AirCasting Application Development for iOS Platform

POSTERA CRESCAM LAUDE

THE UNIVERSITY OF MELBOURNE

Md Akmal Hossain (662254)

Supervisor: Prof. Richard O. Sinnott

Melbourne School of Engineering

University of Melbourne

This report is submitted for

COMP90019 Distributed Computing Project (25 credit points)

(Software Development Project)

Acknowledgement

My sincere acknowledgement goes to the project supervisor Professor Richard O Sinnott for his support and guidance throughout the project. I would also like to extend my acknowledgement to my project partner Renji Luke Harold, technical consultant Dr. Marcos Nino-Ruiz and Jiajie Li for their continuous support with technical knowledge, which have been invaluable for completion of this project.

Declaration

I hereby declare that

- *This thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.*

- *Where necessary I have received clearance for this research from the University's Ethics Committee (Approval Number ....) and have submitted all required data to the Department*

- *The thesis is 2881 words in length (excluding text in images, table, bibliographies and appendices).*

Md Akmal Hossain

November 2015

Abstract

An iOS application for recording air pollution data has been developed. The focus of the project is to replicate an existing environmental data recording application, which is compatible for Android platform. The project also comprises of developing a routing solution based on air pollution level of the environment. The application has been developed in Xcode 6.4 with Swift 1.2 programming language and routing algorithm has been developed implementing A* algortihm.

Table of Contents

List of Figures

1.0 Introduction

1.1. Purpose of Project

The purpose of the project is to develop an iOS application that is capable of recording certain environmental data (Decibel, Humidity, Temperature and Particulate Matter), which can be used to measure air pollution level. The collected data can be used for various purposes such as sustainable development, urban planning, pollution identification, environment quality improvement etc. At present, a US based non-profit organization HabitatMap offers a web-based platform named AirCasting to collect environmental pollution data using smart phone application and external sensor device. The existing AirCasting application is only operational within Android platform, which isolates around ~40% of the iOS smart phone users [5]. The objective of this project is to provide a solution to make the usage of AirCasting application more universal. This will be beneficial to projects like Australian Urban Research Infrastructure Network (AURIN) and Clean Air and Urban Landscape (CAUL) initiatives. The application can also be beneficial to individuals, who are environmental enthusiasts, by providing location specific pollution information.

1.2. Scope of Project

The project scope has been defined based on the existing AirCasting platform and Android application. The first scope of the project is to develop an iOS application that replicates the existing AirCasting application for Android platform. Replication needs to include both operational and visual aspect of the existing application. As an additional feature, application should allow real time streaming (upload) of data to the remote AirCasting database. The second scope of the project is to develop a routing process based on A* algorithm which can recommend route between source and destination considering pollution rating of the locations.

1.3. Report Outline

The report has been organized in following segments. Segment 2 titled "Background", presents a brief overview of the existing projects where use of the application will be beneficial. It also includes brief description about the AirCasting platform and AirBeam sensor device used in this project. Segment 3 titled "Design Specification" describes the operation of the AirCasting server and current Android application architecture. Segment 4 titled "Design Implementation" contains the details of the developed iOS application architecture and functionalities. Segment 5 titled "Challenges and Solutions" outlines the key challenges encountered during project development and solutions implemented to overcome them. At the end of the report, recommended future works for the application have been discussed in the segment 6 titled "Future Directions" and conclusion has been drawn in segment 7 titled "Conclusions".

2.0 Background

2.1. AURIN

Australian Urban Research Infrastructure Network (AURIN) is a national collaboration among several Australian Government entities, National Research Infrastructure for Australia (NCRIS) and University of Melbourne to establish an e-research infrastructure to facilitate better decision making for urban settlement and development.

Clean Air is of the many different projects that comprise AURIN. It is an initiative to estimate the air quality of different localities. The data can allow authorities to take appropriate actions for more polluted areas to improve living quality and individuals to travel within less polluted areas etc. [1]

2.2. CAUL

Clean Air and Urban Landscape (CAUL) is a research project funded by Department of Environment and supported by Green Building Council Australia (GBCA) with participation of University of Melbourne, RMIT University, University of Wollongong and University of Western Australia. The project focuses on environmental research within urban localities to help government and industry to make better decisions about future planning and development. [8]

2.3. AirCasting

AirCasting is an open source online platform, which offers recording, mapping and sharing of environmental and health data through smart phone application and external sensor devices. The application communicates with an external sensor device to record pollution level and visualizes the data within a web application named CrowdMap.

AirCasting platform is used among environmental research organizations, academic institutions and environmental enthusiasts who like to contribute towards green environment initiative. [2]

2.4 Smart Citizen

Similar to AirCasting, Smart Citizen is also an online platform, which allows user to contribute and gather environmental data. Data is captured with Smart Citizen Kit (SCK) hardware set and can be visualized through online APIs. [6]

2.5. AirBeam

AirBeam is a wearable monitoring device developed by HabitatMap that contains Humidity, Particulate Matter and Temperature sensors. The device allows Bluetooth communication service to allow smart device application to collect data detected by the sensors.

