive a text notification when the location is successfully updated. Farmworker should receive the current weather information for the new location along with the confirmation message.
- The system should throw an error if the location field is left blank.

15. Test Cases

Table 11: TC-08-01 Update Farmworker location with a valid ZIP code

Description: I am testing the equivalence partitioning on when a farmworker tries to update his location using a valid ZIP code, he/she should be able to successfully update the location which gets persisted in the database.
Importance: A user might want to self-update his location and this is useful to test that condition. The test case would be used to validate the expected behavior when a correct ZIP code is entered.

Test Case Number	TC-08-01 Update Farmworker location with a valid ZIP code.
Test Item	Update of Farmworker profile and his location update is tested using correct information.
Test Priority	M (Must Have)
Pre-conditions	The system displays the profile page with the current location and the ability to edit the location.
Post-conditions	The location in the profile is correctly updated.
Input Specifications	6. Click Edit Profile 7. Change the current location with a valid ZIP code 8. Click Save Changes
Expected Output Specifications	The updated profile page is shown with the new location.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The new location gets saved in the database correctly. • The old location is removed from the farmworker's profile. • The new location name and the ZIP code is then displayed on the updated profile page correctly. Fail criteria: • The new location does not gets saved in the database. • The new location does not gets updated in the profile page • There is any error on entering a valid ZIP • The system does not recognize/updates a valid ZIP
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled.

	• The user have permission to edit the location field.
Dependencies	N/A
Traceability	WC_4184: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 12: TC-08-02 Update Farmworker location with an invalid ZIP code / farm name / city name

Description: I am testing the equivalence partitioning on when a farmworker tries to update his location using an invalid ZIP code / area name / farm name / city name, he/she should not be able to update the location.

Importance: This is important to catch any invalid entries being persisted in the database as they later would cause problems while trying to fetch weather for the location.

Test Case Number	TC-08-02 Update Farmworker location with an invalid ZIP code / city name / farm name / farm code
Test Item	Update of Farmworker profile and his location update is tested using incorrect information.
Test Priority	M (Must Have)
Pre-conditions	The system displays the profile page with the current location and the ability to edit the location.
Post-conditions	The location in the profile is not updated and remains what is was before the change was requested.
Input Specifications	1. Click Edit Profile 2. Change the current location with an invalid ZIP code / location code / area name / farm code or name 3. Click Save Changes
Expected Output Specifications	• An error message is displayed that the entered location is incorrect, please try again • the location on the profile is not updated
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The new location does not gets saved in the database. • The old location is not touched and the field does

	not get updated. • The old location is displayed on the updated profile page. • An error message is shown that tells the user that the location entered is incorrect Fail criteria: • The incorrect location gets saved in the database • An error message is not shown • The old location is removed from the database / profile • The profile shows the new incorrect location
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to edit the location field.
Dependencies	N/A
Traceability	WC_4184: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 13: TC-08-03 Check if the user receives a text message with a successful message on his location being updated and the weather information. Farmworker should receive the current weather information for the new location along with the confirmation message.

Description: I am testing the equivalence partitioning on when a farmworker successfully updates his/her location, he should receive a confirmation text message that update was successful. The user should also receive a text message with the current weather information of the new location.

Importance: This would test the functionality that a confirmation message is sent assuming Text Sending API works fine. This is important as the user would want to be informed when this happens. This would also test the functionality that the farmworker receives weather of the new location assuming weather API and message API works fine. This is important as the user would want to be informed about the weather of the new location and take proactive measures if required.

Test Case Number	TC-08-03 Check if the user receives a text message with a successful message on his location being updated and the weather information. Farmworker should receive the current weather information for the new location along with the confirmation message.

Test Item	Receiving success text message on updating farmworker location with a valid location is tested.
Test Priority	M (Must Have)
Pre-conditions	The system displays the profile page with the current location and the ability to edit the location.
Post-conditions	The location in the profile is correctly updated and the user receives a text message that tells him that his update was successful along with the current temperature at the new location.
Input Specifications	1. Click Edit Profile 2. Change the current location with a valid location 3. Click Save Changes
Expected Output Specifications	The updated profile page is shown with the new location and the user receives a text message with the new location and the temperature at that location.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The new location gets saved in the database correctly. • The old location is removed from the farmworker's profile. • The new location is then displayed on the updated profile page correctly. • The user receives a text message with a successful message that the location was updated in his preferred language • The message is sent in the user's preferred language • The user receives the current temperature information for the new location that was updated Fail criteria: • No confirmation text message is sent to the user • Text message received does not specify what action was taken, for instance, location being updated • The text message does not give information on what was the previous location and what has it been changed to • The message is sent in a language other than the preferred language of the user
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled.

