

Project: Red Alert UV Index Notifications
Team: Group 1
Class: CSE 4322; Fall 2016 Module: Iteration 4 Deliverable: Design Document

Version: [1.4]

Date: [12/05/2016]

Contributors:

John Green

Revision History

<i>Version number</i>	<i>Date</i>	<i>Originator</i>	<i>Reason for change</i>	<i>High level description of changes</i>
1.0	09/12/2016	John Green	Initial draft	
1.1	10/02/2016	John Green	Design and Timelines	Add designs for Iteration 1 and time tables for priority list
1.2	10/24/2016	John Green	Update timelines and progress reports	Modify the timelines to accurately represent project
1.3	11/14/2016	John Green	Updates	Update progress and shrink requirements section down
1.4	12/05/2016	John Green	UI screen shots, updates	Update progress, add screen shots of all views

TABLE OF CONTENTS

1. INTRODUCTION AND PROJECT OVERVIEW.....	3
2. MARKET PREVIEW AND COMPETITORS	4
3. FEATURE SPECIFICATION	5
3.1 BUSINESS Features	5
3.2 SYSTEM Features	5
4. SYSTEMS REQUIREMENTS	6
4.1 “Location Acquisition” Requirements	6
4.2 “UV Index API Lookup” Requirements	6
4.3 “Alert Notifications” Requirements	6
4.4 “Exposure Tracking” Requirements	6
5. SOFTWARE PROCESSES AND INFRASTRUCTURE	7
5.1 Hardware, Software, and Infrastructure	7
5.2 Source Code and Version Control	7
5.2.1 Compile on Android Studio W/ Gradle	7
5.3 UML.....	8
5.3.1 Landing Page Activity.....	8
5.3.2 Hourly Check Activity	9
5.4 Prototypes	10
5.4.1 Landing Page Prototype.....	10
5.4.2 Notifications Prototype	11
5.4.3 Screen Shots.....	12
6. ASSUMPTIONS AND CONSTRAINTS	18
6.1 ASSUMPTIONS	18
6.2 CONSTRAINTS	18
6.3 Out of Scope material	18
7. RISK MANAGEMENT	19
8. TIME TABLES AND PRIORITIZATION	20
9. TEST PLAN.....	23

1. Introduction and Project Overview

Modern consumers are keenly aware of the dangers posed by ultra violet radiation (UV). This knowledge is demonstrated by the rising use of sunscreen and concerns over skin cancers. However, actions to protect one's self and others from the sun are generally considered for extended outings only. The short periods of exposure throughout the day, constitutes the majority of one's risk to UV. Providing the user with the information to better regulate the daily risks is the primary goals of this project

The goal of this project is the creation of a background Android application to monitor the current UV index at the user's location. The application has three distinct functionalities. First, the application allows the user to retrieve UV index information, based on a provided location. The location can be user defined or pulled from the phones location data. The information provided by the application consists of the current UV index and the predicted hourly index levels. Second, the application notifies the user, through Androids alert message functionality, of UV levels exceeding a defined threshold. While operating in the background of the phone, the application will periodically lookup the UV index value for the phones current location. If the UV level exceeds the user's defined threshold, the application generates a warning message and displays the message with Android alert. Third, the application tracks the users total hourly and daily exposure rates. Every alert triggered by the application is logged for future reference. A persistent hourly and daily exposure total may be accessed by the user for informational purposes.

2. Market Preview and Competitors

The current market is serviced by local news outlets and on phone weather applications. These services are limited and passive options. Local television and radio news outlets provide an inflexible service that requires the users to modify their routing. If a user misses a UV bulletin, the user may proceed with the day unaware of the risk. On phone weather applications provide current UV index information on demand. However, the information can be obfuscated by all the other weather information present. The signature difference between these two options and Red Alert UV Index Notifications is the active alerts. The previous two options require users to seek out the information. Once setup, Red Alert provides the user with UV index warnings without user input. The second defining difference is the exposure tracking. The data provided will allow the user to make informed decisions about their daily schedule.

3. Feature Specification

3.1 BUSINESS FEATURES

The following is a list of business Features:

Feature 1: User can setup the application to use location data

Feature 2: The application shall allow the user to lookup current UV data based on zip code

Feature 3: The application shall allow the user to lookup UV hourly projection data based on zip code

Feature 4: The application shall check UV index levels hourly

Feature 5: The application shall alert the user when UV index exceeds level five

Feature 6: The application shall log the hours of exposure and level

Feature 7: The application shall keep a count of high UV index days

3.2 SYSTEM FEATURES

The following is a list of system Features:

Feature 1: System uses the user specified default location data for UV lookup

Feature 2: System uses phone location data for UC lookup if selected by the user

Feature 3: System use the Envirofacts Data Service API for UV index lookup

Feature 4: System alerts the user with Andriod's default notification system

Feature 5: System operates as a background application

Feature 6: System stores exposure data in a local sqlite database

4. Systems Requirements

4.1 “LOCATION ACQUISITION” REQUIREMENTS

Sequence No:	Short Description	Detailed Description
001	User default setup	During initial setup, the user will define a default location “home”. This will be used for current location when the phones location data is unavailable or disabled
002	User requested location	User provides a location, city name or zip code
003	Phone location data	Current location is gathered from the phones internal location data

4.2 “UV INDEX API LOOKUP” REQUIREMENTS

Sequence No:	Short Description	Detailed Description
001	Current UV index lookup	Application requests current UV index level from the Envirofacts Data Service API using the provided zip code
002	Hourly predictions lookup	Application requests daily UV index predictions from the Envirofacts Data Service API using the provided zip code

4.3 “ALERT NOTIFICATIONS” REQUIREMENTS

Sequence No:	Short Description	Detailed Description
001	Hourly index check	Every hour the application will perform a current UV index lookup for the current location.
002	Alert for high UV levels	If the current UV index value is above the threshold, issue an alert to the user

4.4 “EXPOSURE TRACKING” REQUIREMENTS

Sequence No:	Short Description	Detailed Description
001	Persistent exposure hours	Track total accumulated hours at each exposure level
002	Persistent exposure days	Track total number of days an alert has been issued

5. Software Processes and Infrastructure

5.1 HARDWARE, SOFTWARE, AND INFRASTRUCTURE

- Mobile phone with sufficient performance to run Android
- Android operating system version 4.4 or later
- Mobile data connection or wifi connectivity
- GPS location capabilities

5.2 SOURCE CODE AND VERSION CONTROL

The source code for the project is stored as a public repository on GitHub:

<https://github.com/johngreen-uta/RedAlertUVNotifications>

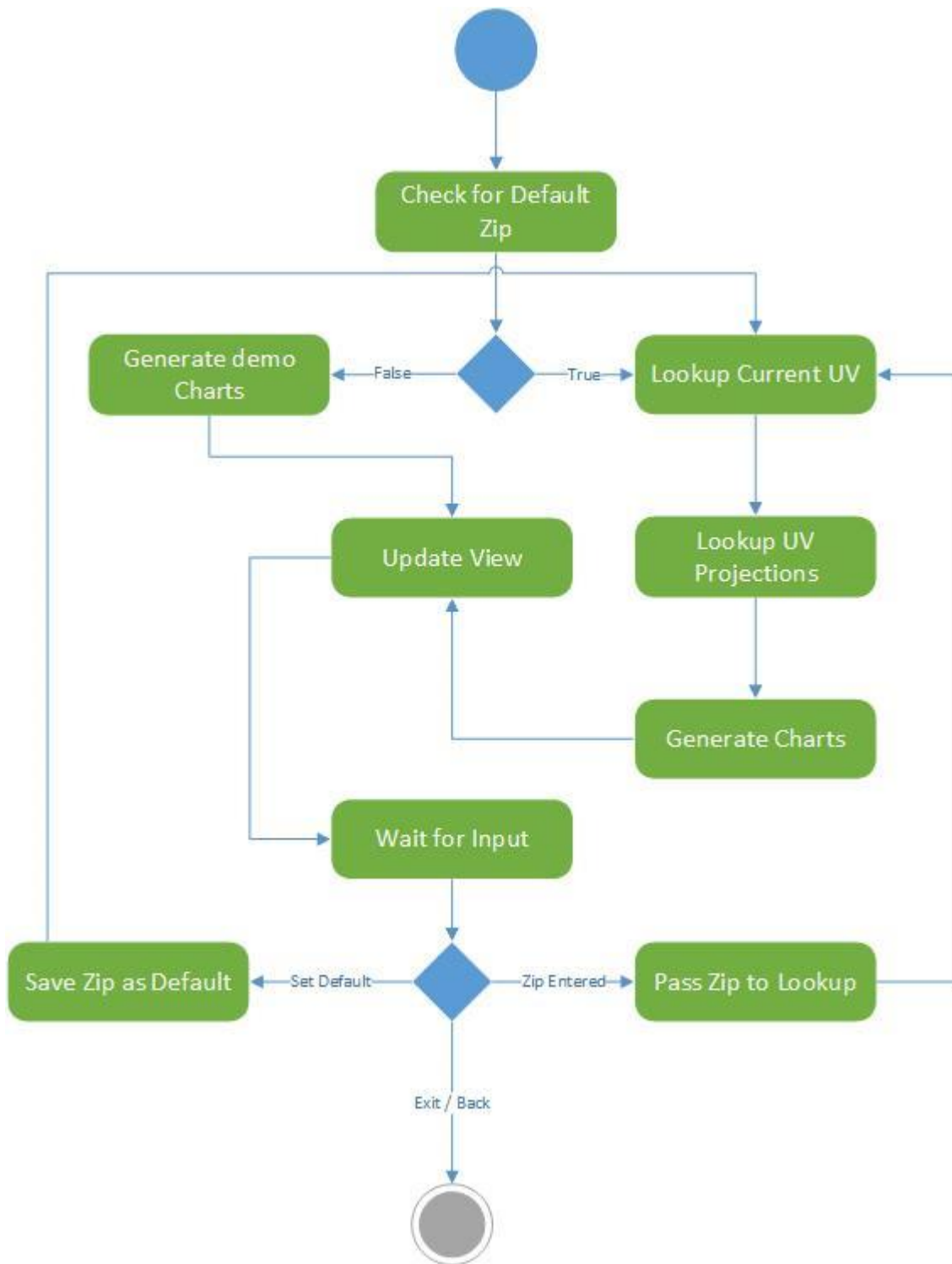
5.2.1 COMPILE ON ANDROID STUDIO W/ GRADLE

After cloning the repository and importing into Android studio

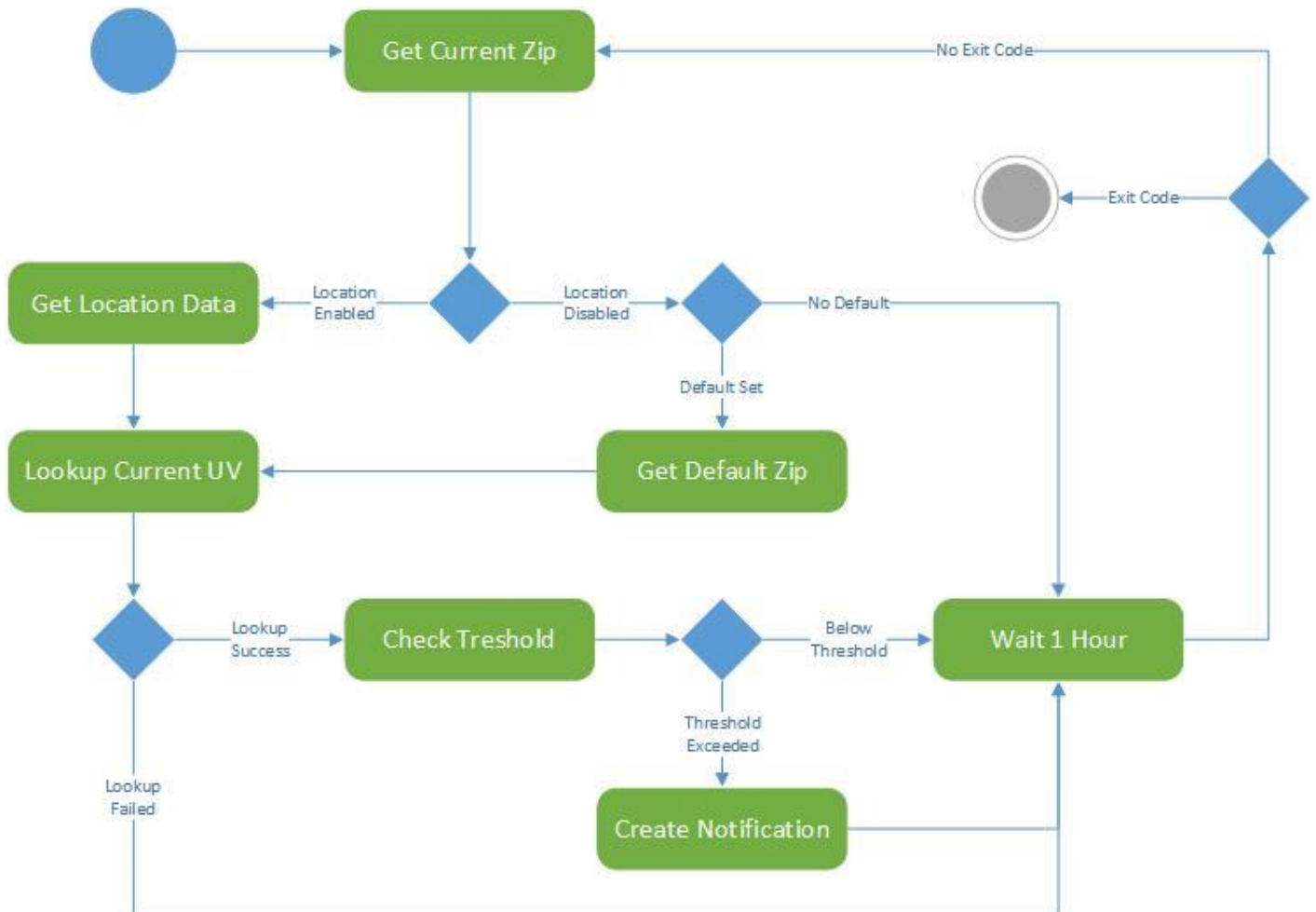
1. click on the drop down menu on the toolbar at the top (usually with android icon and name of your application)
2. select Edit configurations.
3. click plus sign at top left corner or press alt+insert.
4. select Gradle.
5. choose your module as Gradle project.
6. in Tasks: enter assemble.
7. press OK.
8. press play.