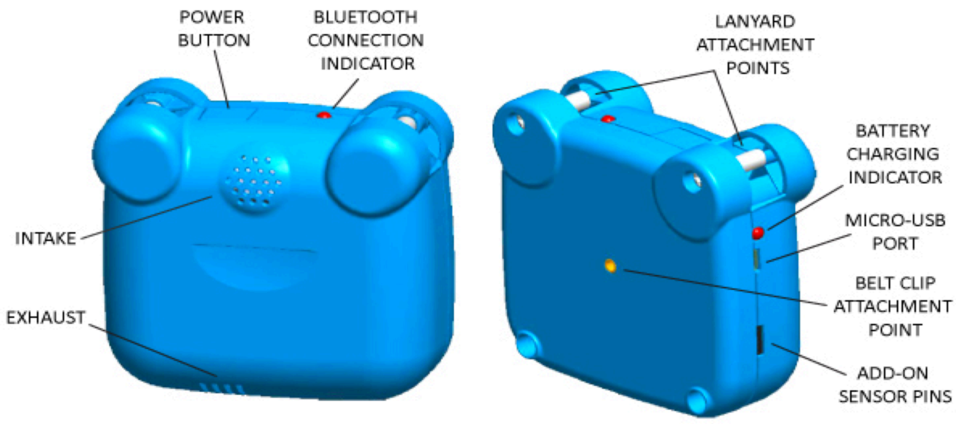


Figure 1: AirBeam Sensor Device

The device is battery powered and can be recharged via micro USB cable. It contains LED indicators for Bluetooth connection and battery charging status. It also includes additional connecting pins to add new sensors. Device hardware is Arduino Leonardo based and reprogrammable with Arduino IDE software. [2][7][9]

3.0 Design Specification

3.1. System Architecture

The AirCasting system contains four key components: sensor device, smart phone application, remote database server and web application. Environmental data is detected via the sensor device, which is recorded in the smart phone application. Each recorded session is uploaded to the remote database server. Application can also download saved sessions from the server. The server also provides a web application to visualize each session data. [2]

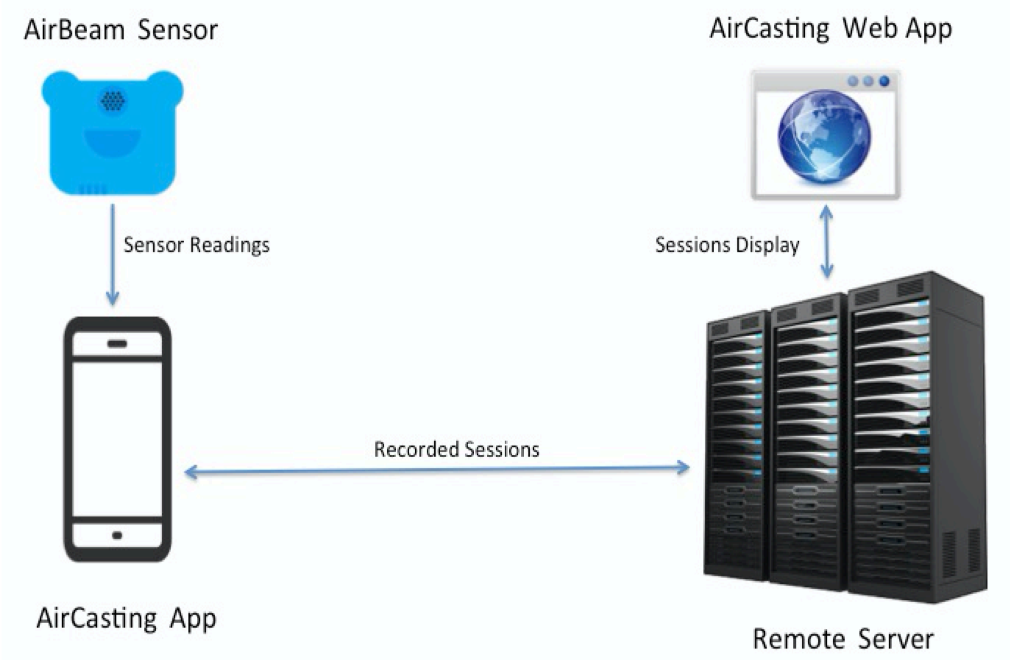


Figure 2: AirCasting System

3.2. Application Architecture

The Android AirCasting application contains three views: Dashboard, Graph and Map. The Dashboard view is loaded when the application launches. It displays all sensor data and allows commencing data record sessions. It contains navigation control to the other two views. Graph view offers the data to be represented in line chart while recording. The chart is overlayed on top a multicolour background where each coloured region represents a measurement threshold. The view also contains labels to indicate which sensor value is getting plotted. Map view visualizes the location of the data as geo-coordinates in a map layout. It also contains sensor labels that are similar to graph view to indicate the current visualized data. Both graph and map view also provides session recording control buttons. Figure – 3 represents the three views of the application. [4]

