Design Document

Computer Science Department

Texas Christian University

Date: May 5, 2014

Authors: Stockton Ackermann, Nicholas Capurso, Eric Elsken, Myrella Garcia, Casey Stephens, and David Woodworth.

Revision History

The following is a history of document revisions.

Version	Changes	Edited
Version 1.0	Initial Draft	November 21, 2013
Version 1.1	Numerous consistency fixes Section 1: • Edited Purpose subsection • Changed “Overview” to “Section Overview” Section 2: • Reworded some general constraints • Changed “Developmental Methods” to “Development Environment” Section 3: • Revised/Added “Major Component Descriptions” subsection Section 4: • Updated diagrams Section 5: • Updated UML models	December 05, 2013
Version 2.0	Section 3: • System Architecture Diagram update All Sections: • Changed “collision” to “accident”	January 30, 2014
Version 3.0	Section 2: • Programming Environment update Section 3: • System Architecture Diagram update • Changed “wireless” to “Bluetooth” All Sections: • Changed “smart phone” to “smartphone” • Changed “Sensor Tag” to “SensorTag”	April 10, 2014
Version 4.0	Table of Contents: • Updated page numbers to be consistent Section 6: • Updated prototypes	May 5, 2014

Revision Sign-off

By signing the following, the team member asserts that he or she has read the entire document and has, to the best of knowledge, found the information contained herein to be accurate, relevant, and free of typographical error.

Name	Signature	Date
Stockton Ackermann s.ackermann@tcu.edu		
Nicholas Capurso nick.capurso@tcu.edu		
Eric Elsken eric.elsken@tcu.edu		
Myrella Garcia m.garciaurena@tcu.edu		
Casey Stephens casey.stephens@tcu.edu		
David Woodworth d.woodworth@tcu.edu		

Table of Contents

1. Introduction

1.1. Purpose

This document provides a complete description of the FrogStar system design. Included in this document are the design constraints, system architecture and major component descriptions, storage formats, Unified Modeling Language (UML) diagrams (state, class, and sequence), and the user interface prototypes.

1.2. Project Background

In the event of a vehicular accident, there are many scenarios in which the occupants become incapacitated and unable to call for assistance. There exist systems, e.g., OnStar, that currently provide accident detection and roadside assistance services. However, the cost of these proprietary systems and their availability for all vehicular models limit their use.

Recently, Texas Instruments released an inexpensive, Bluetooth-enabled SensorTag that features an on-board accelerometer and a gyroscope. These sensors can be used to detect a high-speed accident when used in tandem.

Project FrogStar employs the use of TI SensorTags with two controlling devices, a single board computer as an on-board control unit (OBCU) and a smartphone, to explore the possibility of detecting a serious accident. The controlling devices serve as redundant checks when an accident is detected. In the event of an emergency, the smartphone has the capability to contact emergency responders via cellular networks. The use of NFC tags provides a convenient way of enabling or disabling the system.

1.3. Section Overview

Section 2 – Design Constraints: Describes assumptions and dependencies, general constraints, and developmental methods for the FrogStar system.

Section 3 – System Architecture: Displays a model of the architecture and major component descriptions.

Section 4 – File Design: Describes how data from sensors is stored on the smartphone and the on-board control unit (OBCU).

Section 5 – UML Models: Displays the state, class and sequence diagrams.

Section 6 – User Interface: Gives screenshots from the prototype of the smartphone application.

Section 7 – Definition of Terms: Defines technical and project-specific terms used in this document.

2. Design Constraints

2.1. Assumptions and Dependencies

The FrogStar system will assume the following:

- TI sensors and smartphone are assumed to have adequate battery life.
- OBCU is plugged in and turned on by the vehicle.
- NFC tag must be placed in the vehicle.
    - User will toggle system on and off with the NFC tag.
- Smartphone must have cellular service in order to contact emergency responders.
    - Wi-Fi may also be used to contact responders, but similarly, the system must be connected to a wireless AP.
- Devices must be within Bluetooth communication range of each other.
- Smartphone must be equipped with Bluetooth, NFC, and Wi-Fi capabilities as well as the necessary sensors.

2.2. General Constraints

- Sampling Rate Constraints:
    - The SensorTags must be queried at or above the minimum possible sampling interval such that accidents can be accurately detected.
- Data Storage Constraints:
    - The amount of storage used on the OBCU and the smartphone must not exceed the maximum storage capacity.
    - The amount of data that needs to be stored on an NFC tag must not exceed the tag's maximum storage capacity.
- Application Interface (API) Constraints:
    - User's smartphone must be running at or above Android 2.0 for Bluetooth capability.
- Communication Constraints:
    - System data transfer is limited by the speed of the Bluetooth communications.

2.3. Development Environment

Programming Environment

- Android Development Toolkit Plugin (2.2) for Eclipse.
- BlueZ 5.2
- Eclipse: Kepler (4.3.1) Service Release 1
- GCC 4.8.2
- Raspbian Linux Distribution
- Windows 7 64-bit SP 1

File Transfer and Version Control

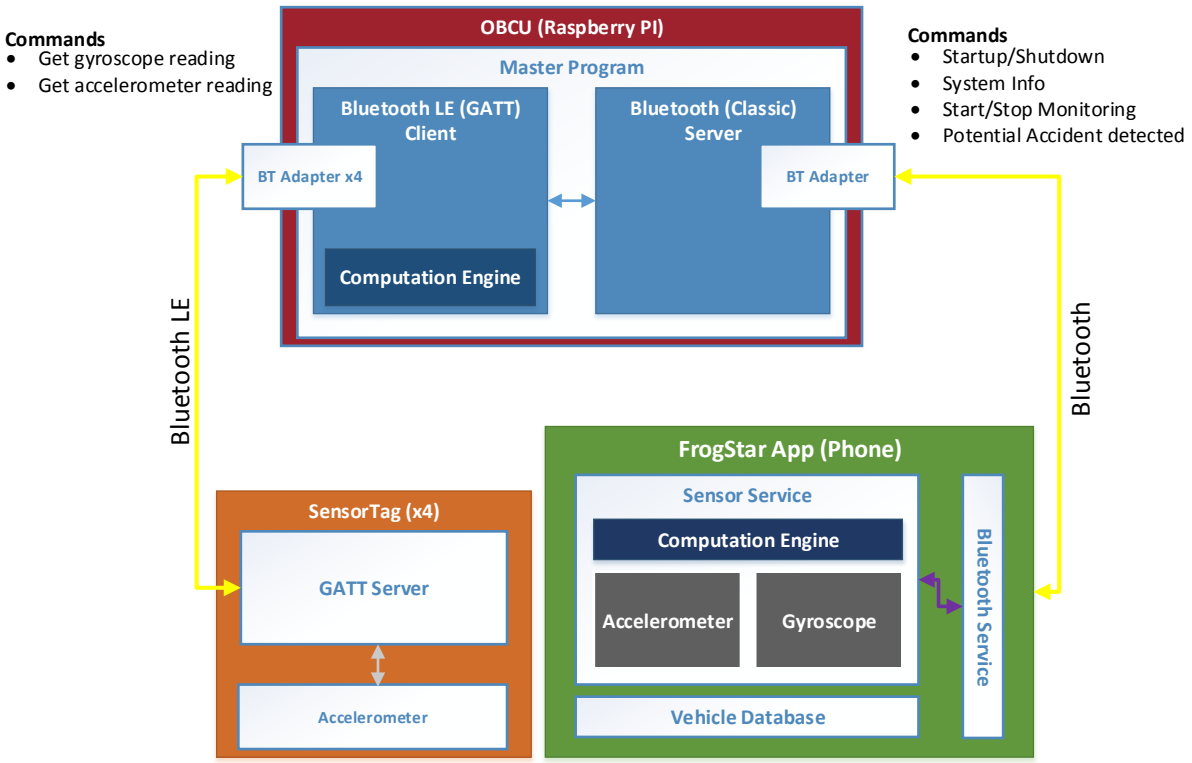
- CoreFTP Lite 2.2
- Tortoise SVN 1.7.10
- Windows 2008 Server running Subversion & IIS