5.3 UML

5.3.1 LANDING PAGE ACTIVITY

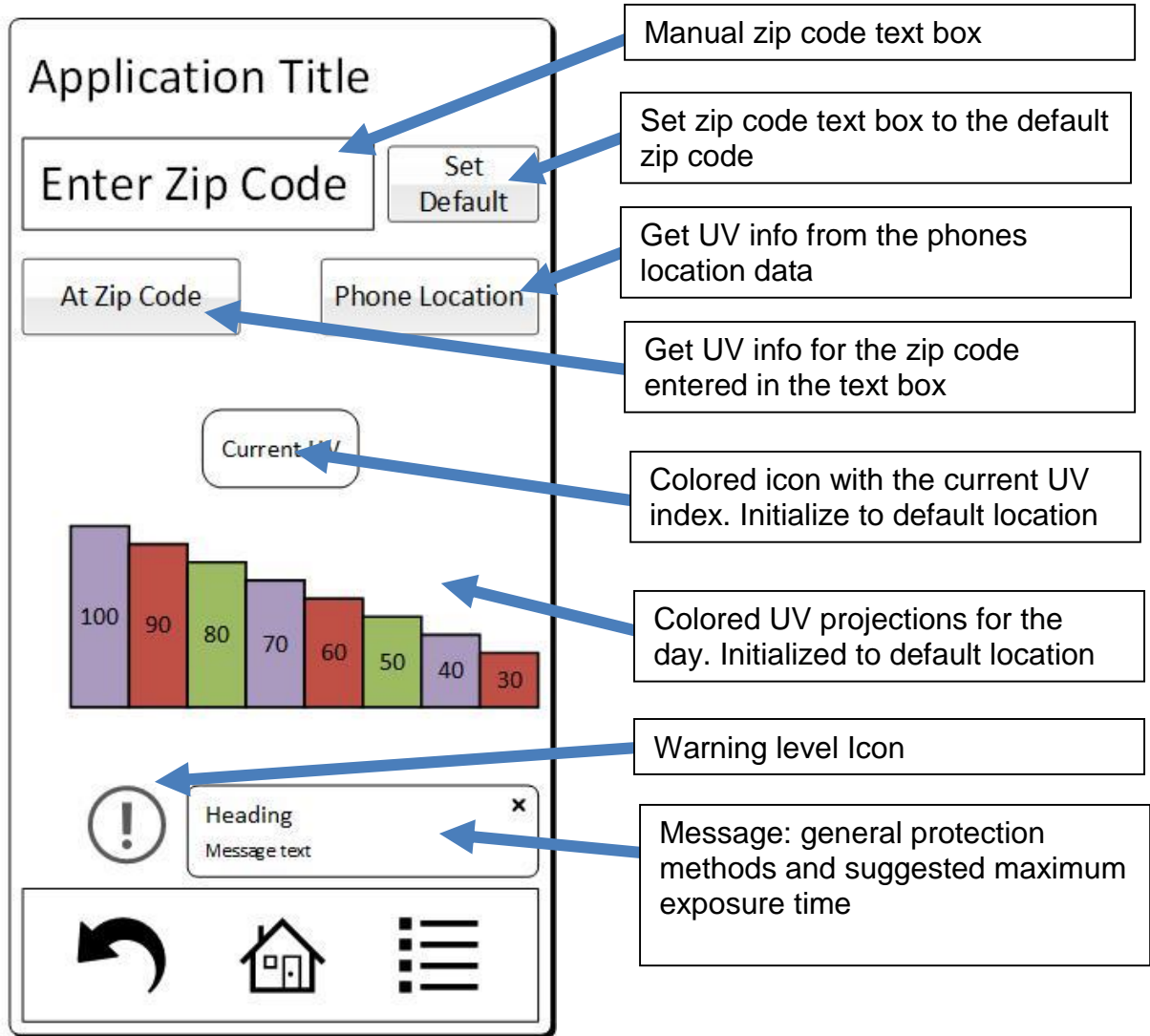


5.3.2 HOURLY CHECK ACTIVITY



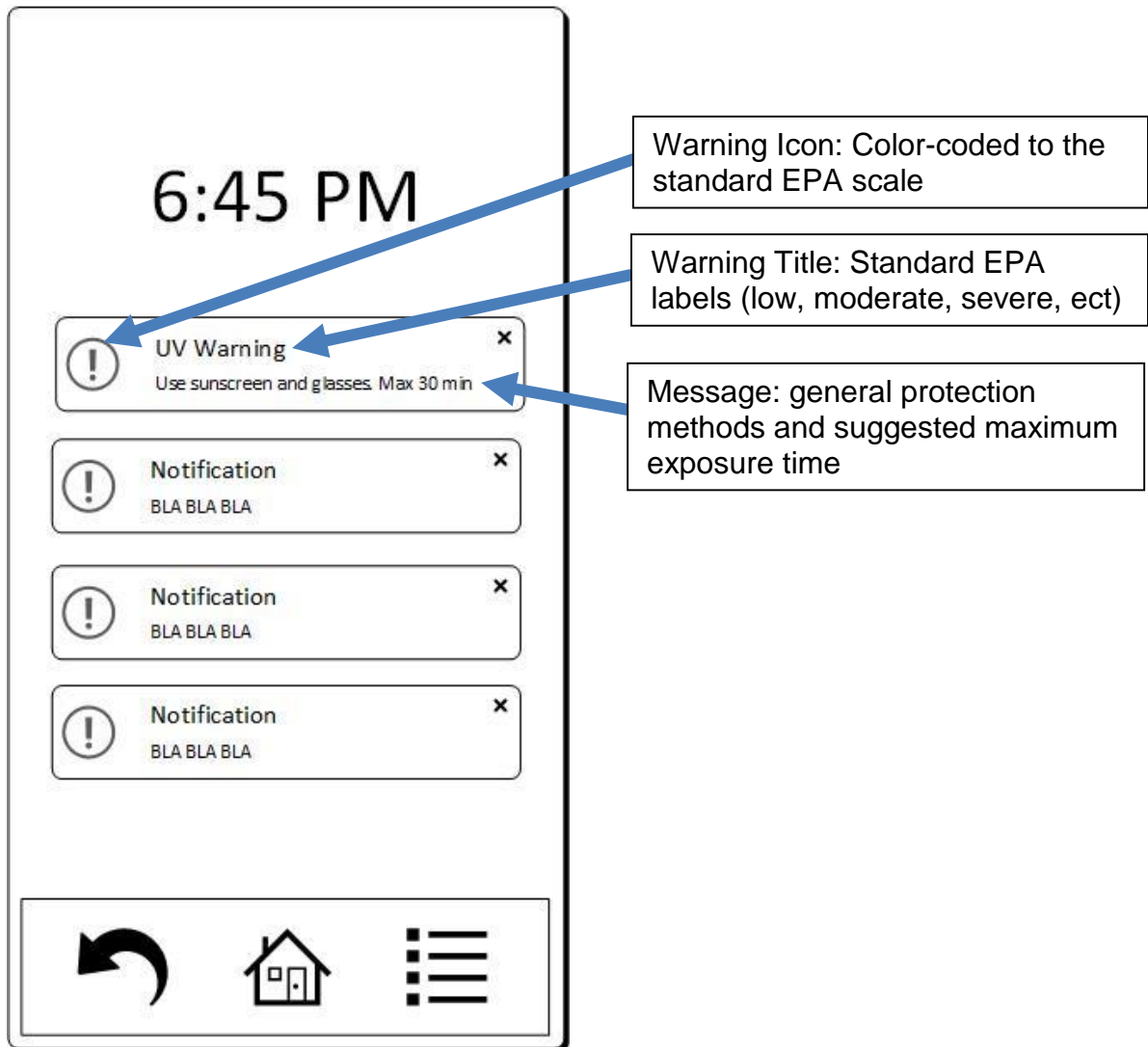
5.4 PROTOTYPES

5.4.1 LANDING PAGE PROTOTYPE



Application main landing page

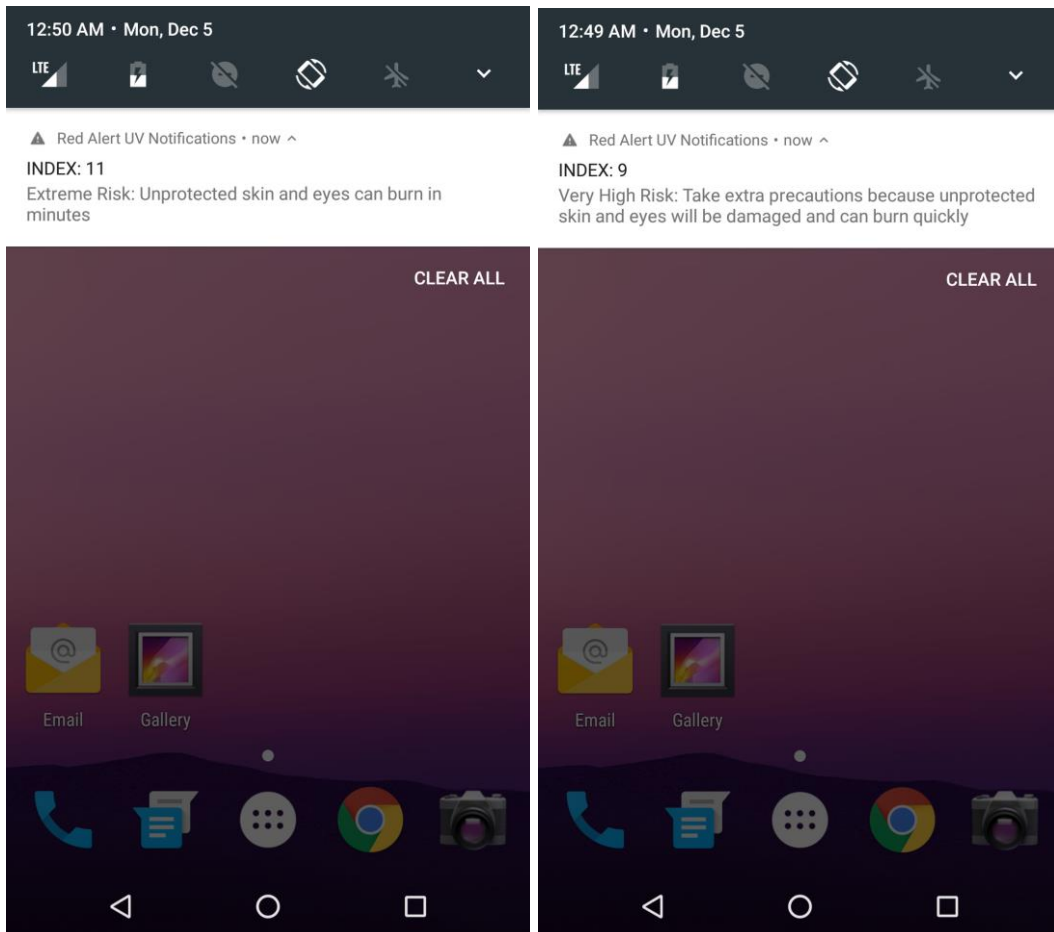
5.4.2 NOTIFICATIONS PROTOTYPE