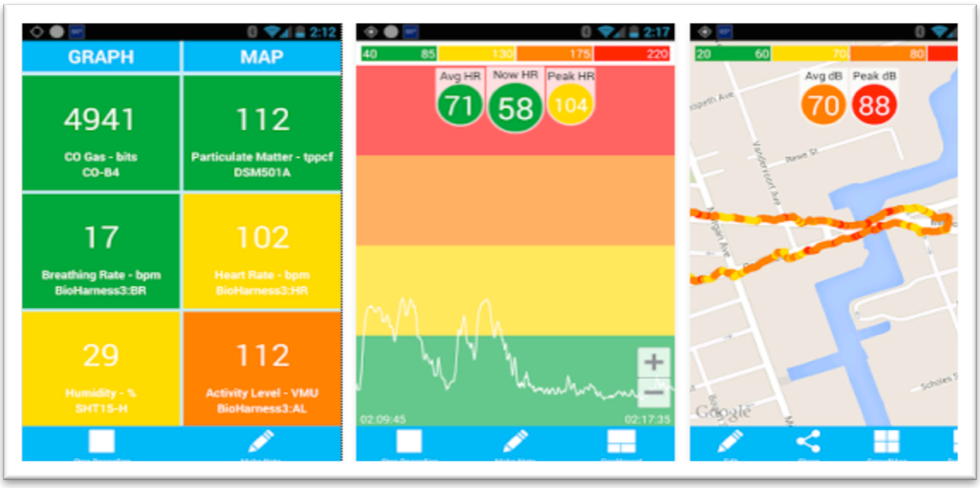


Figure 3: Android Application Views

4.0 Design Implementation

4.1. Application Views

The fundamental objective of the project has been the replication of features and visual aesthetics of the Android AirCasting application. Matching to the existing design, the application contains three main views. Each view provides a distinct layout and set of features that contributes to the overall operation of the application. The application also incorporates a set of minor features to support the main activities, which can be accessed from any of the views.

4.1.1. Dashboard

The first view is entitled Dashboard, which is loaded when the application launches. The view allows the user to see the sensor data captured by the application in real time.

As shown in Figure – 3, the view contains a navigation control bar at the top of the window, which allows the user to alternate between views of the application. Underneath the control bar, the view displays all sensor readings in a grid layout. There are four sensors: Sound

(Decibel) Level, Humidity, Particulate Matter and Temperature. Decibel readings are captured through iOS device's microphone. The other three sensor readings are captured via Bluetooth communication with AirBeam device. In the bottom part of the view, there is a recording control button to allow user to save the sensor data and clean route button to calculate a route between source and destination (refer to Renji Luke Harold's report for details). During recording, the sensor labels also display average and peak value of each sensor measurements.

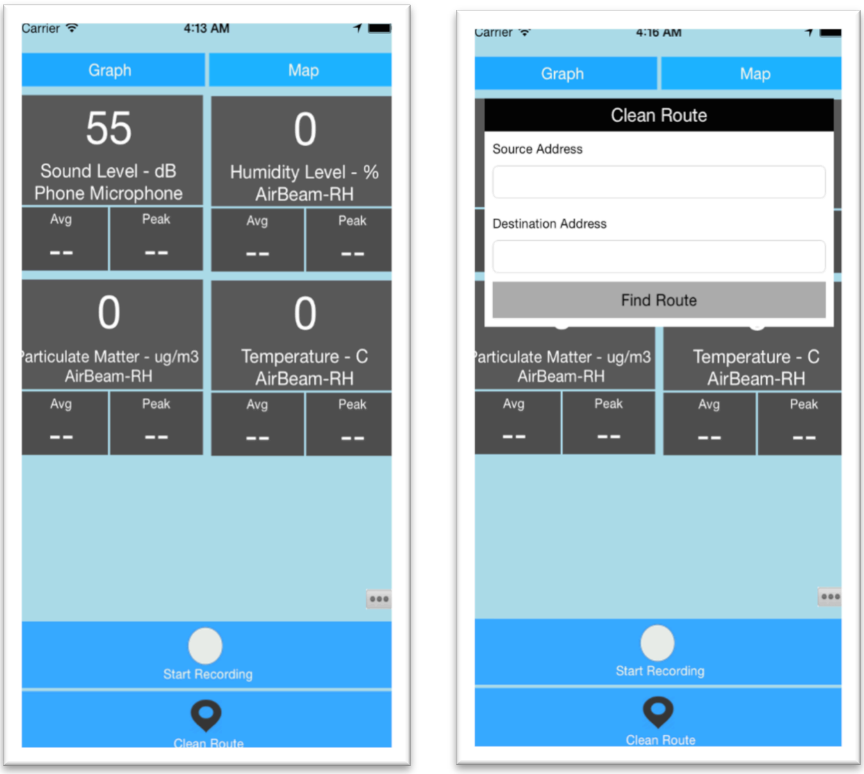
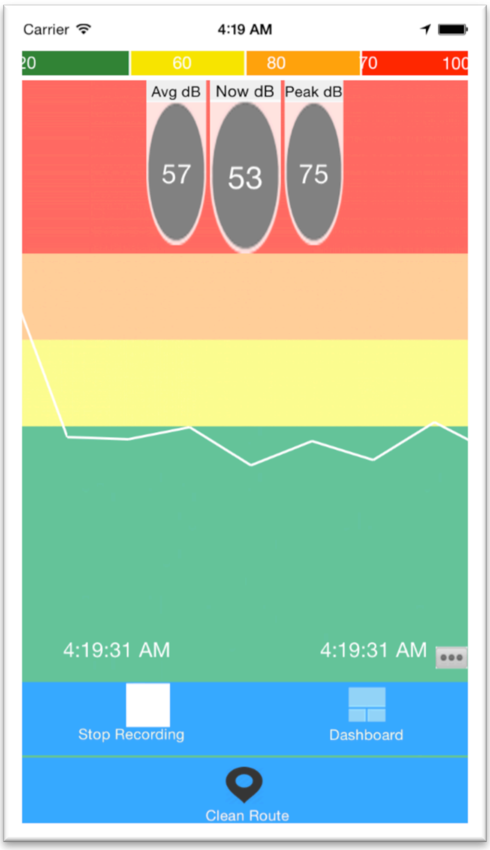