Productivity Software

- Adobe Photoshop CS6
- SCR Screen Recorder
- Microsoft Visio 2013
- Microsoft Word 2010
- Microsoft PowerPoint 2010
- Notepad++ 6.5.1

3. System Architecture

3.1. System Architecture Diagram



3.2. Major Component Descriptions

3.1.1. Smartphone

The primary role of the smartphone is to query its own sensors and perform accident detection. Upon deciding that an accident has occurred, the smartphone will confirm readings with the on-board control unit (OBCU) over a Bluetooth connection. Sensor readings will also be stored locally on the smartphone and backed up in the event of an accident.

If an accident is indeed confirmed and the user requires assistance, the smartphone can alert emergency responders via cellular network.

The smartphone also handles startup and shutdown of the system via NFC communication.

In addition to its primary roles, an application running on the smartphone will allow the user to manage user and vehicle profiles. This data will be stored locally on the phone.

3.1.2. On-Board Control Unit (OBCU)

The primary role of the OBCU is to manage the querying of the various SensorTags over a Bluetooth LE connection and also perform accident detection similar to the smartphone. Upon deciding that an accident has occurred, the OBCU will confirm readings with the smartphone over a Bluetooth connection. Sensor readings will also be stored locally on the OBCU and backed up in the event of an accident. Finally, the OBCU will monitor the current health status of the sensors, network strength and will alert the smartphone if attention is necessary.

3.1.3. TI CC2541 SensorTags

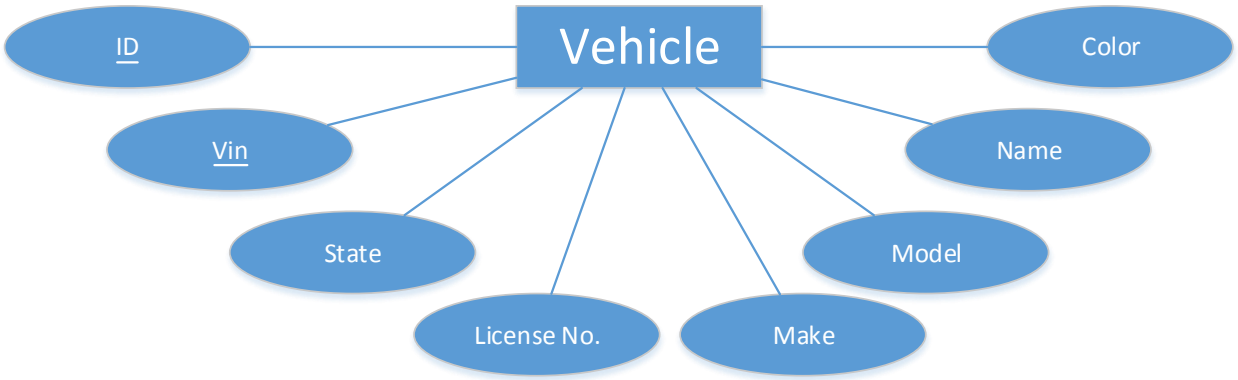
The TI SensorTags are each equipped with an accelerometer and gyroscope. Communication can occur through a Bluetooth LE connection. Four SensorTags will be utilized per vehicle in order to have redundancy.

4. File Design

4.1. Smartphone Application

The user profile will be stored in a preferences text file, whereas the vehicle profiles will be stored locally in a database. Sensor readings from the smartphone in the last minute will be stored.