Notifications displayed on Androids home screen

5.4.3 SCREEN SHOTS

Android notification bar

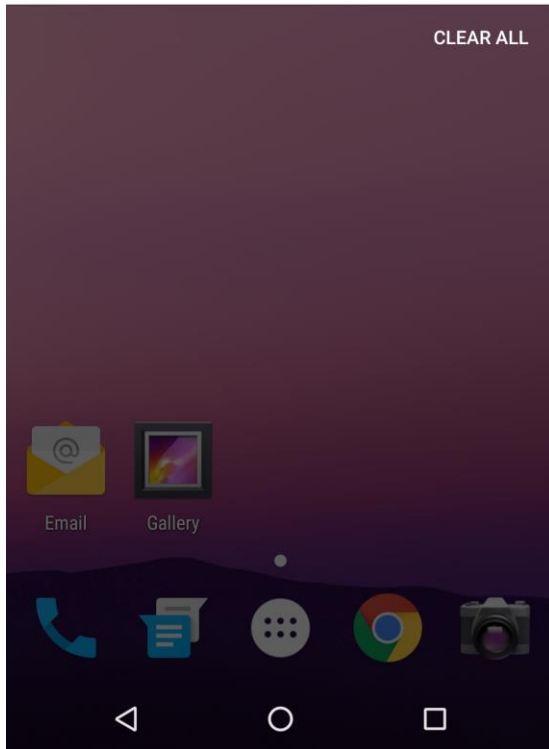




▲ Red Alert UV Notifications • now ^

INDEX: 7

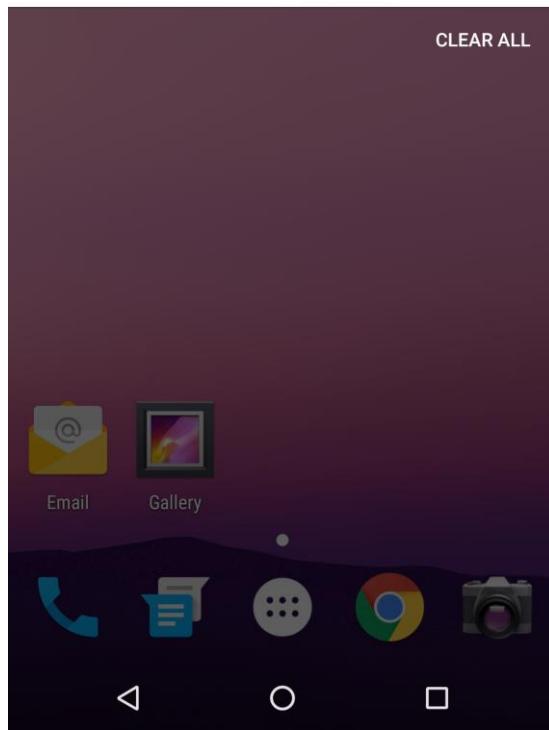
High Risk: Protection against skin and eye damage is needed