	• The user has permission to edit the location field. • User's mobile number is available in the system • The user's mobile has permissions to receive texts from our system • The text message service is working properly and sends messages when instructed to without any issues. • No weather information text message is sent to the user • Text message received does not specify that the temperature is for the new location
Dependencies	N/A
Traceability	WC_4172: as a user, I should receive the information about the tomorrow's weather

Table 14: TC-08-04 The system should throw an error if the location field is left blank.

Description: I am testing the equivalence partitioning on when a farmworker tries to update his location with an empty/null value in the location field.

Importance: This is important to catch any null / empty entries being persisted in the database as the location is a mandatory field and should always be present in the system. The system should not allow NULLs to be persisted or processed in the system and an appropriate error message should be thrown.

Test Case Number	TC-08-04 The system should throw an error if the location field is left blank
Test Item	Update of Farmworker profile and his location update is tested using empty information.
Test Priority	M (Must Have)
Pre-conditions	The system displays the profile page with the current location and the ability to edit the location.
Post-conditions	An error message is thrown to prompt user to enter a valid location. The location in the profile is not updated and remains what is was before the change was requested until a valid entry is entered.
Input Specifications	1. Click Edit Profile 2. Change the current location with an empty data field 3. Click Save Changes

Expected Output Specifications	• An error message is displayed that the entered location is empty, please try again • the location on the profile is not updated and nothing is processed
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The new location does not gets saved in the database. • The old location is not touched and the field does not get updated. • The old location is displayed on the updated profile page. • An error message is shown that tells the user that the location entered is empty and prompts to enter a valid location Fail criteria: • An empty location gets saved in the database • An error message is not shown • The old location is removed from the database / profile • The profile shows an empty location
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to edit the location field.
Dependencies	N/A
Traceability	WC_4184: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 15: TC-08-05 Update Farmworker location with a valid farm / area / city name.

Description: I am testing the equivalence partitioning on when a farmworker tries to update his location using a valid farm name / city name / area name, he/she should be able to successfully update the location which gets persisted in the database.

Importance: This is a very important feature for proactive farmworkers who would want to self-update their new location and be informed of any weather related harsh conditions or get tips for the new location prior to going out for work.

Test Case Number	TC-08-05 Update Farmworker location with a valid farm name / area/city name.
Test Item	Update of Farmworker profile and his location update is tested using correct information.
Test Priority	M (Must Have)
Pre-conditions	The system displays the profile page with the current location and the ability to edit the location.
Post-conditions	The location in the profile is correctly updated.
Input Specifications	1. Click Edit Profile 2. Change the current location with a valid area/city name like Baker's field / Los Angeles 3. Click Save Changes
Expected Output Specifications	The updated profile page is shown with the new location.
Pass/Fail Criteria	Pass criteria: This test will pass when meet all the following conditions. • The zip code of the new location gets saved in the database correctly. • The old location is removed from the farmworker's profile. • The new location along with the ZIP code is then displayed on the updated profile page correctly. Fail criteria: • The new location does not gets saved in the database. • The new location does not gets updated in the profile page • There is any error on entering a valid farm name, city name, area name • The system does not recognize/updates a valid area name/city/farm name.
Assumptions and Constraints	• The system does not crash while the test case is running and the database is also up and running • Every other required fields are filled. • The user have permission to edit the location field.
Dependencies	N/A
Traceability	WC_4184: As a user I should be able to create and modify my profile - weight / height / other metrics

1. Test Identifier

TC-09 Contractor/Farmer update of farmworker location
Farmworkers who work for contractors work on different farms during the year. Therefore, they have to update their location every time they move to different farm. Since they might forget to update, the contractors can update their location when they are assigning them to a farm. This feature is open for famers as well so they can update the location of the farmworkers that they hire during the year.