4.1.1. Vehicle Database Entity-Relationship Model



4.1.2. Vehicle Database Schema

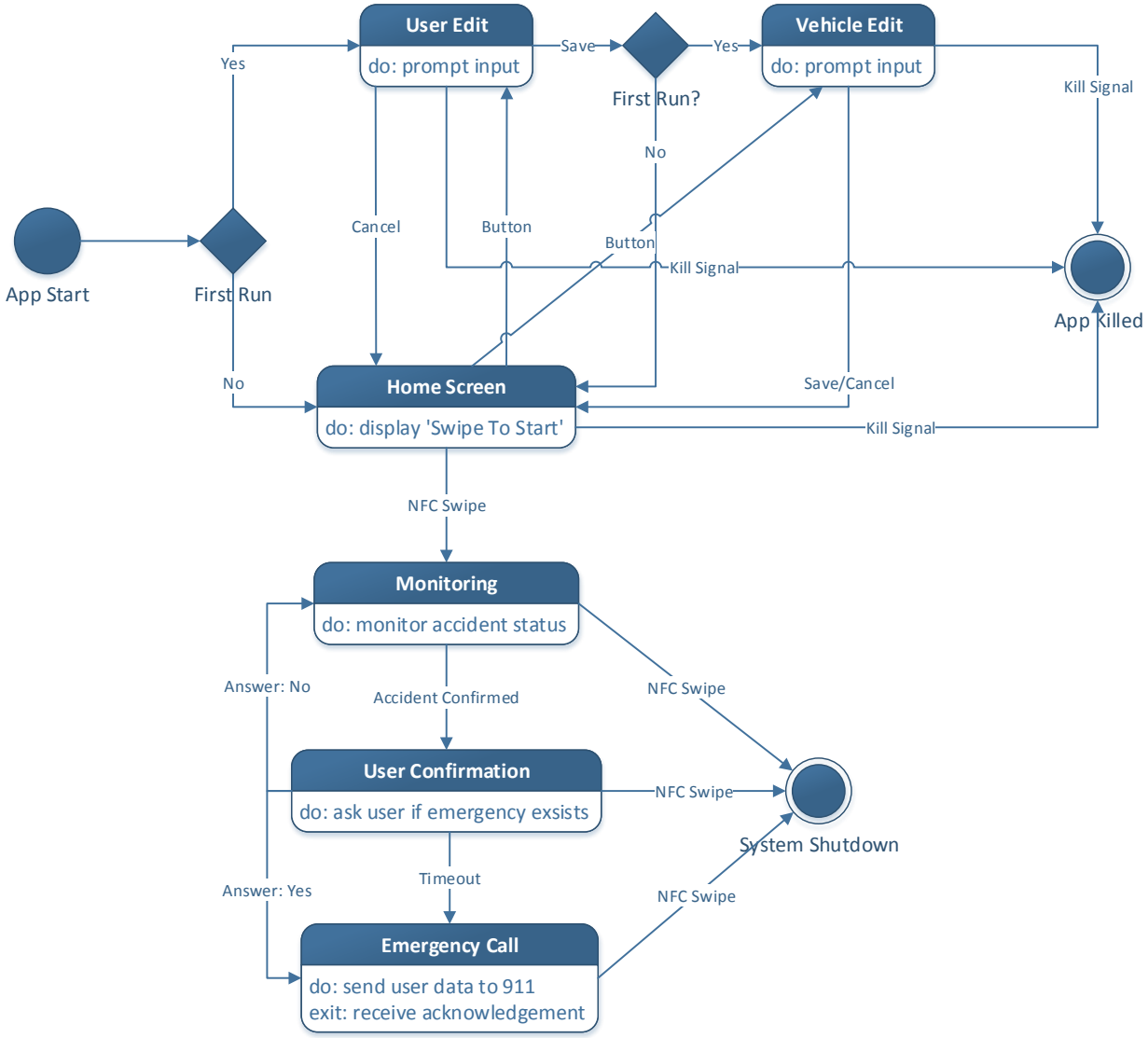
Vehicle

ID	VIN	State	License No.	Make	Model	Name	Color

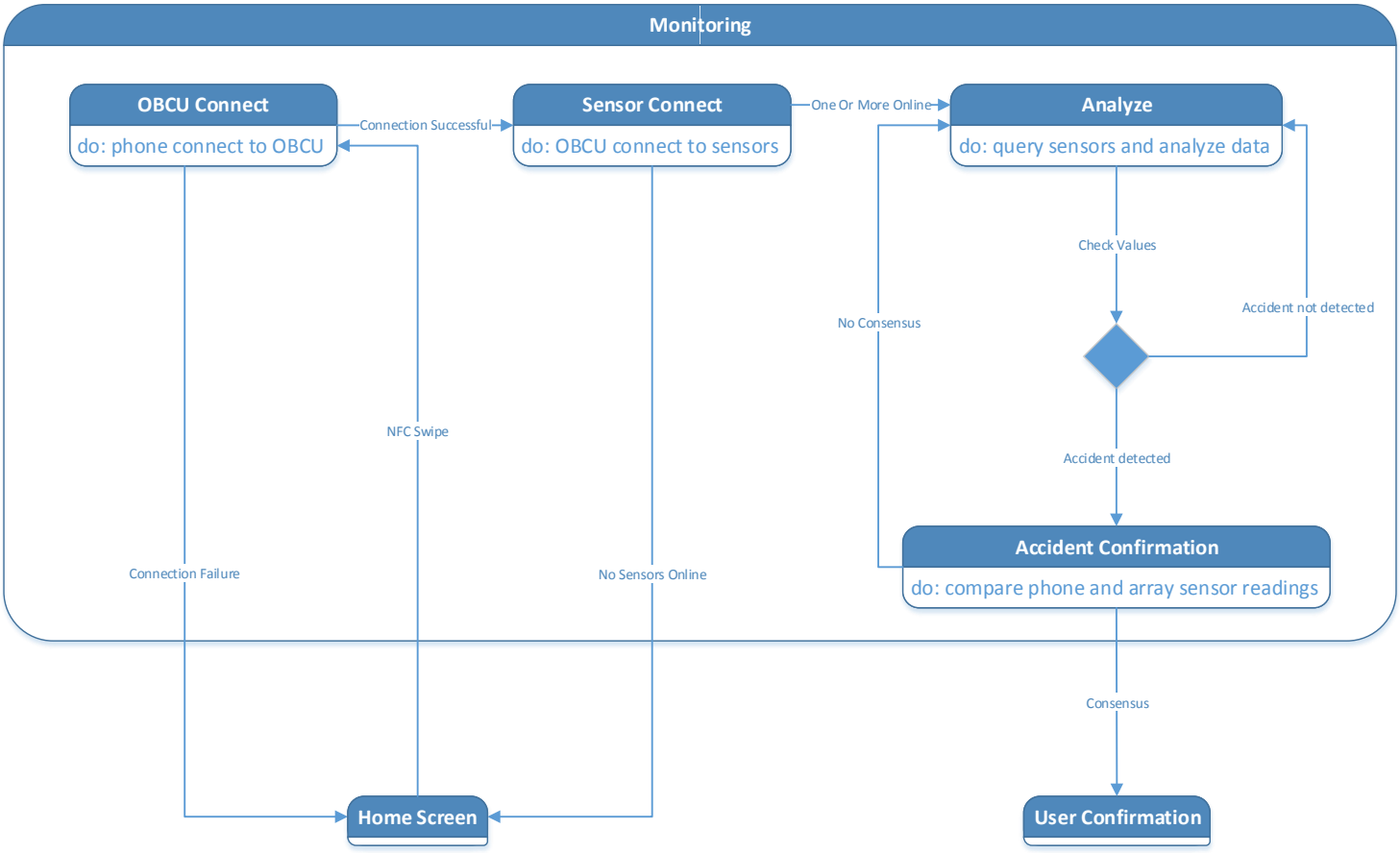
5. UML Models

5.1. State Diagrams

5.1.1. Overall UML State Diagram



5.1.2. Monitoring State



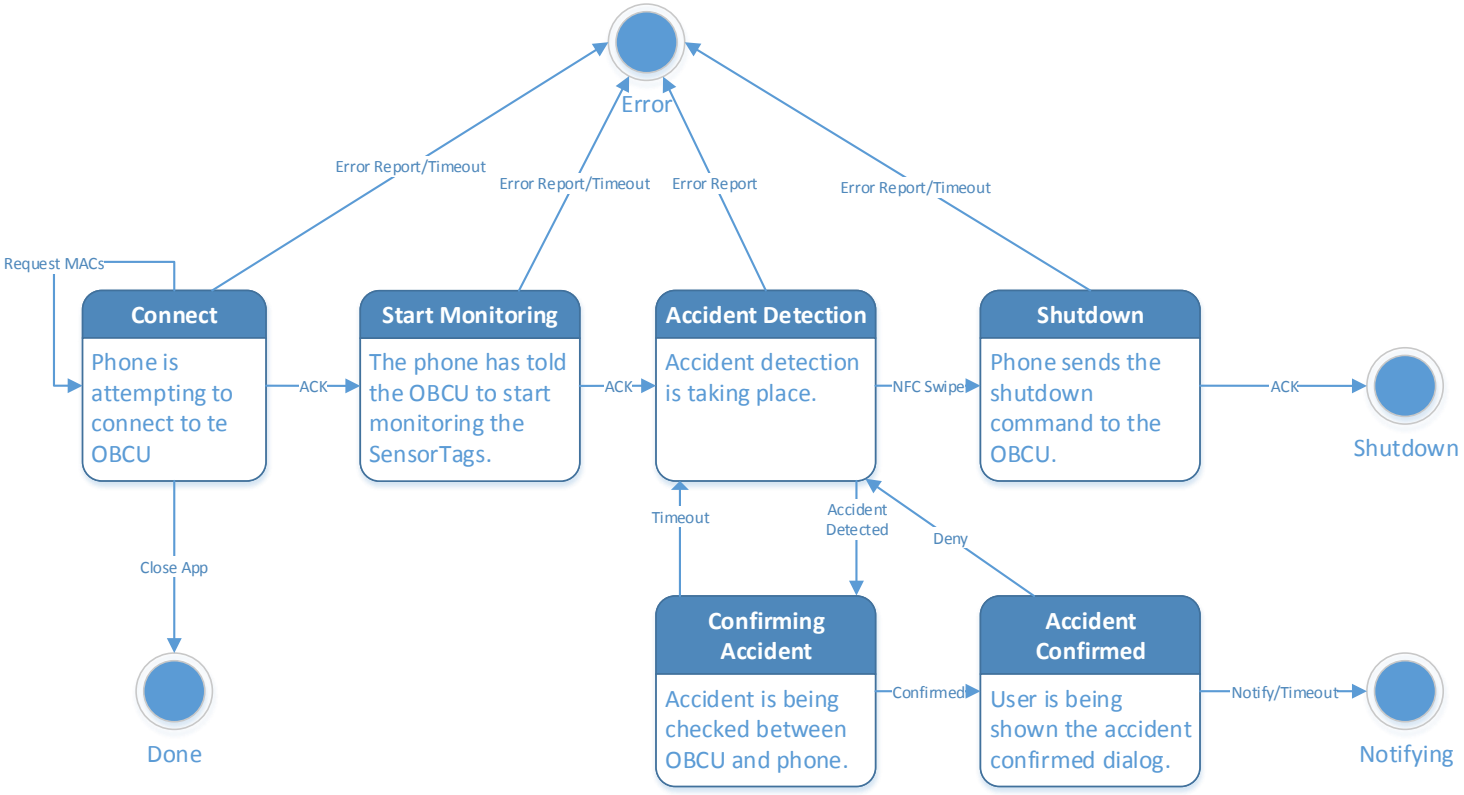
5.1.3. Overall State Diagram Table

State	Description
User Edit	The input fields to edit the user profile are displayed along with the Save and Cancel buttons. If the user has reached this state as part of the application's first run, the Cancel button will be disabled.
Vehicle Edit	The input fields to edit the vehicle profiles are displayed along with the Save and Cancel buttons. If the user has reached this state as part of the application's first run, the Cancel button will be disabled.
Home Screen	The default start screen for a normal run. The display indicates that the phone is ready to be swiped to the NFC tag.
Monitoring	Both the OBCU and the phone are checking sensor readings and comparing possible crash events.
User Confirmation	The phone displays a message asking if the user is okay, along with a countdown timer. The user can click Cancel and the emergency services will not be notified. In the event the user is incapacitated, the timer will run for 30 seconds and if no response is given emergency services are automatically notified.
Emergency Call	The phone dials 911 and relays pertinent user information to emergency services. Additionally, the emergency contact in the database will also be notified.
Stimulus	**Description**
First Run – Yes	This is the first application start (i.e. no previous user/vehicle data has been recorded).
First Run – No	This is not the first application start (i.e. previous user/vehicle data has been recorded).
Button	The user has pressed the button to change screens.
Save	The user has pressed the save button.
Cancel	The user has pressed the cancel button (Note: On first run, the cancel button will not be shown when in the User Edit or Vehicle Edit state).
Kill Signal	The user has killed/quit the application.
NFC Swipe	The user has swiped the NFC tag with the smartphone causing the system to either start or shutdown.
Possible Accident Detected	An accident has been detected based on sensor readings.
Timeout	The user failed to respond to the dialog popup in a predetermined length of time.
Answer: No	The user answers "No, there is no emergency."
Answer: Yes	The user answers "Yes, there is an emergency."