Figure 4: Dashboard View

4.1.2. Graph

The second view of the application is entitled Graph view. The view can be loaded by pressing the "Graph" button on the Dashboard's navigation control bar. This view offers the user a dynamic layout to represent the sensor data in a line chart, where different coloured regions have represented the measurement thresholds. The plotting of the chart is conducted in real time during recording of data.

As shown in Figure – 4, the view contains a touch interactive bar in the top part of the window. It allows the user to open the control menu of the measurement threshold background to define new threshold values or set default values for each coloured region. The view accommodates single sensor value to be represented at one time. Currently plotted sensor name and values are displayed within the round sensor labels. These labels are touch interactive, which opens a list of available sensors where user can select to display on the graph. There are two clock labels in the bottom part of the view displaying the current time, which provides timestamps during recording of data. Underneath the clocks, there is a control bar providing user the option to initiate recording, switch to Dashboard view and clean route search functionality.



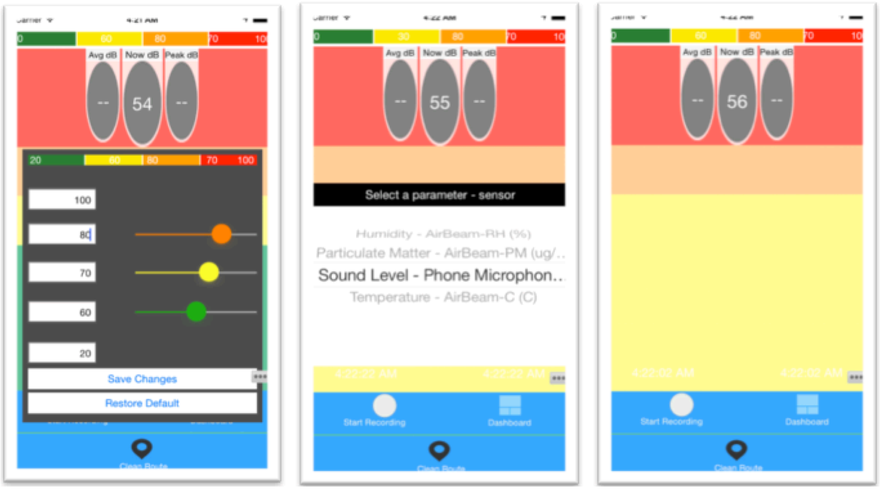


Figure 5: Graph View

4.1.3. Map

The third view of the application is entitled Map view. Similar to graph view, it can be loaded from the Dashboard view navigation control bar. This view provides user the option to visualize the location of the iOS device on a map layout. It also displays the geographic locations from where environmental data has been captured and routes.

Similar to Graph view, it contains the touch interactive bar in the top part of the window to provide the user updating option of measurement threshold background in Graph view and touch interactive round sensor labels for sensor selection as shown in Figure – 5. Each geographical point on the map is represented by annotation with coordinate information. The view also provides recording control, navigation to Dashboard and clean route search at the bottom of the window.