2. Test Level

Testing the TC-09: Contractor/Farmer update of farmworker location is in ***unit testing level*** because we are testing one small feature of location unit of our system.

3. Test Class

Erroneous input testing

4. Test Completion Criteria

The test for Contractor/Farmer update of farmworker location will be completed when:

- Input correct location
- Make the changes only if it is being made by contractor or farmer
- Display confirmation message
- Send update text message to farmworker and inform about the change.

Test Cases

Table 1: TC-09-01 Enter correct location.

Description: I used equivalence partitioning technique to design this test case. This test case is important because we always have to test the base condition to make sure that everything works in base case before moving to other cases.

Test Case Number	TC-09-01 Enter correct location.
Test Item	The location being updated by the contractor or the farmer is within the longitude and latitude range of

	our system. For example: is within the longitude and latitude of the central valley.
Test Priority	M (Must have)
Pre-conditions	The contractor or farmer access the location section of the farmworker profile and input the new location (zip code, farm code, or coordinates)
Post-conditions	The admin receives success message. Farmworkers receive text message for the change made on his/her account
Input Specifications	e. Click on location Text area f. Input the new location g. Click on update button
Expected Output Specifications	Receive success message, and text message.
Pass/Fail Criteria	This test will pass if: d) The expected output occurs e) The contractor or farmer list for selected farm contain the farmworker. Otherwise it fails.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184**: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 2: TC-09-02 Enter out of range location parameter.

Description: I used equivalence partitioning technique to design this test case. This test case is important because human error is unneglectable.

Test Case Number	TC-09-02
Test Item	The Contractor ot the farmer inputs a location parameter which is out of the scope of the app.
Test Priority	M(Must have)
Pre-conditions	The Contractor or the farmer choose the target farmworker and inputs the location of the farm that the farmworker has been or will move to.
Post-conditions	A warning message will be displayed to inform the contractor or farmer that the location is out of the scope of the app.
Input Specifications	d. Click on location Text area e. Input the new location f. Click on update button
Expected Output Specifications	An error message that contains the location entered by the user.
Pass/Fail Criteria	This test will pass if: • The Contractor or the farmworker receive appropriate and descriptive message. • After closing the Error message, the user be able to enter a new location. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184**: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 3: TC-09-03 The user trying to make the change is Contractor or farmer.

Description: I used equivalence partitioning technique to design this test case. This test case is important to test the app security

Test Case Number	TC-09-03
Test Item	The user who is trying to update farmworker location is a Contractor or Farmer
Test Priority	M(Must have)
Pre-conditions	The user selects the farmworker profile and changes the location
Post-conditions	The user receives a confirmation message, the farmworker receives a text message.
Input Specifications	d) Search the Farmworkers name to find the profile e) Access the profile f) Enter the new location parameter g) Click on update
Expected Output Specifications	• Receive success message. • Farmworker receives text message.
Pass/Fail Criteria	This test will pass if: e) The user be able to access the page. f) The expected outcome take place. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184**: As a user I should be able to create and modify my profile - weight / height / other metrics

Table 4: TC-09-04 The user accessing trying to make the change is not Contractor or farmer

Description: I used equivalence partitioning technique to design this test case. This test case is important to test the app security

Test Case Number	TC-09-04
Test Item	The user trying t access farmworker profile is not authorized
Test Priority	M(Must have)
Pre-conditions	The user searches the farmworker name and click on the target farmworker profile to edit.
Post-conditions	Receive error message that explains the user is not authorized to access this page.
Input Specifications	e. Search the farmworkers name f. Click on profile to edit the farmworkers profile.
Expected Output Specifications	Error message to inform the user that the access was denied.
Pass/Fail Criteria	This test will pass if: c) The expected outcome take place Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none

Traceability	**WC_4184**: As a user I should be able to create and modify my profile - weight / height / other metrics

Capability Requirement CR-4: Profile system

1. **Test Identifier**

Identifier: TC-10 Create a user profile
Description: User profiles enable our system to store specific information about the users that are registered on our system. These profiles include a user's phone number so that users can be contacted with safety-critical information, and they include biometric information about the users. This biometric data includes the age gender, weight, and height of the user (farmworker) and is important to make sure that all educational content and safety recommendations are personalized and targeted directly at this user.
This test suite tests the functionality of the user profile creation module to make sure that new user profiles can be added to the system, provided that these profiles are complete and have the correct information.