▲ Red Alert UV Notifications • now ^

INDEX: 4

Moderate Risk: Low Risk of harm from unprotected sun exposure

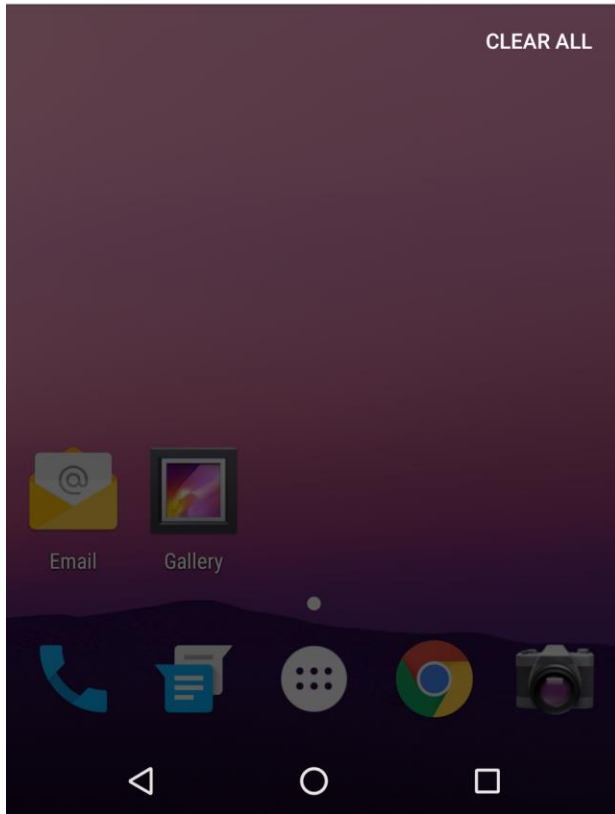




