Functional Testing Document

for

Urban Earth iOS Application, Version 1.0

Prepared by:
Jeffrey Kozik
Niki Wang

November 4, 2021

Table of Contents

1. Introduction

This document is an outline of the goals for testing of the Urban Earth iOS Application. It specifies all of the tests for each class in the project. Functional testing areas include being able to start and stop a user trip manually and automatically along with all of the byproducts associated with these actions such as the sustainability score being displayed and the history being recorded. Other functional test areas include being able to look at graphs summarizing your progress and being able to add and accept friend requests.

1.1 Objectives

1. Outline tests for Urban Earth (UE) and make specific requirements for our functional testing plan.
2. Ensure that all the test cases combined together cover the entirety of the UE app and fulfill the requirements from the SRS document.

2. Project scope

2.1 Scope of Testing

The UE app should be able to perform the following functions: users can manually and automatically start and stop a trip, when the trip stops the data is recorded into the user's history and the sustainability score is displayed. Furthermore, users should be able to add a trip manually or edit a trip manually (including deleting a trip), as well as be able to see a graph of their statistics over time (weekly, monthly, yearly). Finally, the user should get safety alerts and the user should be able to search for friends, add friends, and accept friend requests.

2.2 Testing Schedule

We will test functions within a class once we are done writing that function. If a group of functions can only be tested once they are all written, then we'll wait until we write them all to test them. We will test a class's code once it is complete. Once the tests all pass then we'll move on to the next class. If a certain group of classes can only be tested once they are all written, then we'll test them all once we're done writing them all.

Urban Earth iOS Application	Version: 1.0
Functional Test Plan	Due: 4 November 2021

3. Test Design Specification

3.1 Specification Format

The Urban Earth iOS application has many functions. To make sure all of these functions work as intended, we'll test each function. If the result of the test is what we specified we wanted the result of the test to be, then that function has passed its test. To make sure that a function isn't just getting "lucky" in passing a test, we'll run the tests numerous times to ensure robustness.

3.2 Presumption and Suspension Criteria

If the result of a test is not what we specified that we wanted it to be, then that function fails its test. We will debug this function repeatedly until it passes all tests.

4. Test Specifications

4.1 Manually Starting a Trip

A. The user selects their form of transportation.
B. The user presses start.
C. The app begins displaying the current distance traveled and the current sustainability score of the trip.
D. The app begins tracking the user's location.

4.2 Manually Ending a Trip

A. The user presses stop.
B. The app displays the sustainability score of the trip.
C. The app records the stats of the trip (the form of transportation, the distance, the sustainability score, and the date) to a database.

4.3 Automatically Starting a Trip

A. The user automatically begins a commute (defined as moving outside a substantial radius from the average location the user has been in recently - the exact definition of "substantial" and "recently", denominated in meters and seconds, will be determined later during the coding process).
B. The app begins displaying the current distance traveled and the current sustainability score of the trip.

4.4 Automatically Ending a Trip

A. The user automatically ends a commute (defined as staying within a substantial radius from the average location the user has been in recently - the exact definition of "substantial" and "recently", denominated in meters and seconds, will be determined later during the coding process).
B. The app displays the sustainability score of the trip.

C. The app records the stats of the trip (the form of transportation, the distance, the sustainability score, and the date) to a database.

4.5 Turning Automatic Tracking On and Off

A. The user turns the automatic tracking setting off.
B. The app can no longer always track the user's location, it can only track the user's location when they're using the app.
C. When the user "automatically begins a commute" (as defined in 4.3 Automatically Starting a Trip), a trip isn't started.
D. The user turns the automatic tracking setting on.
E. The app can now track the user's location at all times, not just when the user is using that app.
F. When the user "automatically begins a commute" (as defined in 4.3 Automatically Starting a Trip), a trip is actually started.

4.6 Manually Adding a Trip

A. The user clicks into the history tab of the app.
B. The user clicks the "add" button.
C. A form appears on the screen allowing the user to enter the date, distance traveled, and form of transportation they used.
D. The sustainability score is calculated and the trip is recorded into the database (the form of transportation, the distance, the sustainability score, and the date).

4.7 Manually Editing a Trip

A. The user clicks into the history tab of the app.
B. The user clicks on a trip and presses the "edit" button.
C. A form appears on the screen allowing the user to edit the date, distance traveled, and form of transportation they used.
D. The new sustainability score is calculated and all of the relevant information on the trip is recorded into the database.

4.8 Manually Deleting a Trip

A. The user clicks into the history tab of the app.
B. The user clicks on a trip and presses the "delete" button.
C. All of the information on that trip is deleted from the database.