2. **Test Level**

Unit testing. We will be testing at the unit level because we want to check the profile module and because we, as the developers, will be performing the tests. We want to make sure that the data flows as expected and as designed. We want to test the system on boundary cases to make sure our system functions properly in all scenarios.

3. **Test Class**

Erroneous input testing. We want to make sure that the profile system module handles errors in data entering gracefully, and that it correctly interprets the data when the data is entered correctly.

4. **Test Completion Criteria**

The test for "Create a user profile" will be completed when:

- The system creates and stores user profiles when data is complete, correct, and non-duplicate. Created user profiles can be found and displayed in the system.
- The system prevents the creation of user profiles that have missing information and provides appropriate error messages
- The system prevents the creation of user profiles for which data is entered of the incorrect type or of incompatible types and provides appropriate examples of the correct data format and offers support to the user.

- The system prevents the creation of user profiles with duplicate data of profiles already existing in the system, and it provides appropriate information to fix this issue
- The system checks that phone numbers associated with accounts are reachable, and if they are not reachable, it prevents the creation of the user profile and provides instructions on how to fix the problem.

5. Test Cases

Table 1: TC-10-01 – Test creating a user profile with complete and correct information

Description: We came up with this test case using the equivalence partitioning technique. We will divide all potential input data into partitions based on the expected output of our system.
In this test case, the input data should be correct and complete and our expected outcome is that the user profile is created successfully and that it is active in the system.
Why is it important: This test case is important because we must verify that it is possible to register new users into the system. Creating a new profile is the first step in receiving potentially life-saving information and educational content.

Test Case Number	TC-10-01 Test creating a user profile with complete and correct information
Test Item	Creating a new user profile with complete and correct information is tested. The profile creation module will take in a user's name, phone number, height, weight, gender, and work location, and it will store this information in a database.
Test Priority	M (Must Have)
Pre-conditions	The system should display a profile creation page
Post-conditions	The users' pages for farms and for the system should show profiles of associated users
Input Specifications	4. Click create new profile 5. Input all required information into the provided fields 1. Optionally, also input voluntary information 6. Click "Register Profile"
Expected Output Specifications	"Successfully created user profile" message should be shown, with the option to view details of newly created profile, to create a new profile, or to view all active profiles in the system
Pass/Fail Criteria	Pass criteria: *This test will pass if all the conditions below*

	are satisfied. • The user profile is created correctly • The new user profile is added to the "profile" table in the database • A success message is shown to the user who creates the profile Fail criteria: *This test will fail if any of the following occurs.* • The system does not allow the user to create a profile • The new user profile is not added to the database • The new user profile cannot be found on the system • The profile is linked to an incorrect farm. (The profile does not have to be linked to a farm, but if it is linked to a farm, it must be linked to the correct one) • A success message is not shown to the user
Assumptions and Constraints	• All required information is provided • The phone number provided does not already exist in the system • The system is allowing new profiles to be created
Dependencies	No dependencies
Traceability	WC_4184: As a user, I should be able to create and modify my profile - weight / height / other metrics

Table 2: TC-10-02 – Test creating a user profile with missing information

Description: We came up with this test case using the equivalence partitioning technique. We will divide all potential input data into partitions based on the expected output of our system.
In this test case, the input data should be missing some required values, and our expected outcome is that the user profile is not created successfully and that the user is told which required fields are missing.
Why is it important: This test case is important because we must verify that if some profile information is missing, the system can handle this failure gracefully. Additionally, in case of any failures, the user must be made aware that the profile has not been added to the system, and that the user associated with the profile will not receive information about working conditions or educational content until he is added to the system.