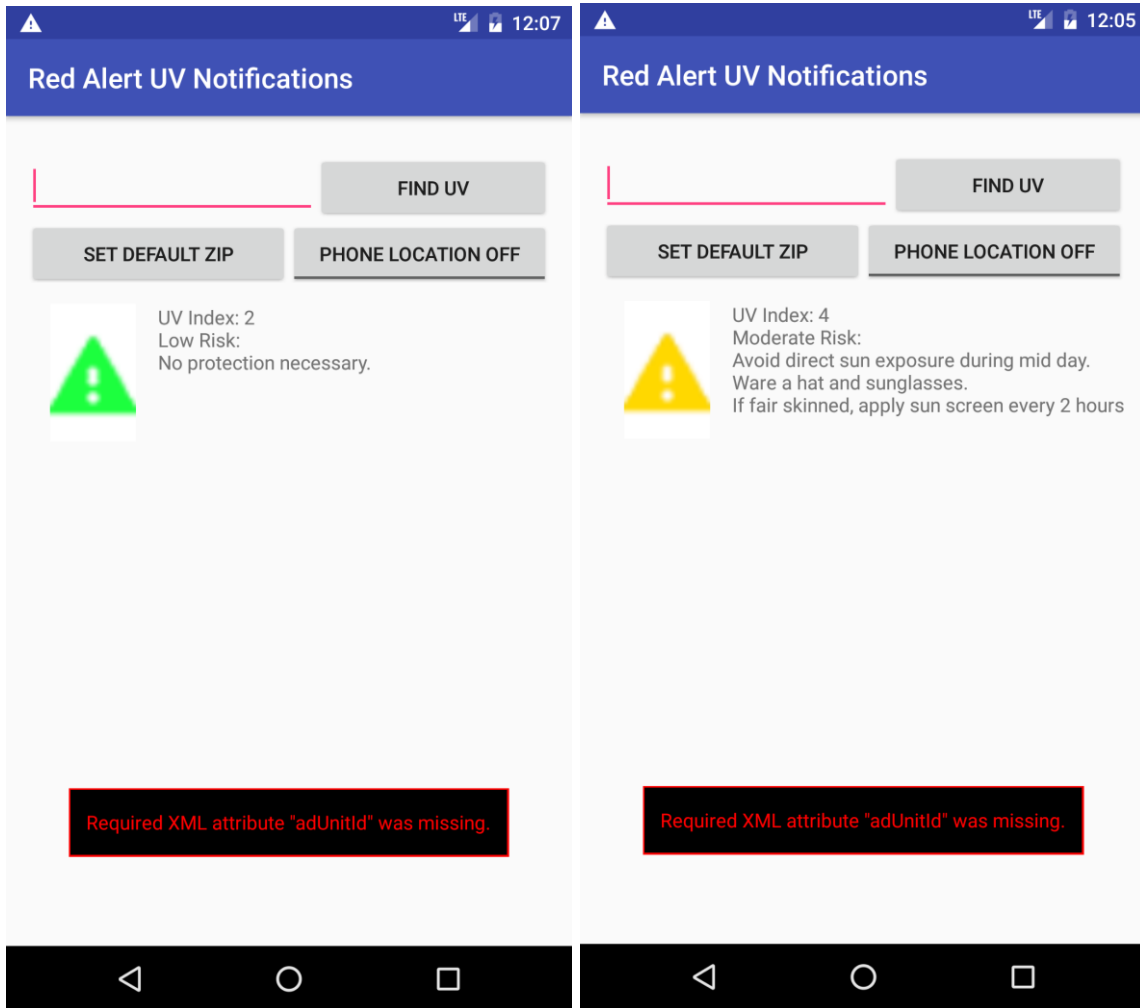
■ Red Alert UV Notifications • now ^

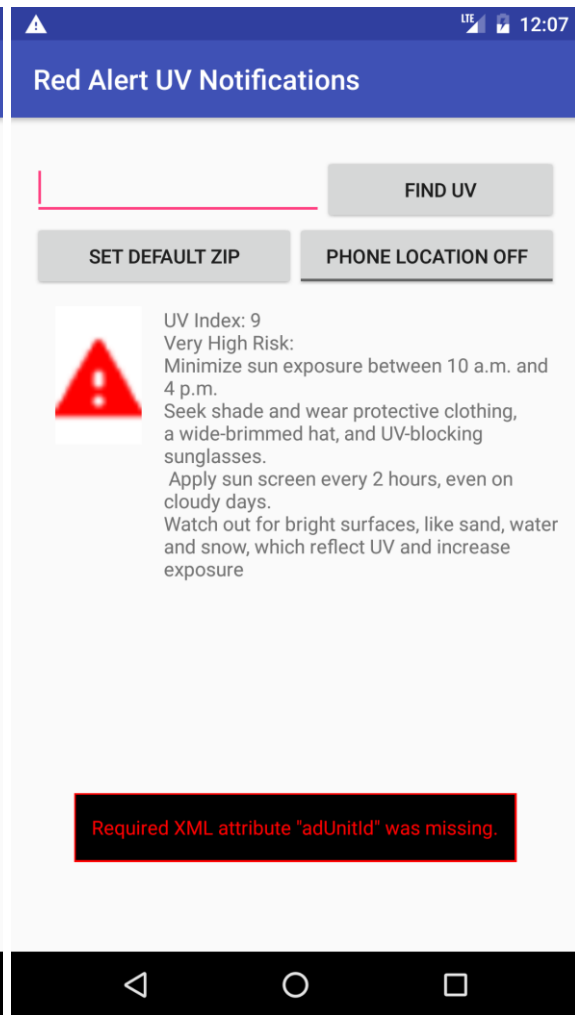
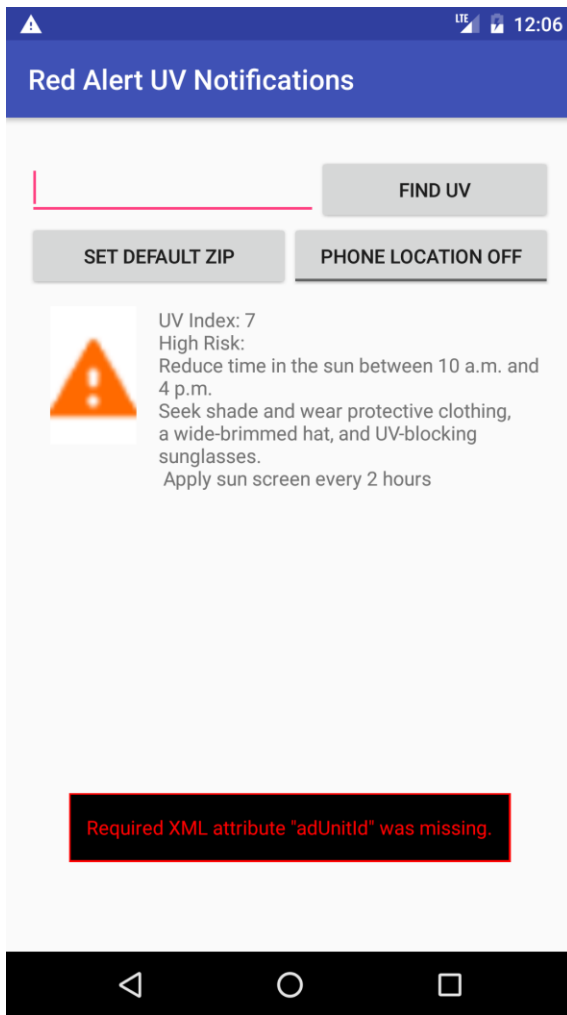
INDEX: 2