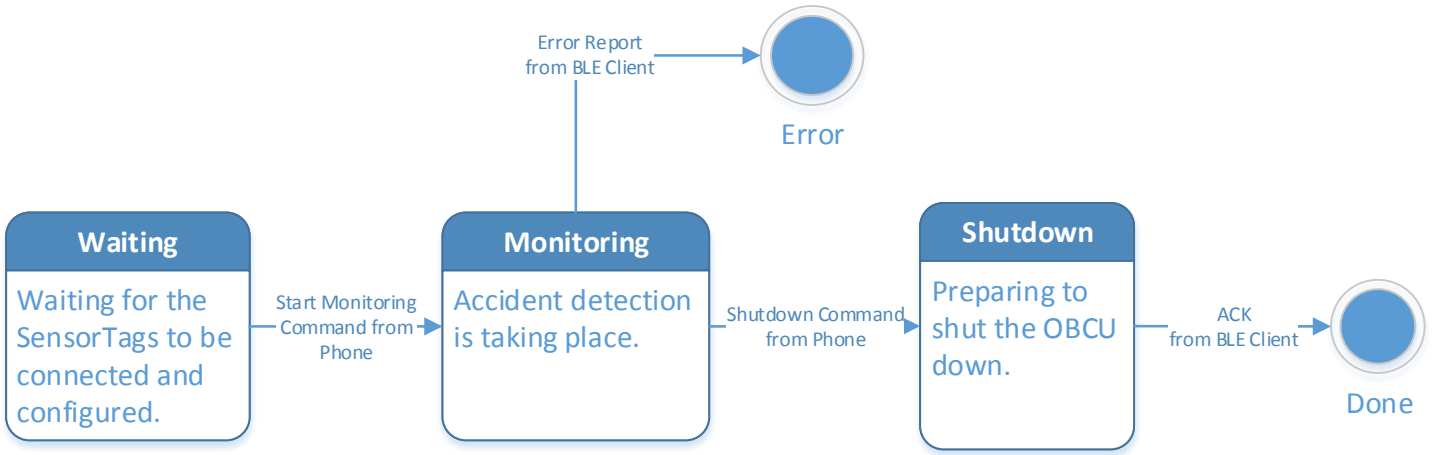
5.1.4. Monitoring State Diagram Table

State	Description
OBCU Connect	The phone is attempting to connect to the OBCU
Sensor Connect	The OBCU is attempting to connect to the sensor array
Analyze	The phone and the OBCU are polling and analyzing data from their sensors.
Crash Confirmation	Either the phone and/or the OBCU have detected an accident and are correlating with each other to confirm or dismiss the event.
Stimulus	**Description**
NFC Swipe	The user has swiped the NFC tag to initiate system start.
Connection Successful	The smartphone has successfully connected to the OBCU.
Connection Failure	The smartphone has failed to connect to the OBCU.
No Sensors Online	No sensors tags can be detected or have all malfunctioned.
One or More (Sensors) Online	At least one or more SensorTags are functioning.
Check Values	Sensor readings are analyzed to detect an accident.
Accident Not Detected	An accident has not been detected.
Accident Detected	An accident has been detected.
No Consensus	The smartphone and the OBCU do not agree that an accident has occurred.
Consensus	The smartphone and the OBCU agree that an accident has occurred.