Test Case Number	TC-10-02 Test creating a user profile with missing information

Test Item	Creating a new user profile with missing information is tested. The profile creation module will check what required information is missing (i.e. name, phone number, height, weight, gender, or work location), and it will instruct the user to fill in the missing fields before the profile can be added to the system.
Test Priority	M (Must Have)
Pre-conditions	The system should display a profile creation page
Post-conditions	The profile creation page should show an error message indicating that required fields are missing. Missing required fields should be highlighted.
Input Specifications	1) Click create new profile 2) Input all required information except at least 1 required field into the provided fields a) Optionally, also input voluntary information 3) Click "Register Profile"
Expected Output Specifications	The system should stay on the create profile page and the "Unable to create new profile. Some required fields are missing" message should be shown, and missing fields should be highlighted. No new data should be added to the database at this time - the profile should not be added to the system.
Pass/Fail Criteria	Pass criteria: *This test will pass if all the conditions below are satisfied.* • The user profile is prevented from being created • The user is shown an error message • Missing required fields are highlighted Fail criteria: *This test will fail if any of the following occurs.* • The system allows the profile to be created • The new user profile with missing information is added to the database • The profile can be found in the system, though it has not officially been created yet • A success message is shown to the user • An error message is not shown to the user • Missing, required fields are not highlighted
Assumptions and Constraints	• Some required information is missing • The phone number provided does not already exist in the system • The system is allowing new profiles to be created

Dependencies	No dependencies
Traceability	WC_4184: As a user, I should be able to create and modify my profile - weight / height / other metrics

Table 3: TC-10-03 – Test creating a user profile with incorrect/incompatible information

Description: As before, we came up with this test case using the equivalence partitioning technique. We will divide all potential input data into partitions based on the expected output of our system.
In this test case, the input data should have all required fields filled but some fields should be filled with the wrong type of information, and our expected outcome is that the user profile is not created successfully and that the user is told which required fields have incorrect information.

For clarity, an example of "incompatible" or the "wrong type" of information is as follows.
Ex 1: Height: "two-hundred" vs "200"
Ex 2: Phone number: "243-tr3-23+=" vs "555-555-5555"

Why is it important: This test case is important because we must verify that if some profile information is of the wrong type, we prevent this information from being entered in our database. Again, as before, the user must be made aware that the profile has not been added to the system, and that the user associated with the profile will not receive information about working conditions or educational content until he is added to the system.

Test Case Number	TC-10-03 Test creating a user profile with incorrect/incompatible information
Test Item	Creating a new user profile with incorrect information or information of the wrong type is tested. The profile creation module should check all fields to make sure entered-data is of expected type.
Test Priority	M (Must Have)
Pre-conditions	The system should display a profile creation page
Post-conditions	The profile creation page should show an error message indicating that some fields have malformed data. These fields should be highlighted on the profile creation page.
Input Specifications	1) Click create new profile 2) Input all required information correctly except at least 1 required field in which the information is entered with

	incorrect formatting or with incompatible values a) Optionally, also input voluntary information b) Can also input incompatible information in voluntary fields 3) Click "Register Profile"
Expected Output Specifications	The system should stay on the create profile page and the "Unable to create new profile. Some fields have incorrect or incompatible values" message should be shown, and missing fields should be highlighted. Examples of correct formatting should be shown next to the incorrect fields. No new data should be added to the database at this time - the profile should not be added to the system.
Pass/Fail Criteria	Pass criteria: *This test will pass if all the conditions below are satisfied.* • The user profile is prevented from being created • The user is shown an error message • Incorrect fields are highlighted Fail criteria: *This test will fail if any of the following occurs.* • The system allows the profile to be created • The new user profile with incorrect/incompatible information is added to the database • The profile can be found in the system, though it has not officially been created yet • A success message is shown to the user • An error message is not shown to the user • Incorrectly filled fields are not highlighted
Assumptions and Constraints	• Some fields in the profile creation page are filled incorrectly • The phone number provided does not already exist in the system • The system is allowing new profiles to be created
Dependencies	No dependencies
Traceability	WC_4184: As a user, I should be able to create and modify my profile - weight / height / other metrics

Table 4: TC-10-04 – Test creating a user profile when an identical profile already exists

Description: As before, we came up with this test case using the equivalence partitioning technique. We will divide all potential input data into partitions based on the expected output of our system.
In this test case, the input data should have all required fields filled, but the phone number should be the same as that of another profile. Our expected outcome is that the user profile is not created successfully and that the user is told that a profile with that phone number already exists.
Why is it important: This test case is important because we want to make sure that we do not have duplicate profiles in our system. From the system performance and system maintainer's perspective, having duplicate profiles will result in unnecessary data management overhead and could create data consistency issues. From the user's perspective, having multiple profiles with the same phone number could cause a nuisance for the user if he receives multiple notifications about working conditions as a result of his having multiple profiles. The user must then be made aware that the duplicate profile has not been added to the system, and that the user can try to create the profile again with a different number.