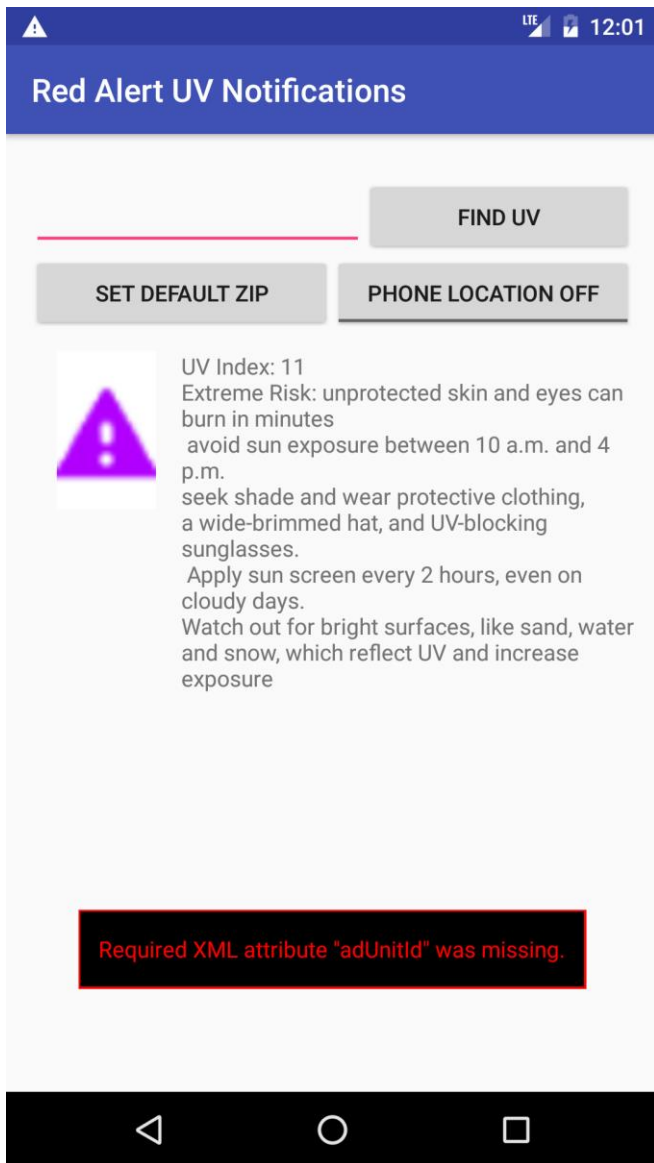
Low Risk: Low danger from the sun's UV rays for the average person



Android main views







6. Assumptions and Constraints

6.1 ASSUMPTIONS

The following is a list of assumptions:

- User has a mobile phone capable of running Android operating system
- Ignore compliance issues
- Location data is available

6.2 CONSTRAINTS

The following is a list of constraints:

- Schedule very aggressive
- Team lacks access to multiple android devices for testing
- Team lacks experience working with Android operating system

6.3 OUT OF SCOPE MATERIAL

The following is a list of “out of scope” material:

- Post Project maintenance is not covered
- Commercial distribution unavailable

7. Risk Management

Personnel scheduling Constraints:

As this project is a small team, any down time due to illness or other affliction will significantly affect the project timetable. To mitigate this risk, the project will be broken down into small modules and prioritized based on necessity. Modules that do not significantly impair usability of the application will be dropped, if a team member is incapacitated.

Personnel knowledgebase:

Team members have limited knowledge and experience working with the Android operating system. To reduce the learning curve associated with a new platform, the project will be implemented using android studio. This IDE is similar to many existing IDEs, is well documented, and preformats the project to work on an Android device. Using Android studio allows focus on program logic over implementation syntax.

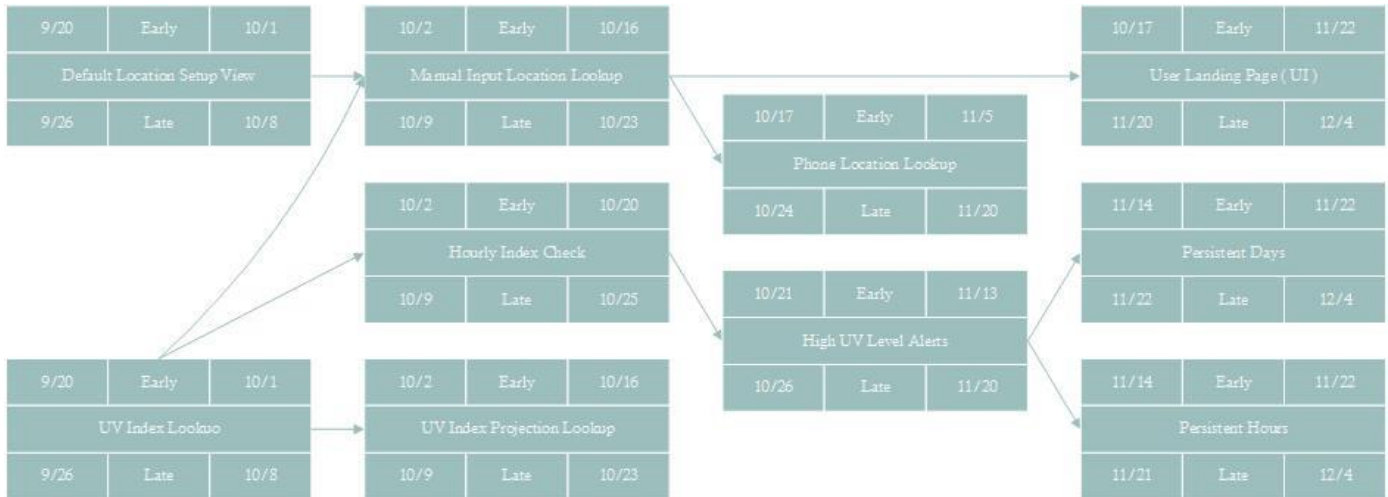
Phone location availability:

GPS phone location will not be available in all situations. To create a smooth user experience when location data is unavailable, the last known location will be used to approximate the current location. This approach will provide a reasonable accurate response in most cases.

Compatibility issues across the android platform:

The application will be developed to work on android version 4.4 Kit Kat. Developing for version 4.4 provides compatibility to %80 of the current android market. In addition, the use of android studio allows for in IDE emulation of many android phones and tablets.

8. Time Tables and Prioritization



Task	Priority	Description	Start	End	Status
Default setup view	1	Basic UI for users to input a default location and select phone location data options	9/20	10/8	Complete
UV index lookup	1	Module that accepts a zip code and connects to the envirofacts api to retrieve the current UV index. Returns index number.	9/20	10/8	Complete
Manual input lookup	2	UI element that allows the user to select a zip code and retrieve the current UV index at that location	10/2	10/23	Complete
Hourly Lookup	2	Module that schedules a UV index lookup based on default or phone location. Passes index value to the notification module	10/2	10/25	Complete
Projection lookup	4	Module accepts a zip code and retrieves hourly projections from the envirofacts api. Generates a bar graph of the values. Returns the graph	10/2	10/23	Cancelled
Phone location lookup	3	Module that connects to the Android api and retrieves the phones current location. returns the zip code	10/17	11/20	Complete
Alert notifications	3	Module that receives an index value and generates a notification. Returns the notification	10/21	11/20	Complete
User landing page (user experience)	4	Integrate graphs and notifications into the UI. Create icons and streamline the user experience	10/17	12/4	Complete
Exposure day count	4	Module that track the number of days that exceeded a set UV level and integrate in the UI	11/14	12/4	Cancelled
Exposure hours count	4	Module that track the number of hours that exceeded a set UV level and integrate in the UI	11/14	12/4	Cancelled

Initial Delays (Iteration 1):

Acclimation to the android environment and the adoption of a version control system, Github, has pushed the initial start dates on design and the first two modules. This delay has been compensated for, by pushing the low priority Projection Lookup module.