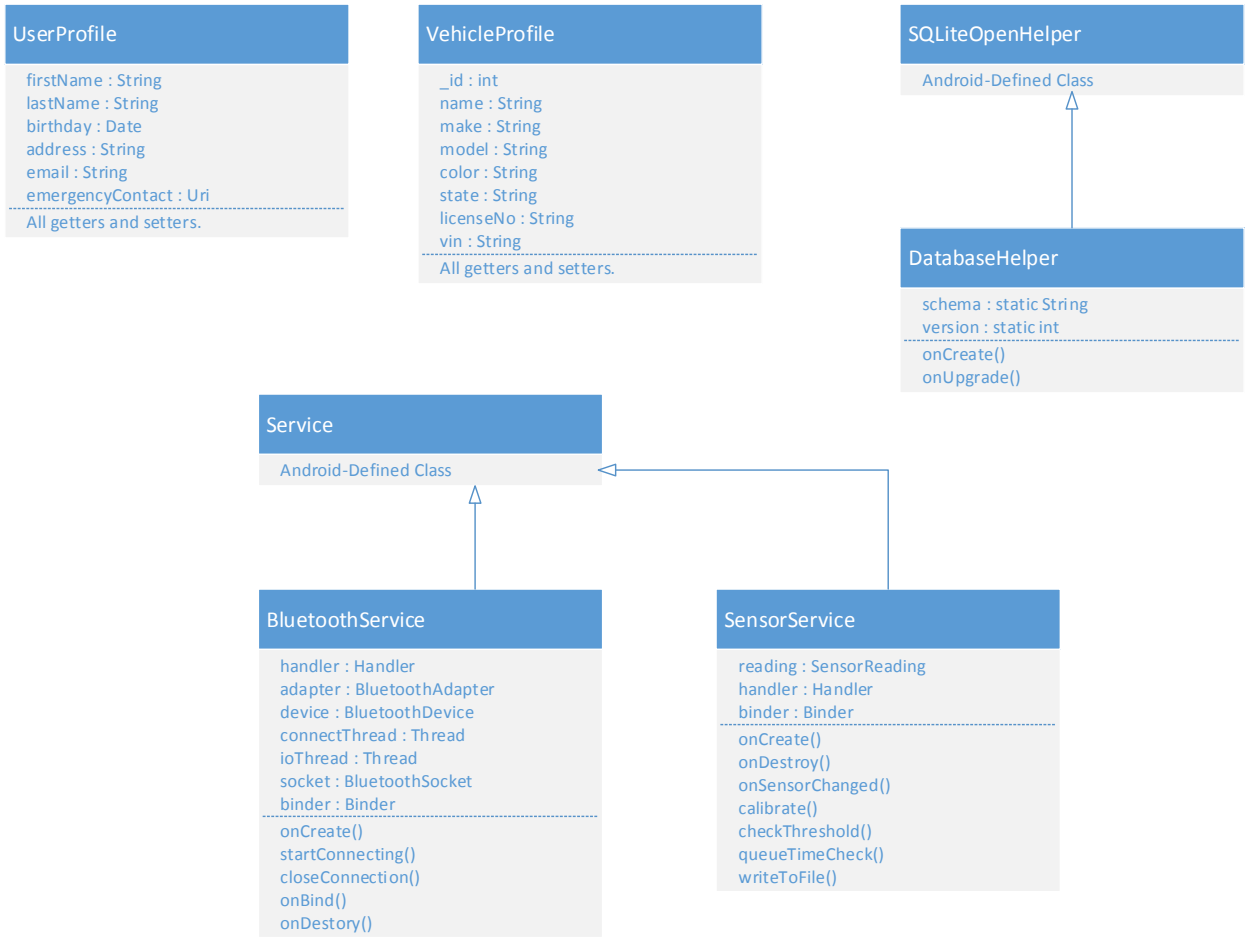
5.1.5. Bluetooth Service State Diagram



5.1.6. Bluetooth Server State Diagram

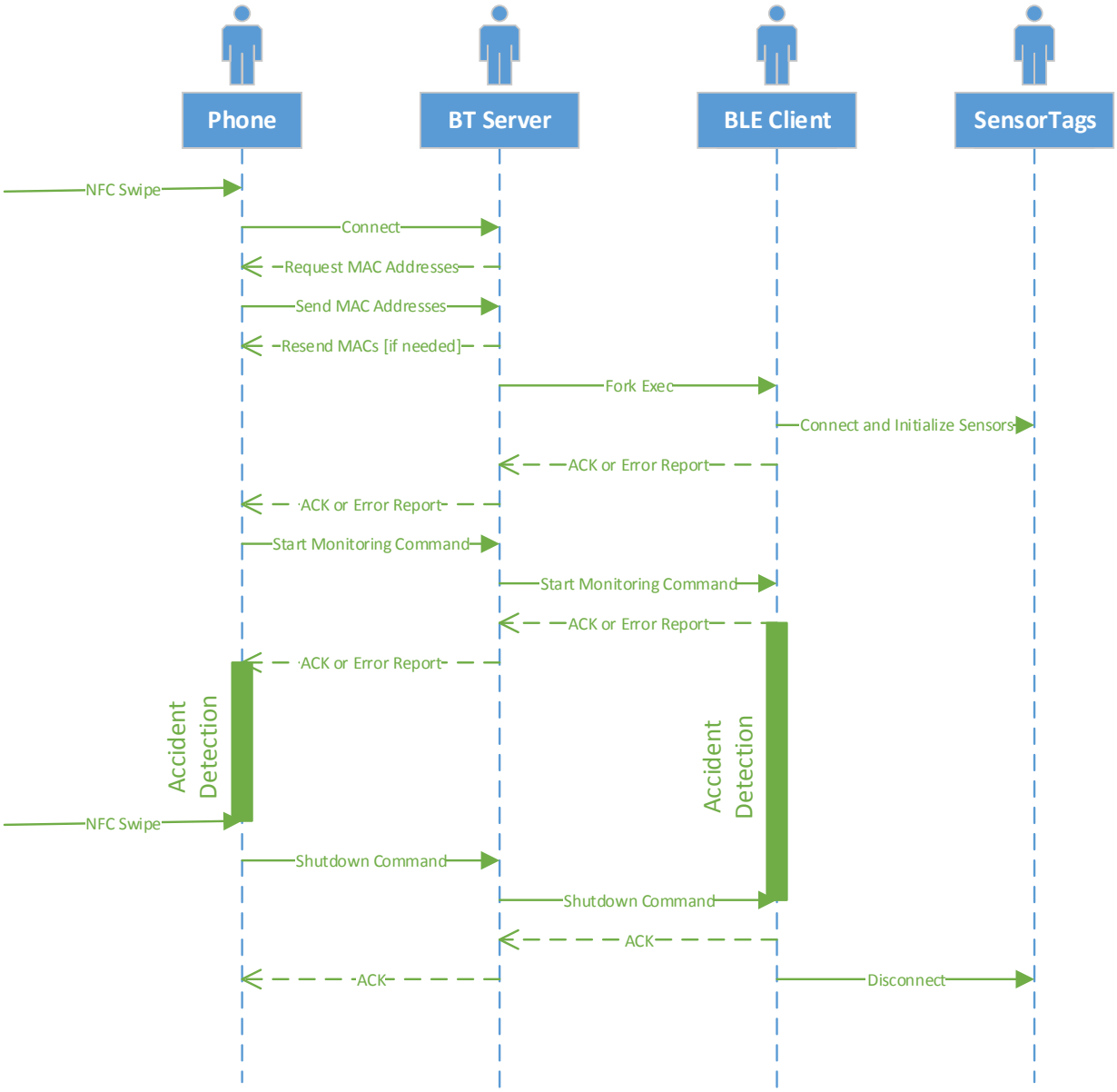


5.2. Class Diagram

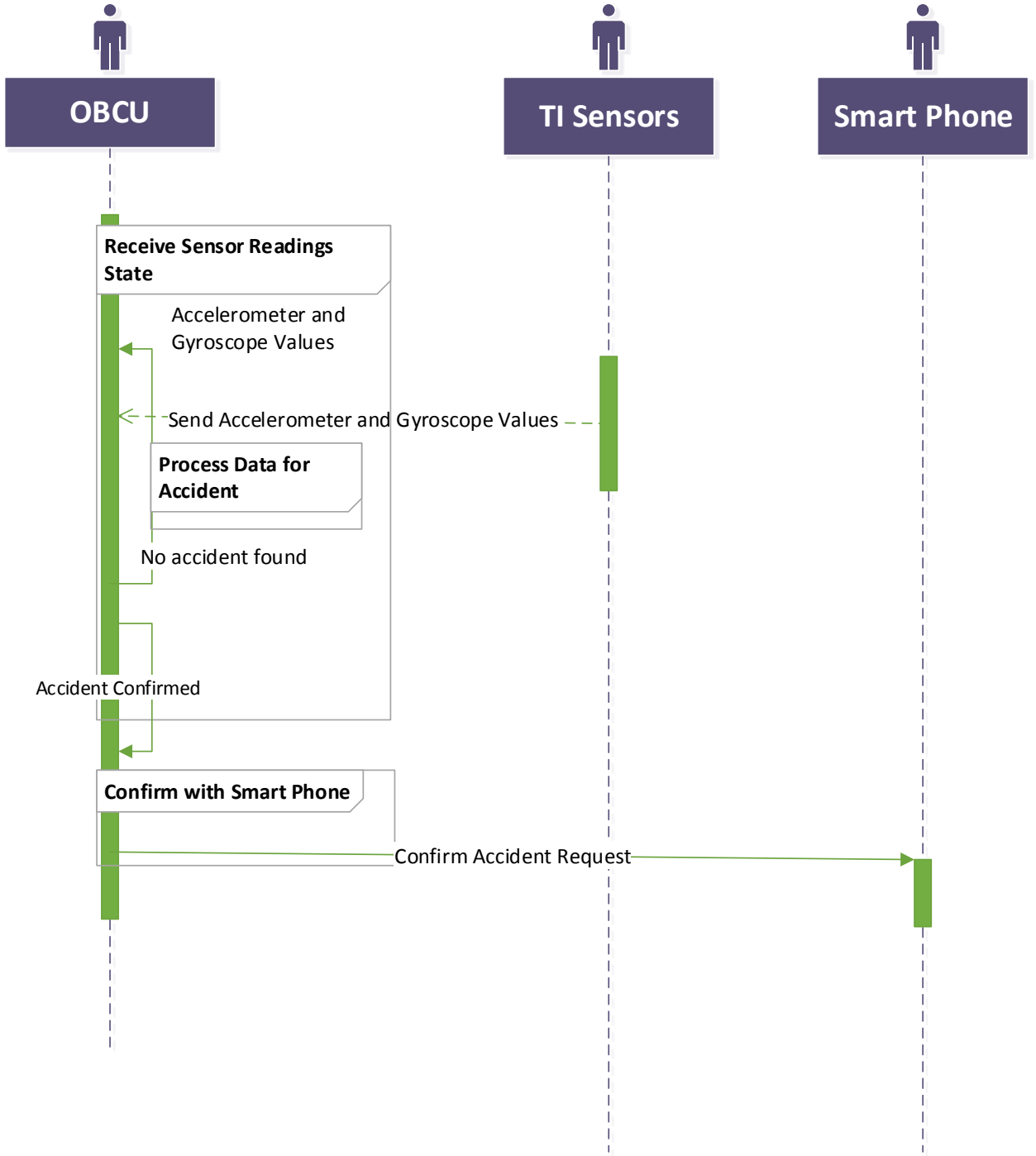


5.3. Sequence Diagrams

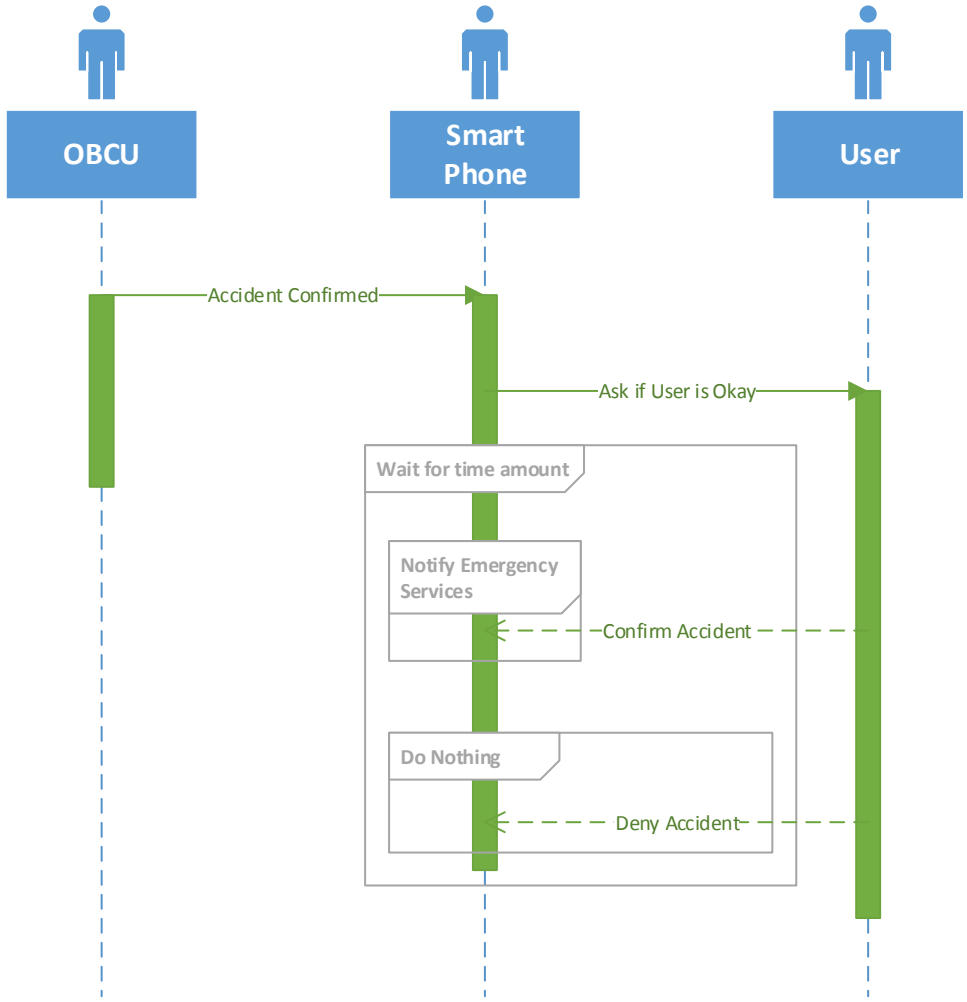
5.3.1. Overall Networking



5.3.2. Accident Detection



5.3.3. Ask if User is Unharmed



5.3.4. Networking Commands

Command	Value	Purpose
Error	-1	An error has occurred on the OBCU, an error report will follow.
Acknowledgment	12	Acknowledgment that some event has occurred successfully.
Request MACs	13	The OBCU is requesting the MAC addresses be sent to the OBCU.
Start Monitoring	14	To tell the OBCU that it can begin accident detection.
Shutdown	15	To tell the OBCU that it can begin the shutdown sequence.
Confirm Accident	16	An accident has been detected and needs to be confirmed.