4.9 Viewing a Progress Graph

A. The user clicks into the graph tab of the app.
B. The app displays a graph showing the users sustainability score over time.

4.10 Changing the Scale of the Progress Graph

A. The user selects the scale of the graph as weekly

B. The app displays the user's sustainability score each day for the last week.
C. The user selects the scale of the graph as monthly.
D. The app displays the user's sustainability score each day for the last month.
E. The user selects the scale of the graph as yearly.
F. The app displays the user's sustainability score each day for the last year.

4.11 Receiving a Safety Alert

A. When the sun sets in the user's location, the user is sent a notification urging them to be cautious and to stay aware because it is dark out.

4.12 Searching for a Friend

A. The user selects the friend tab of the app.
B. The user types in the name of a friend.
C. Search results are displayed showing all users with a name similar to what was searched.

4.13 Adding a Friend

A. After searching for a friend, the user selects the "add" button next to a certain user.
B. This changes the name of the button to "withdraw request"
C. A notification is sent to the user that was "added" them urging them to accept or decline the friend request.
D. If the user selects the "withdraw request" button, a notification is sent to the requisite user telling them that the friend request has been withdrawn.

4.14 Accepting & Declining a Friend Request, Removing a Friend

A. A user receives a notification that someone has requested them as a friend.
B. If the accept button is pressed, the fact that the two are friends is recorded in the database and a notification is sent to the requisite user along with the text of the button changing to "remove".
C. If the decline button is pressed, this is communicated back to the requisite user in the form of a notification and change to the text of the button from "withdraw request" to "add".
D. A user clicks the "remove" button next to a friend.
E. This is communicated to the former friend in the form of a notification and the text of the button is changed to "add". The fact that the two are friends is removed from the database.

5. Functional Tests

5.1 Manually Starting a Trip Class Test

Test	Description	Setup	Expected Result
MSTCT-1	User selects form of transportation	List of forms of transportation is generated	When sustainability score is calculated, it does it with respect to this form of transportation
MSTCT-2	User presses start	A form of transportation is selected and the user is ready to start a trip	Location starts being tracked, and sustainability score calculated
MSTCT-3	Sustainability score is shown as well as distance traveled	Start has been pressed	Live update of sustainability score and distance
MSTCT-4	Location starts being tracked	Start has been pressed	Location is tracked so that sustainability score can be calculated

5.2 Manually Ending a Trip Class Test

Test	Description	Setup	Expected Result
METCT-1	Location no longer shared	Stop has been pressed	Location is no longer tracked
METCT-2	Final score	Stop has been pressed	Final sustainability score is shown
METCT-3	Stats recorded to database	Stop has been pressed	Stats of the trip are recorded to a database

5.3 Automatically Starting a Trip Class Test

Test	Description	Setup	Expected Result
ASTCT-1	“Automatic commute” begun	User moves outside of a predefined radius after a predetermined amount of time	A notification is sent to the user telling them their trip has begun
ASTCT-2	Stats displayed	User moves outside of a predefined radius after a predetermined amount of time	Current distance traveled and sustainability score are displayed

5.4 Automatically Ending a Trip Class Test

Test	Description	Setup	Expected Result
AETCT-1	“Auomatic commute” ended	User stays within predefined radius for predetermined amount of time	A notification is sent to the user telling them their trip has ended
AETCT-2	Final score	User’s commute is done	Final sustainability score for trip is displayed
AETCT-3	Database record	User has stopped	The stats about the trip are recorded to a database

5.5 Turning Automatic Tracking On and Off Class Test

Test	Description	Setup	Expected Result
TATOOCT-1	Automatic tracking off	User turns automatic tracking off	Screen changes toggle to show that automatic tracking is off
TATOOCT-2	Location while using app	Automatic tracking turned off	The app can no longer always track the user’s location, just while they’re using the app
TATOOCT-3	Automatic trip no longer works	Automatic tracking turned from on to off	When a user does the equivalent of starting an “automatic commute” a trip isn’t started
TATOOCT-4	Automatic tracking on	User turns automatic tracking on	Screen changes toggle to show automatic tracking is on
TATOOCT-5	Location always available	Automatic tracking turned on	The app can now always track the user’s location
TATOOCT-6	Automatic trip now works	Automatic tracing turned from off to on and trip started	A trip is actually recorded now because automatic tracking is on

5.6 Manually Adding a Trip Class Test

Test	Description	Setup	Expected Result
MATCT-1	History tab	The user clicks into history tab of app	History page displays showing all previous trips by the user
MATCT-2	Pressing add button	The user clicks the add button	App is told to create popup screen
MATCT-3	Popup screen displays	The app is told to create popup screen	Popup screen displays
MATCT-4	New trip sent to the database	The user enters in the info on the new trip and presses save	The new trip's stats are sent to the database