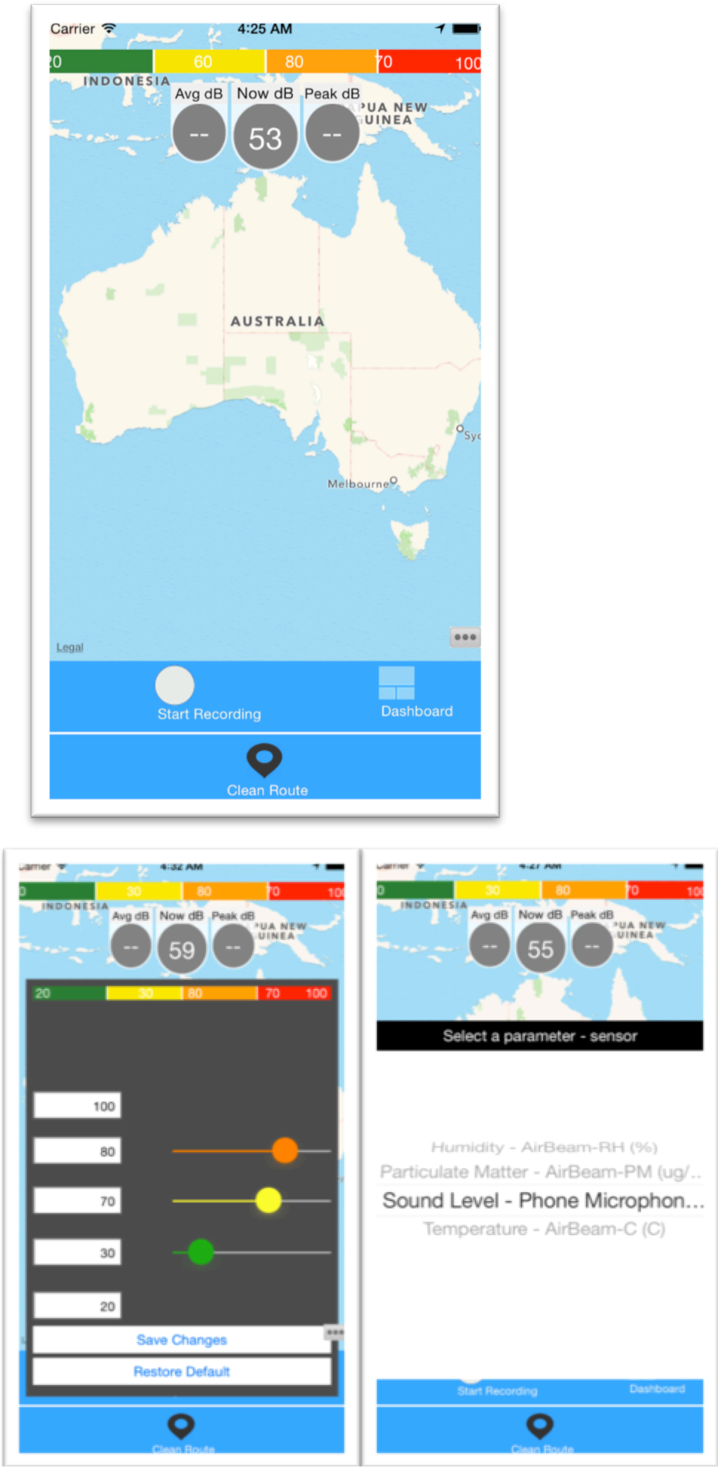


Figure 6: Map View

4.2. Sensor Connectivity

The communication with the external AirBeam device is established via Bluetooth. The application initiates a scans of nearby Bluetooth devices when launched. It searches for AirBeam device from scanned device list and connects to the first available one. Once the connection is established, the sensor data is downloaded from the device periodically.

Due to the incompatible nature of the AirBeam Bluetooth module with iOS, connectivity to the sensor device has been altered compare to the standard procedure. An intermediate device has been implemented as a Base station which interfaces with the AirBeam device via USB. The base station provides the Bluetooth service for the application to communicate.
The base station is a Raspberry Pi 2 system with Bluetooth Low Energy (BLE) module. The device implements Raspbian OS and Bleno, which is a Node.js module, to provide Bluetooth service. Implementation details can be found in the Challenges and Solution section of the report.

4.3. Other Features

The application contains several supporting features, which can be accessed from any view by pressing the "more" button in the bottom right corner of the view. It displays two options: Sessions and Settings.

The Sessions feature provides access to all saved data sessions in the remote and local database of a user. These sessions are downloaded and listed in two categories (Synced and Saved) when application switches to the sessions view. Each session label of both lists is touch interactive which allows the user to visualize the session's geographical location in the Map view as route. Figure – 6 represents the sessions list view and a session being displayed on the Map view.