Future Man Hours Delays:

The probability of a team member becoming ill increases as the semester continues. If sickness delays the development of the project, the low priority modules will be cut from the final product. The development tasks have been scheduled to limit dependence on previous modules. The final increment of the project consists of the lowest priority modules, allowing functionally significant modules some softer deadlines.

Progression (Iteration 2):

Iteration 2 has been more productive than iteration 1. The basic user interface for zip code based UV index lookup completed earlier than expected (10/18/2016). The manual zip code lookup now accepts a user provided zip code and displays the current UV index information. However, the hourly projections module has been postponed to allow more time on higher priority modules. This postponement has allowed work on the core feature, alert notifications, to begin a week early.

Future Projections:

The postponement of the hourly projection module has highlighted the under estimated time to completion on the modules. As the first round of projections were aggressive, the two low priority data tracking modules may be dropped from the project. If the phone location and alerts modules are not fully completed by the end of iteration 3, the data tracking will be cut in favor of improving the user interface and hourly projections.

Progression (Iteration 3):

Iteration 3 has progressed according to the schedule. The hourly lookup scheduler has been completed and runs every hour between 9am – 7pm. The notification module has also been completed. The notification is created once and updated every hour. Work has started on the phone location data. The location data is about half way through production.

Future Projections:

The postponement of the hourly projection module has allowed the team to catch up to the schedule. After the phone location data is finished, the hourly projections will be resumed. The final stretch will be building a unified user interface.

Final Deliverable (Iteration 4):

Iteration 4 has progressed slowly. The phone location lookup has been completed. However, the translation from lat/lon coordinates to zip code requires an active data connection. The notification strings have been updated with relevant information. The main view now has a color coded icon for each uv level. The main view also contains detailed information on precautions not contained in the notification. The main view contains an advertising space at the bottom. Ads will require a google play store id to be requested.

Cancelled Features:

Three initial features have been cancelled. The holiday season and end of semester limited the available time to development. The cancelled features were not relevant to the core functionality of the app. The decision was made to forgo development on the cancelled features to focus on the user experience.

9. Test Plan

ID	4.1.001.1
Title	Default lookup
Pre-conditions	Default location set to: 76180
Test Steps	1: pass default zip code to class 2: return results
Expected Results	Int: $0 \leq n < 12$
Post Conditions	N/A
Test Data	N/A

ID	4.1.001.2
Title	Default lookup
Pre-conditions	Default location not set
Test Steps	1: pass default zip code to class 2: return results
Expected Results	Int: -1
Post Conditions	N/A
Test Data	N/A

ID	4.1.002.1
Title	User requested lookup
Pre-conditions	Valid zip code
Test Steps	1: pass default zip code to class 2: return results
Expected Results	Int: $0 \leq n < 12$
Post Conditions	N/A
Test Data	N/A

ID	4.1.002.2
Title	User requested lookup
Pre-conditions	Invalid zip code
Test Steps	1: pass default zip code to class 2: return results
Expected Results	Int: -1
Post Conditions	N/A
Test Data	N/A

ID	4.1.003.1
Title	Phone location lookup
Pre-conditions	Phone location enabled
Test Steps	1: call class 2: return results
Expected Results	Int: $0 \leq n < 12$
Post Conditions	N/A
Test Data	N/A

ID	4.2.001.1
Title	Current UV lookup
Pre-conditions	Passed valid zip code
Test Steps	1: connect to API 2: retrieve data 3: parse UV index
Expected Results	String: $0 \leq n < 12$
Post Conditions	N/A
Test Data	N/A

ID	4.2.001.2
Title	Current UV lookup
Pre-conditions	Passed invalid zip code
Test Steps	1: connect to API 2: retrieve data 3: parse UV index
Expected Results	String: NULL
Post Conditions	N/A
Test Data	N/A

ID	4.2.002.1
Title	Hourly prediction lookup
Pre-conditions	Passed valid zip code
Test Steps	1: connect to API 2: retrieve data 3: parse UV predictions 4: store results in array
Expected Results	Array of 12 ints between 0 and 12
Post Conditions	N/A
Test Data	N/A

ID	4.2.002.2
Title	Hourly prediction lookup
Pre-conditions	Passed invalid zip code
Test Steps	1: connect to API 2: retrieve data 3: parse UV predictions 4: store results in array
Expected Results	Empty array
Post Conditions	N/A
Test Data	N/A

ID	4.3.001.1
Title	Hourly UV index check
Pre-conditions	Phone time corresponds to scheduled lookup
Test Steps	1: scheduled function is triggered 2: call hourly lookup passing a zip code 3: pass return value to alert notification 4: alert notification returns a flag
Expected Results	Bool: true
Post Conditions	N/A
Test Data	N/A

ID	4.3.002.1
Title	Alert notification
Pre-conditions	Passed an int between 0 and 12
Test Steps	1: switch based on int's value 2: generate string based on value 3: generate notification from string and value 4: post notification to OS 5: return post result
Expected Results	Bool: true
Post Conditions	N/A
Test Data	N/A

ID	4.4.001.1
Title	Persistent exposure hours
Pre-conditions	Preset int n to some value
Test Steps	1: call add time function 2: value is incremented 3: value is stored 4: refresh display value
Expected Results	Displayed value equals n+1
Post Conditions	N/A
Test Data	N/A

ID	4.4.001.2
Title	Persistent exposure hours
Pre-conditions	Preset int n to some value
Test Steps	1: call get time function 2: value is retrieved 3: value is returned
Expected Results	Int: n
Post Conditions	N/A
Test Data	N/A

ID	4.4.002.1
Title	Persistent exposure days
Pre-conditions	Persistent exposure hours
Test Steps	1: call add time function 2: value is incremented 3: value is stored 4: refresh display value
Expected Results	Displayed value equals n+1
Post Conditions	N/A
Test Data	N/A

ID	4.4.002.2
Title	Persistent exposure hours
Pre-conditions	Preset int n to some value
Test Steps	1: call get time function 2: value is retrieved 3: value is returned
Expected Results	Int: n
Post Conditions	N/A
Test Data	N/A