5.7 Manually Editing a Trip Class Test

Test	Description	Setup	Expected Result
METCT-1	History tab	The user clicks into history tab of app	History page displays showing all previous trips by the user
METCT-2	Pressing edit button	The user clicks on a trip and presses "edit"	App is told to create popup screen
METCT-3	Popup screen displays	The app is told to create popup screen	Popup screen displays
METCT-4	New trip stats sent to the database	The user enters in the edits and presses save	The edits are sent to the database to be recorded

5.8 Manually Deleting a Trip Class Test

Test	Description	Setup	Expected Result
MDTCT-1	History tab	The user clicks into history tab of app	History page displays showing all previous trips by the user
MDTCT-2	Pressing delete button	The user clicks on a trip and presses "delete"	The record of this trip is deleted from the database
MDTCT-3	Returning to history tab	The trip has been deleted	History page no longer shows this trip

5.9 Viewing a Progress Graph Class Test

Test	Description	Setup	Expected Result
VPGCT-1	Graph tab	The user clicks into the graph tab of the app	The graph page is displayed to the user
VPGCT-2	Graph is displayed	The graph page has loaded	A graph showing the user's sustianbaility score over time is displayed

5.10 Changing the Scale of the Progress Graph Class Test

Test	Description	Setup	Expected Result
CSCT-1	Weekly selection	The user selects the scale of the graph as weekly	App is told to display graph with updated scale
CSCT-2	Weekly display	App is told to display graph with weekly scale	New graph is displayed with weekly scale
CSCT-3	Monthly selection	The user selects the scale of the graph as monthly	App is told to display graph with updated scale
CSCT-4	Monthly display	App is told to display graph with monthly scale	New graph is displayed with monthly scale
CSCT-5	Yearly selection	The user selects the scale of the graph as yearly	App is told to display graph with updated scale
CSCT-6	Yearly display	App is told to display graph with yearly scale	New graph is displayed with yearly scale

5.11 Receiving a Safety Alert Class Test

Test	Description	Setup	Expected Result
RSACT-1	Safety alert at sundown	Sun sets in user's location	Notification is sent to user urging them to be safe

5.12 Searching for a Friend Class Test

Test	Description	Setup	Expected Result
SFCT-1	Friend tab	User selects friend tab	App displays search bar to search for friends
SFCT-2	Searching for friend	User types in name of friend into search bar	App is told to search for this friend in the database
SFCT-3	Search results	App is told to search for friend	Search results are displayed showing users with names closest to what was searched

5.13 Adding a Friend Class Test

Test	Description	Setup	Expected Result
AFCT-1	Add friend is executed	User has searched for a friend and presses add next to a user	Database is sent information about who they requested
AFCT-2	Text of button changed	Database is sent information about friend request	Text of button is changed to “withdraw request”
AFCT-3	Friend request notificaiton	Friend has been requested	Notification is sent to potential friend asking them to accept or decline
AFCT-4	Withdrawn friend request	User selects withdraw button	Notification is sent to no longer potential friend telling them request has been withdrawn

5.14 Accepting & Declining a Friend Request, Removing a Friend Class Test

Test	Description	Setup	Expected Result
ADFRCT-1	Friend request notification	A user receives a friend request notification	In app accept or decline buttons are shown

ADFRCT-2	Friend request accepted	Accept button is pressed	The fact that the two are friends is recorded in database and a notification is sent, text of button is changed as well
ADFRCT-3	Friend request declined	Decline button is pressed	The non-friend is notified and the text of the button is changed, and the database reflects this
ADFRCT-4	Remove button clicked	Remove button is clicked next to current friend	The database reflects the fact the two are no longer friends
ADFRCT-5	Notification sent to former friend	Friend has been removed	Text of button changes to “add” and the former friend is sent a notification

6. Defect Resolution

6.1 Issue Tracking

If an issue comes up that cannot be solved by the person who encountered the issue, they will alert the other person in the group through text or verbally. The two group members will discuss the issue and brainstorm solutions for solving it before implementing a solution.

6.2 Issue Resolution

It is the responsibility of the person who encountered the issue to ensure that the issue is solved. That being said, we function best as a group if instead of repeatedly trying to solve something we can’t figure out, we reach out to the other group member to combine our skill sets and resolve the problem.

7. Exit Criteria

The Urban Earth iOS Application Verion 1.0 project is complete when all of the tests described in this document pass (meaning all of the requirements listed in the SRS have been satisfied). Each member of the team is responsible for ensuring that their parts of the code pass their tests.