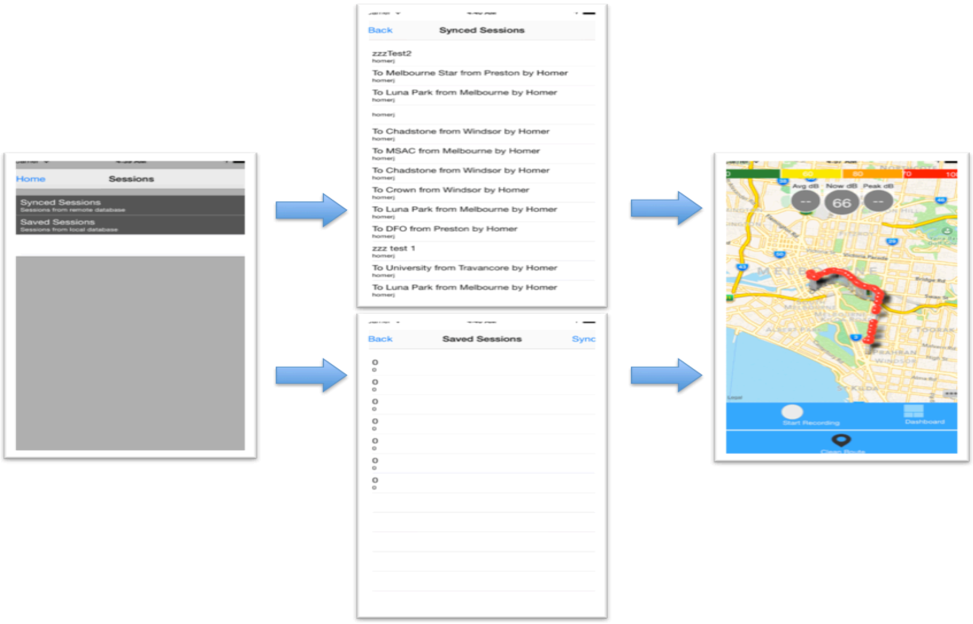


Figure 7: Session in Map View

The Settings view of the application lists out some supporting features and properties, which can be set by user. Figure – 7 display the settings view of the application. Currently, the application provides user profile management, external device connection, data upload method, map view style selection and backend settings configuration features.

Figure 8: Settings Features

The external device settings allow the user to initiate Bluetooth device scan and connect to the specified device name. Due to time constraints of the project and sensor device incompatibility, the external device list feature has been configured with a fixed device name.

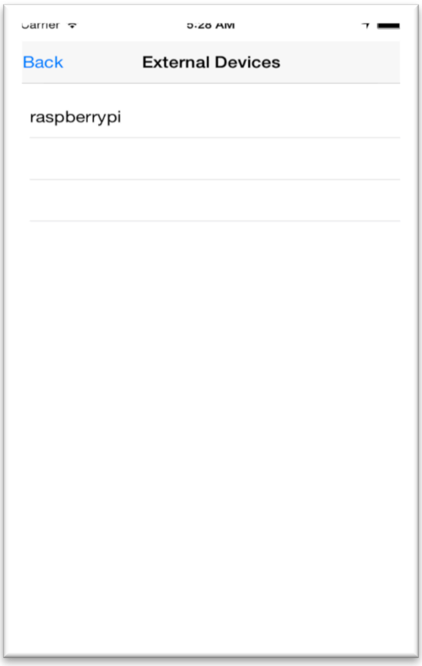


Figure 9: External Device List

The profile feature permits user to log in to an existing user profile with valid username and password. It also provides an option to create new user profiles. User sign-in is required to record data and view existing sessions within the application. Figure – 8 displays instances of existing user sign-in and new user creation.

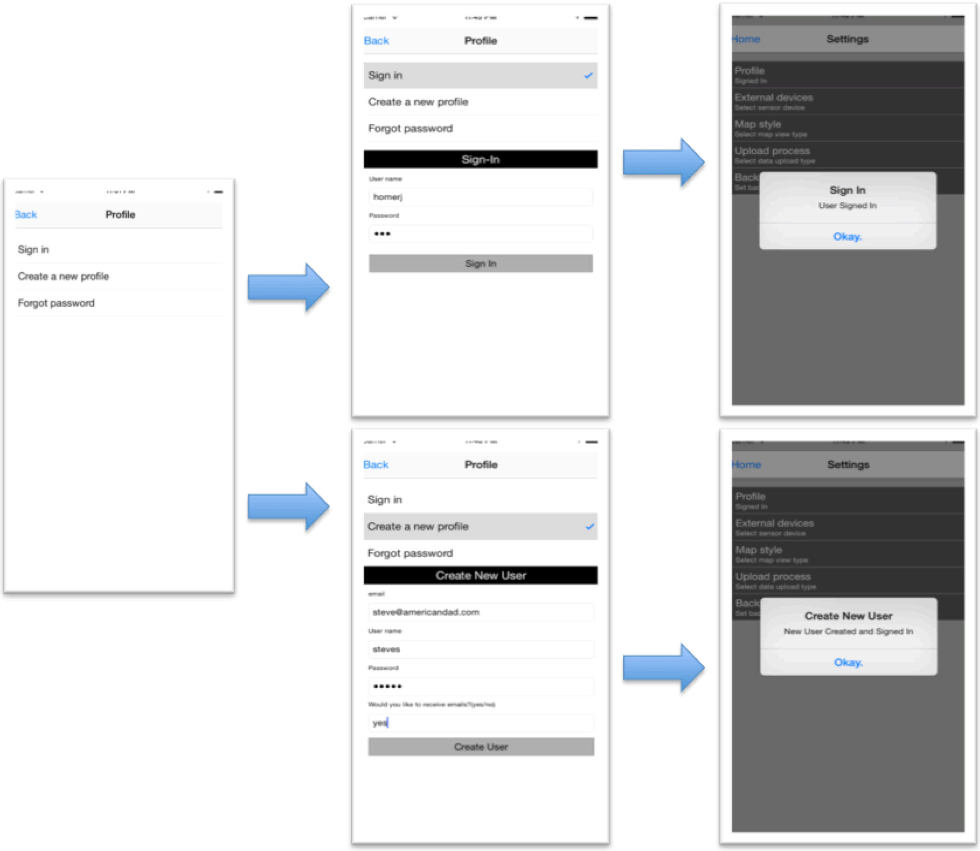
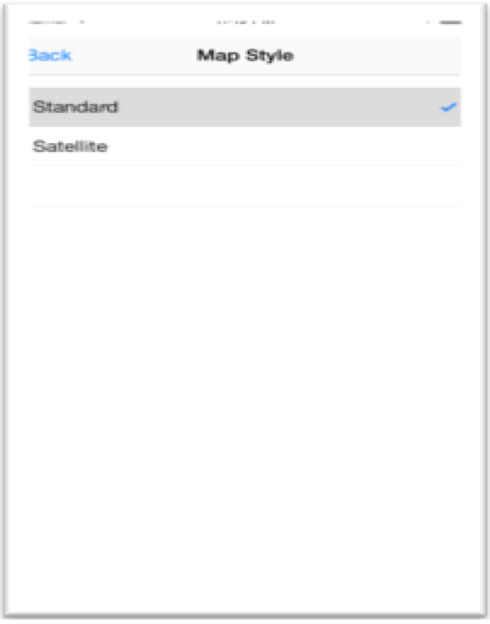