Test Case Number	TC-10-04 Test creating a user profile when an identical profile already exists
Test Item	Creating a new user profile with duplicate information as another profile is tested. The profile creation module should check the database to make sure that no profile exists with the same phone number as in the profile to be created.
Test Priority	M (Must Have)
Pre-conditions	The system should display a profile creation page. At least one user profile must exist in the system.
Post-conditions	The profile creation page should show an error message indicating that the entered phone number is already associated with a profile. The phone number field should be highlighted. A "help" button should appear, allowing the user to submit a support ticket requesting more details or requesting that the other profile be deleted.
Input Specifications	1) Click create new profile 2) Input all required information correctly and input the same phone number as the phone number associated with an existing profile a) Optionally, also input voluntary information 3) Click "Register Profile"
Expected Output Specifications	The system should stay on the create profile page and the "Unable to create new profile. The phone number entered

	for this profile is already associated with another profile in the system" message should be shown, and the phone number field should be highlighted. A "help" button should appear. No new data should be added to the database at this time – the profile should not be added to the system.
Pass/Fail Criteria	Pass criteria: *This test will pass if all the conditions below are satisfied.* • The user profile is prevented from being created • The user is shown an error message • Phone number field is highlighted • Help button appears Fail criteria: *This test will fail if any of the following occurs.* • The system allows the profile to be created • The duplicate user profile is added to the database • Duplicate profiles can be found in the system • A success message is shown to the user • An error message is not shown to the user • Phone number field is not highlighted
Assumptions and Constraints	• The phone number provided already exists in the system • The system is allowing new profiles to be created
Dependencies	No dependencies
Traceability	WC_4184: As a user, I should be able to create and modify my profile - weight / height / other metrics

Table 5: TC-10-05 – Test creating a user profile when phone number cannot be verified

Description: As before, we came up with this test case using the equivalence partitioning technique. We will divide all potential input data into partitions based on the expected output of our system.

In this test case, the input data should have all required fields filled, but the phone number should be unable to be verified. Our expected outcome is that the user profile is not created successfully and that the user is told that his or her phone number cannot be verified. We should then provide steps to the user for how to verify his number.

For more information, please see Nexmo's (SMS API) documentation: https://docs.nexmo.com/messaging/sms-api/api-reference. See status code "7", description "Number barred".

Or, refer to Twilio's more verbose explanation of the same error: https://www.twilio.com/docs/api/errors/21610.

The above to resources explain that a number cannot be verified if the owner of that number opted out of receiving automated text messages at some point in the past by texting "STOP" to that number. To fix this issue, a user must simply text "START" to that same number, and the number will then be verified.

Why is it important: This test case is important because we want to make sure that we can reach all users of our system. If a profile's provided phone number is unreachable, the user associated with that profile will not receive any notifications about working conditions from our system, which is a key feature of our system. The user must be made aware that the provided phone number cannot be verified and he or she must be given instructions on how to resolve the problem.

Test Case Number	TC-10-05 Test creating a user profile when phone number cannot be verified
Test Item	Creating a new user profile with a phone number that cannot be verified. The profile creation module should verify this phone number via the Nexmo API when the user attempts to register a profile.
Test Priority	M (Must Have)
Pre-conditions	The system should display a profile creation page.
Post-conditions	The profile creation page should show an error message indicating that the entered phone number cannot be verified. The phone number field should be highlighted. A "verify your phone number" button should appear, giving the user instructions on how to verify his number.
Input Specifications	1) Click create new profile 2) Input all required information correctly a) Optionally, also input voluntary information 3) Click "Register Profile"
Expected Output Specifications	The system should stay on the create profile page and the "Unable to create new profile. The phone number entered for this profile cannot be verified" message should be shown, and the phone number field should be highlighted. A "verify your phone number" button should appear. No new data should be added to the database at this time – the profile should not be added to the system.
Pass/Fail Criteria	Pass criteria: *This test will pass if all the conditions below are satisfied.* • The user profile is prevented from being created

	• The user is shown an error message • Phone number field is highlighted • "Verify your phone number" button appears Fail criteria: *This test will fail if any of the following occurs.* • The system allows the profile to be created • The user profile is added to the database though the phone number has not been verified • A success message is shown to the user • An error message is not shown to the user • Phone number field is not highlighted
Assumptions and Constraints	• The system is allowing new profiles to be created
Dependencies	No dependencies
Traceability	WC_4184: As a user, I should be able to create and modify my profile - weight / height / other metrics

1. Test Identifier

TC- 11 Edit/Update user profile
Farmworkers should be able to update their profile information such as: contact information, password, and location.