No Accident	17	The detected accident could not be confirmed.
Accident Confirmed	18	The detected accident has been confirmed – alert the user.
Accident Denied	19	The user has denied emergency response.

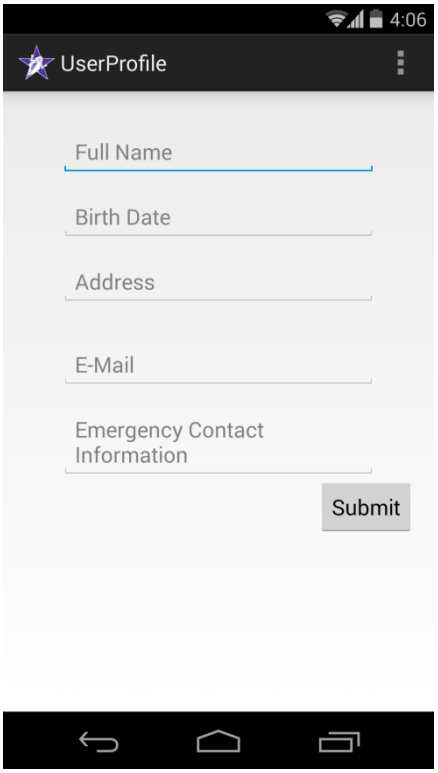
*Note: See ResultCodes.java in the smartphone application to view these commands and their values. They must also match the command values in the OBCU networking programs.

** Commands start at the number 12 because of other constants that are defined in the ResultCodes.java file.

6. User Interface Prototype

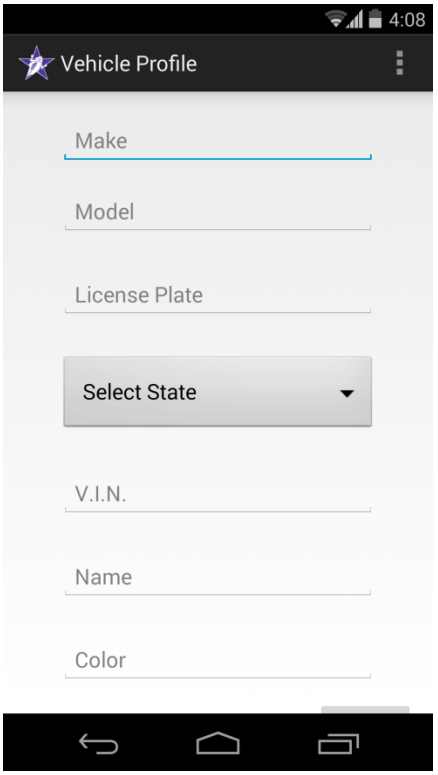
6.1. User Profile Information Edit

The User Profile screen allows the user to store his personal information in the system. The user must enter correct information into all of the text fields or he will be prompted to change it. This screen will be shown on first run or if the user would like to change his information.