Figure 10: Profile Feature

The application supports two style of map view: Standard and Satellite. Preferred style can be set via Map Style property from the settings view. Application also provides two types of data upload process to the remote server: Record and Stream. Record process permits user to save data in the local database of the iOS device and upload after recording finishes. On the other hand, Stream process allows the user to upload the data to the remote server in real time while capturing from the environment. Figure – 9 represent the view of Map Style and Upload Process.



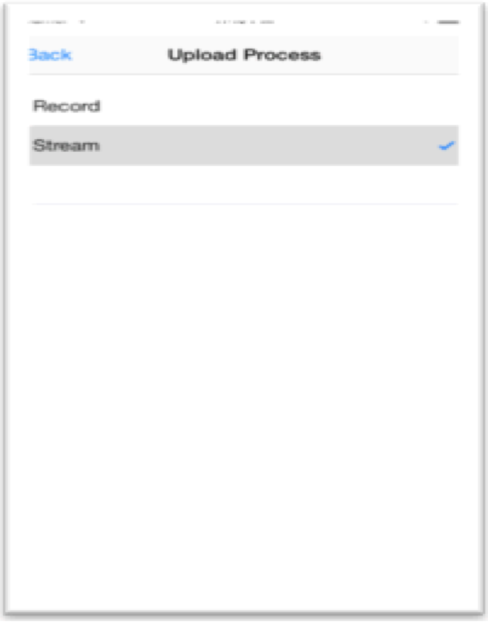


Figure 11: Map Style and Upload Process

For further details about the upload process, refer to Renji Luke Harold's report of the project.

The final feature of the settings view is the Backend Settings. It displays the current remote database server address and lets the user change addresses and port number if required.

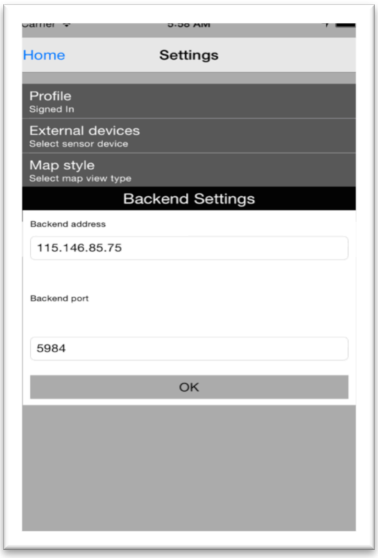


Figure 12: Backend Settings

4.4 Data Storage

The application's data model implements two different databases: Local SQLite and Remote CouchDB. The SQLite database acts as a temporary storage facility during Record upload process while the remote CouchDB database provides storage for Stream upload process. The remote database server has been configured in NeCTAR cloud. For further details about the data storage and usage procedure, refer to Renji Luke Harold's report of the project.

4.0 Challenges and Solutions

5.1. AirBeam Compatibility

One of the key challenges of the project has been the compatibility of the AirBeam device with iOS. Apple supports two types of Bluetooth modules for iOS: Classic and BLE. Classic Bluetooth module communication can be established with iOS External Accessories framework but it requires a registered set of communication protocol with Apple MFi licensing program. The initial research, testing and information provided by the manufacturer revealed that the AirBeam device has been designed with a classic Bluetooth module, which has no registered communication protocol with Apple. As a result, iOS application cannot communicate with the device directly without changing the AirBeam hardware and firmware. Within the project timeframe, a new replacement device was not been considered a viable solution.

Further investigation of the AirBeam hardware indicated that the device has been developed using a hardware platform (Arduino Leonardo) that provides serial USB communication to its I/O pins. As a result, the sensor data can be read via micro USB interface of the device. This provided an opportunity to implement a provisional solution using Raspberry Pi 2 system and develop the application with all key features that are available in the Android version. Figure – 10 represents the communication model with Raspberry Pi as base device.