2. Test Level

Testing the TC-11: update user profile is in ***unit testing level*** because we are testing one small feature of profile unit.

3. Test Class

Erroneous input testing

4. Test Completion Criteria

The test for user profile update will be completed when:

- Able to view the current information.
- Able to change the information.
- Confirm if phone number is being updated.
- Able to save the information.

Test Cases

Table 1: TC-11-01 Change the information

Description: I used equivalence partitioning technique to design this test case. This test case is really important because the users (farmworkers, contractors, and farmers) should be able to update their profile information when its needed.

Test Case Number	TC-11-01 Enter correct location.
Test Item	The user is able to change his/her own profile information.
Test Priority	M (Must have)
Pre-conditions	The user opens his/her profile and inputs the new information and click save.
Post-conditions	I phone number is being updated the user should receive a code on his/her phone, to confirm the phone number. After confirmation the information will be saved.
Input Specifications	h. Click on profile i. Click on edit j. Enter new information k. Click on submit l. Enter confirmation code m. Click on confirm
Expected Output Specifications	Receive message that the information was updated.
Pass/Fail Criteria	This test will pass if:

	f) The expected output occurs g) When user goes back to profile can see the updated information. Otherwise it fails.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184:** As a user I should be able to create and modify my profile - weight / height / other metrics

Table 2: TC-11- 02: The user doesn't change phone number but changes location.

Description: I used equivalence partitioning technique to design this test case. This test case is needed to make sure the user receives correct feedback (location: message, and text message to inform the change) is being generated by system.

Test Case Number	TC-11-02
Test Item	The user might want to update location only.
Test Priority	M(Must have)
Pre-conditions	The user opens the profile and enter new location and click submit.
Post-conditions	If the location entered by the user is within the range it will be updated otherwise error message will displayed.
Input Specifications	g. Click on profile h. Click on edit i. Enter new location j. Click submit

Expected Output Specifications	Confirmation message displayed.
Pass/Fail Criteria	This test will pass if: • The expected outcome takes place. • The user receives text message about change in location. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184:** As a user I should be able to create and modify my profile - weight / height / other metrics

Table 3: TC-11-03 The user enters wrong phone number

Description: I used equivalence partitioning technique to design this test case. This test case is important because the human error is unneglectable and we have to make sure our system can handle the situation.

Test Case Number	TC-11-03
Test Item	The user makes mistake and enters wrong phone number.
Test Priority	M(Must have)
Pre-conditions	The user open his/her profile and update the phone number.
Post-conditions	The website confirms that the code has been sent to the new phone number. However the user doesn't receive any code on his phone.
Input Specifications	h) Click on profile

	i) Click on edit j) Enter the new phone number k) Click submit l) Click on didn't receive code. m) Choose from resend or check phone number
Expected Output Specifications	• A message that shows the code had been sent • Option to inform the code was not recieved • Option to resend the code. • Option to Change the phone number
Pass/Fail Criteria	This test will pass if: g) The expected outcome take place. Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4184:** As a user I should be able to create and modify my profile - weight / height / other metrics

1. Test Identifier

TC-12 Delete user profile
Farmworkers, farmers, and contractors should be able to delete their profile from the Farmworkers' Safety app anytime they want to.

2. Test Level

Testing the TC-11: update user profile is in ***unit testing level*** because we are testing one small feature of profile unit.

3. Test Class

Erroneous input testing

4. Test Completion Criteria

The test for user profile delete will be completed when:

- The system shows option for delete the user profile
- Ask the user the reason for deleting the profile
- Shows confirmation message to the user
- Delete the profile from the database

Test Cases

Table 1: TC-12-01 Successfully delete the profile

Description: I used equivalence partitioning technique to design this test case. This test case is really important because the users (farmworkers, contractors, and farmers) should be able to deactivate or delete their profile.