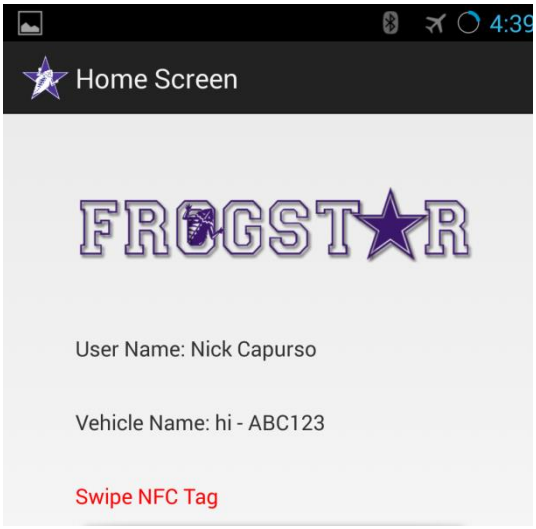
6.2. Vehicle Profile

The Vehicle Profile screen allows the user to enter information about his vehicle. If the data entered by the user is incorrect he will be prompted to fix that piece of information. On the submit button click, the vehicle data the user inputs will be stored into a database.



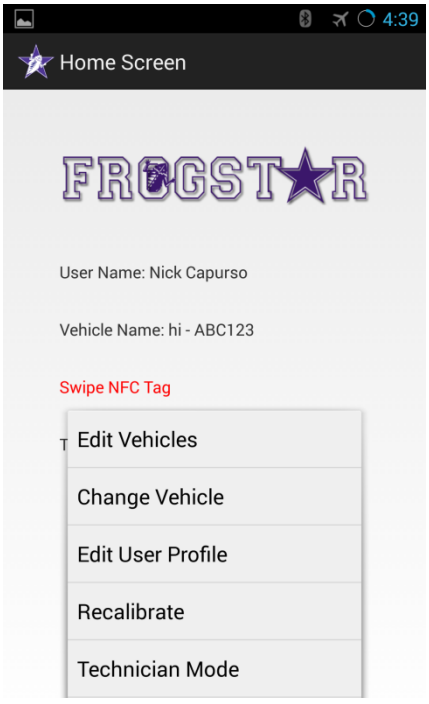
6.3. Home Screen

The main screen of the application will show the user name and vehicle that is currently being used. This screen will also show if the phone has successfully or unsuccessfully connected to the OBCU. If the connection was successful the app will display how long the entire system has been running.



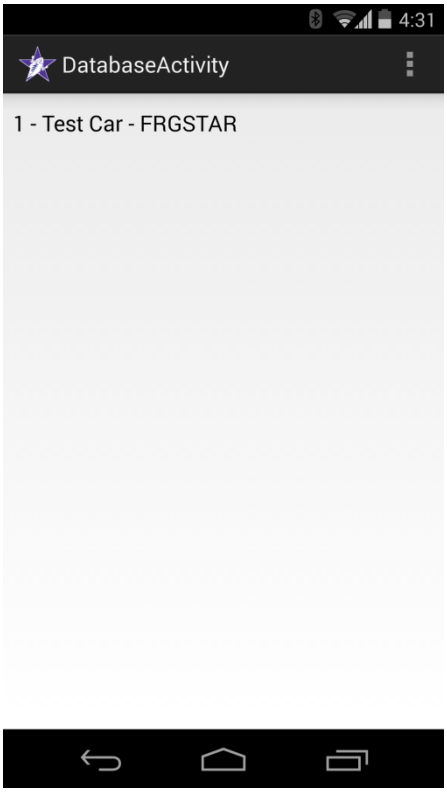
6.4. Home Screen Menu

From the home screen, the user may utilize the menu provided to edit his profile and vehicle or change which vehicle is being displayed. Also, technician mode may be accessed from this menu in order for the testing of the system.



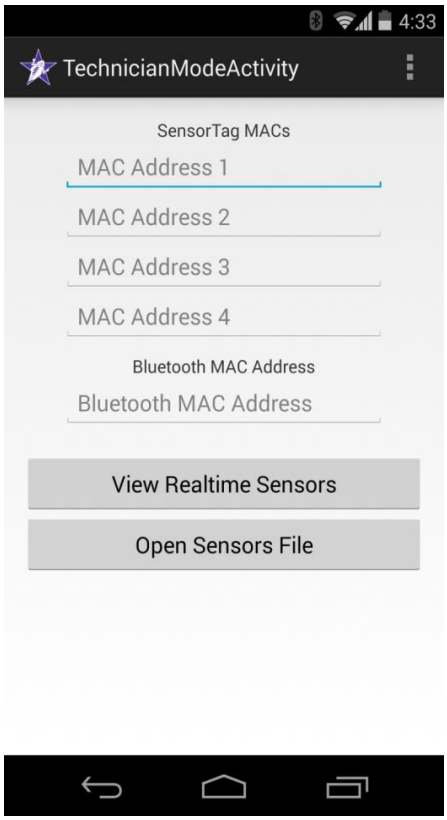
6.5. Database

The user can select which vehicle he is using at the time from the database. User can select which vehicle to edit, and even choose to delete a vehicle. Also if the user clicks on the menu button on the smartphone the user may add more vehicles to the database.



6.6. Technician Mode

The Technician mode home screen allows FrogStar Technicians to edit the MAC addresses stored on any NFC tag. Text fields are provided to set the four SensorTag MAC addresses as well as the OBCU MAC address. From here a technician may also choose to view real-time sensor readings in order to test the smartphone's sensors or open a sensors file.



7. Definition of Terms

Accelerometer – A device that measures acceleration, or the rate at which speed changes.

Android – An open-source operating system developed for mobile devices by Google.

API – Application Interface – specifies how software components interact, or interface, with each other.

Bluetooth – A short-ranged, peer-to-peer, wireless communication protocol. Bluetooth LE refers to a low-energy Bluetooth standard.

GATT – General Attribute, protocol used by Bluetooth LE communication.

Gyroscope – A device that measures orientation in terms of yaw, roll, and pitch.

NFC – Near-field communication – A set of standards that allow devices to communicate in very close proximity.

TI CC2541 – A Bluetooth-capable SensorTag offered by Texas Instruments that houses various sensors including an accelerometer and a gyroscope.

UML – Unified Modeling Language – A language specification for creating graphical models for use in software engineering.

Wi-Fi – A medium-ranged, one-to-many, wireless communication protocol.