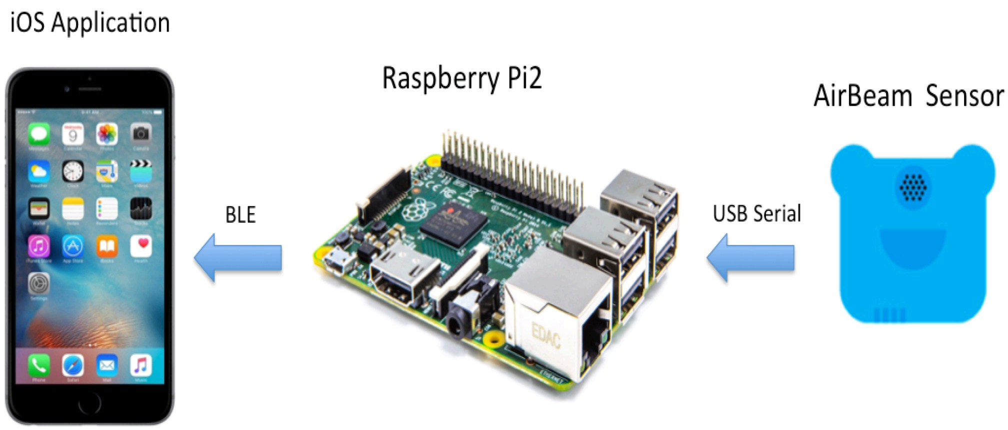


Figure 13: AirBeam with Raspberry Pi

The Raspberry Pi system has been installed with a Bluetooth 4.0 USB module (BLE), which is iOS compatible with CoreBluetooth framework. To create communication service, Bleno Node.js module has been implemented. The AirBeam device connects to the Raspberry Pi system via USB port where sensor data can be read periodically. Captured data from the sensor device is then advertised through a notification service of Bluetooth communication. The service script written in JavaScript advertises the Raspberry Pi system as peripheral which can be connected to the iOS application for sensor data retrieval.

As this solution is provisional, Bluetooth communication segment of the software has been developed in a generic approach to support a replacement sensor device with minimal changes of the source code.

5.0 Future Directions

5.1. Updated Sensor Device

One of the key findings of the project is the compatibly of AirBeam device with different OS platform. To be able to communicate seamlessly among iOS and Android, a new sensor device with Bluetooth Low Energy module is recommended.

5.2. Upgrade to Swift 2.0

The application has been developed with Swift version 1.2. Apple has released Swift version 2.0 recently, which contains significant syntax changes compare to version 1.2. Swift 2.0 provides better error handling model with "Try/Catch and Throw" mechanism. The updated language syntax is more expressive and implements more features of Objective C. One of the future considerations can be to upgrade the application from version 1.2 to version 2.0 to enhance the application source code and take advantage of the new features.

6.0 Conclusion

In this report, iOS version of AirCasting application has been presented. The project has been targeted for replication of Android version of the application. The project has been planned with two scopes. First scope of the project refers to the development of User Interface of the application and key functionalities of the android equivalent. As an additional feature, real time data upload/streaming has been implemented within the application. Within the second scope of the project, a routing process is under development, which calculates a route between source and destination address based on pollution level of the area.

In conclusion of the project, key user interface features and functionalities have been completed within the project timeframe. Due to resource and time constraints, the clean route feature development scope has been revised where modified version of A* algorithm has been implemented (Refer to Renji Luke Harold's report).

7.0 Online Repository

The project source code is available through Github, which can be found on through the following link:

https://github.com/MIT-FYP/AirCasting-iOS.git

There is also a video available on YouTube, which describes the application and its usage:

https://www.youtube.com/watch?v=06W8Ast7rGY&feature=youtu.be

8.0 References

[1] ‘*About AURIN*’, Australian Urban Research Infrastructure Network, viewed 8th August 2015, url <http://aurin.org.au/about/>

[2] ‘*AirCasting*’, Habitat Map, viewed 5th August 2015, url <http://aircasting.org/>

[3] ‘*AirBeam: Share & Improve Your Air*, Habitat Map, viewed 6th August 2015, url <https://www.kickstarter.com/projects/741031201/airbeam-share-and-improve-your-air>

[4] ‘*AirCasting Application*’, Google Play, viewed 2nd August 2015, url <https://play.google.com/store/apps/details?id=pl.llp.aircasting&hl=en>

[5] http://www.comscore.com/Insights/Market-Rankings/comScore-Reports-May-2015-US-Smartphone-Subscriber-Market-Share

[6] Smartcitizen.me,. 'Smart Citizen : Citizen Science Platform For Participatory Processes Of The People In The Cities'. N.p., 2015. Web. 3 Nov. 2015, url <https://smartcitizen.me/>.

[7] Besser, Alex. *Evaluation Of A Low-Cost Sampler For Assessing Personal Fine Particulate Matter Exposures*. 1st ed. New York: New York University, 2015

[8] The CAUL Hub,. 'The CAUL Hub'. N.p., 2015. Web. 3 Nov. 2015, url <http://urbannesp.com/>.

[9] Takingspace.org,. 'Airbeam Technical Specifications, Operation & Performance : Taking Space'. N.p., 2015. Web. 3 Nov. 2015, url <http://www.takingspace.org/airbeam-technical-specifications-operation-performance/>.