Test Case Number	TC-12-01 Successfully delete the profile
Test Item	The user would like to delete or deactivate his/her profile.
Test Priority	M (Must have)
Pre-conditions	The delete profile option is available to user on profile page.
Post-conditions	The system prompts the user to choose the reason why he/she is leaving the system. Prompts the user to confirm that he/she wants to delete the profile. Then success message is being shown to user.
Input Specifications	n. Click on profile

	o. Click on Delete Profile p. Choose the reason q. Click submit r. Click Yes to confirm that you want to delete the profile s. Click on confirm
Expected Output Specifications	After choosing the reason and confirm the he/she wants to delete the profile. User receives a message that the profile was deleted. The profile is being deleted from the database, and user cannot log back in.
Pass/Fail Criteria	This test will pass if: h) The expected output occurs Otherwise it fails.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4180:** As a user, I should be able to disable my account.

Table 2: TC-12- 02: The delete function deletes only the selected profile.

Description: I used equivalence partitioning technique to design this test case. This test case is important because we don't want to lose all the users data when a profile is being deleted.

Test Case Number	TC-12-02

Test Item	After user performs delet on his profile only his account is being deleted and all other accounts or profiles are still available on system.
Test Priority	M(Must have)
Pre-conditions	The user completes the profile delete
Post-conditions	Other User is still able to sign in using his credentials
Input Specifications	k. Click Yes to confirm the Delete l. Click on Sign-in m. Enter other user credentials n. Click log-in
Expected Output Specifications	After a user successfully deleted his profile another user is still able to log in to his account.
Pass/Fail Criteria	This test will pass if: • A user deleted his account successfully • Another user logs in to the system with his own credentials Otherwise it will fail.
Assumptions and Constraints	none
Dependencies	none
Traceability	**WC_4180:** As a user, I should be able to disable my account.

Table 3: TC-12-03 The user is unable to log in with a deleted account

Description: I used equivalence partitioning technique to design this test case. This test is important because we want to make sure that the profile has been deleted entirely from the database and the user cannot log back in without creating an account again.

Test Case Number	TC-12-03
Test Item	The user is not able to log in with the account that he has deleted previously.
Test Priority	M(Must have)
Pre-conditions	The user completes the profile delete module, but later tries to log back in with the deleted account.
Post-conditions	The user enters his deleted account information to log in. The user receives an appropriate message that his username doesn't exist.
Input Specifications	n) Click on profile o) Click on Delete Profile p) Choose the reason q) Click submit r) Click Yes to confirm that you want to delete the profile s) Click on confirm t) Click on log-in u) Input information v) Click on log-in w) Click Ok
Expected Output Specifications	After user deleted the account and tried to log back in with deleted account information. He will receive error message that account doesn't exist.
Pass/Fail Criteria	This test will pass if: h) The expected outcome take place. Otherwise it will fail.

Assumptions and Constraints	we assume that the user did not register again with same information
Dependencies	none
Traceability	**WC_4180:** As a user, I should be able to disable my account.

4. Resources and schedule

4.1 Resources

Currently as per the requirements we developed a prototype to showcase the most important capabilities. Initial testing was done for these capabilities and the document also describes the testing to be done for all the capabilities of the project.
The final testing would be done once all the functionalities are available.
Data Set: We used sample data to test the modules.
Software: PHPUnit testing, PHPSpec, NUnit for ASP.NET
Budget: All test units are available for free, so $).
For integrated and continuous testing, we'll use Selenium.

4.2 Staffing and Training Needs

All team members performed unit testing on their own modules. Shobhit, Juan and Akshay were the lead for test suites and the entire team contributed in discussions.
Training required: The future team would need to decide on what test tool to use for automated testing and train the V&V person on that

4.3 Schedule

Table 16: Testing Schedule

Date	Test Identifier	Responsible person	Resources	Training needs
11/14/2016	TC-01, TC-02	Juan Andrade	NUnit ASP testing	N/A
11/30/2016	TC-10, TC-11, TC-12	Shobhit Agarwal	SQL Server testing	N/A
12/02/2016	TC-07	Akshay Aggarwal	PHPUnit testing	N/A