Requirements Specification

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 7, 2013
Version 1.1	Section 1: • Fixed wording on subsection 1.3. • Added Appendices to subsection 1.5. Section 2: • Revised constraints. Section 4: • Reworded subsection 4.2. Section 5: • Updated terminology. Section 7: • Removed and added certain terms.	November 18, 2013
Version 2.0	Change the term “collision” to “accident” on all appearances. Section 7: • Added certain terms. Appendix D - F • Edited Use-Case Scenarios to remove duplicate items. • Fixed order of Use-Case Scenarios. Appendix G • Updated prototypes.	January 22, 2014
Version 3.0	Changed the terms “smart phone” to “smartphone” and “Sensor Tag” to “SensorTag”.	March 10, 2014
Version 4.0	Updated tables to match formatting for consistency. Section 1: • Updated References	April 15, 2014
Version 4.1	Revised for consistency. Appendix D: Use-Case Scenarios • Removed System Health use case Section 2: • 2.3: Placed actions in logical order.	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 errors.

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 all of the functional and non-functional requirements for the TCU Computer Science Senior Design project, FrogStar. In addition, Use-Case diagrams and descriptions are included in order to show the functionality of the system. These requirements are given to the development team by the clients, Dr. Liran Ma and Dr. Donnell Payne, from the TCU Computer Science Department.

1.2. Intended Audience

This document is written for the development team of FrogStar. It provides the necessary references for the team in order to see what is required to finish the project. This document is also intended for Dr. Liran Ma and Dr. Donnell Payne to check their needs against verifiable requirements and give feedback.

1.3. Scope and Objectives

The main objective of FrogStar is to explore the possibility of developing an inexpensive, but highly available system that can detect a serious accident using sensor readings. A smartphone coupled with an on-board control unit (OBCU) will be the central devices of the system. These devices have the responsibility of determining if an accident has taken place as well as taking the appropriate action if a serious accident has occurred. The system could be used by anyone who owns a vehicle and a smartphone and purchases the necessary components of the system.

1.4. References

[1]CNXSOFT. (2013 July). Texas Instruments SensorTag Unboxing, Getting Started with Bluetooth Low Energy in Linux (with a Raspberry Pi). [Online]. Available: http://www.cnx-software.com/2013/07/21/texas-instruments-sensortag-unboxing-getting-started-with-bluetooth-low-energy-in-linux-with-a-raspberry-pi/

[2] Saunby, Michael. (2013 April). Raspberry Pi and TI CC2541 SensorTag. [Online]. Available: http://mike.saunby.net/2013/04/raspberry-pi-and-ti-cc2541-sensortag.html

[3] Weaver, CS. Sloan, BK. Brizendine, EJ. Bock, H. (February 2006). An analysis of maximum vehicle G forces and brain injury in motorsports crashes. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16531891

[4] White,Jules. Thompson, Chris. Turner, Hamilton. Dougherty, Brian. Schmidt, Douglas. (2013). WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones. [Online]. Available: http://www.dre.vanderbilt.edu/~schmidt/PDF/wreckwatchj.pdf

1.5. Section Overview

Section 2 – Overall Description: Lists the users, general functions, the constraints, and the needed environment for the system.

Section 3 – System Architecture: Gives a diagram of the system architecture.

Section 4 – External Interface Requirements: Describes the requirements that need to be fulfilled by components external to the system.

Section 5 – Functional Requirements: Gives a clear description of all of the functions of the software systems and its components.

Section 6 – Non-Functional Requirements: Defines constraints and behaviors of the system.

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

Appendices A-C – Use-Case Diagrams: Use-Case diagrams of the different actors in the system.

Appendices D-F – Use-Case Scenarios: Descriptions of the individual use-cases.

Appendix G – Prototypes: User interface prototype screenshots.

2. Overall Description

2.1. Product Perspective

FrogStar is a system designed to continuously monitor accelerometer and gyroscope readings in order to detect and report whether an accident has occurred. The system consists of a smartphone application and an OBCU, which interfaces with SensorTags, recording accelerometer and gyroscope readings in the interest of detecting an accident. If the data received by the OBCU from the SensorTags indicates an accident and the smartphone's sensor readings also confirm the same, the proper authorities will be alerted if the user is unable to do so.

2.2. Product Functions

FrogStar must allow for the transfer of system commands from the OBCU to a smartphone, via a Bluetooth connection. The smartphone shall query its own accelerometer and gyroscope. The OBCU shall query the SensorTags' same sensors via a Bluetooth LE connection. These data readings will be simultaneously analyzed by both the OBCU and the smartphone in order to determine whether an accident has occurred. In the event of a confirmed accident, FrogStar will alert the proper authorities. An NFC tag shall also be used to initiate startup or shutdown of the system.

2.3. User Characteristics

FrogStar is designed to be used as an alternative to highly-priced car safety systems for all vehicle owners. Vehicle owners will startup or shutdown the system via an NFC tag. Insert user and vehicle-specific profile information into the application and interact with the system appropriately to alert the correct authorities in the case of an emergency event.

2.4. Constraints

- Time Constraints
  - Development time limited by school year (May 2014).
- Smartphone Limitations
  - Limitations by battery life.
- Data Storage and Reporting
  - Amount of data that can be held in device.
- Communication Limitations
  - Smartphone must have cellular reception in order to contact the proper authorities.
  - Must have a cellular base station in range.
  - Devices must be close enough to communicate with each other.
  - Number of Bluetooth connections that can be made at once.
  - Bluetooth bandwidth requirements.

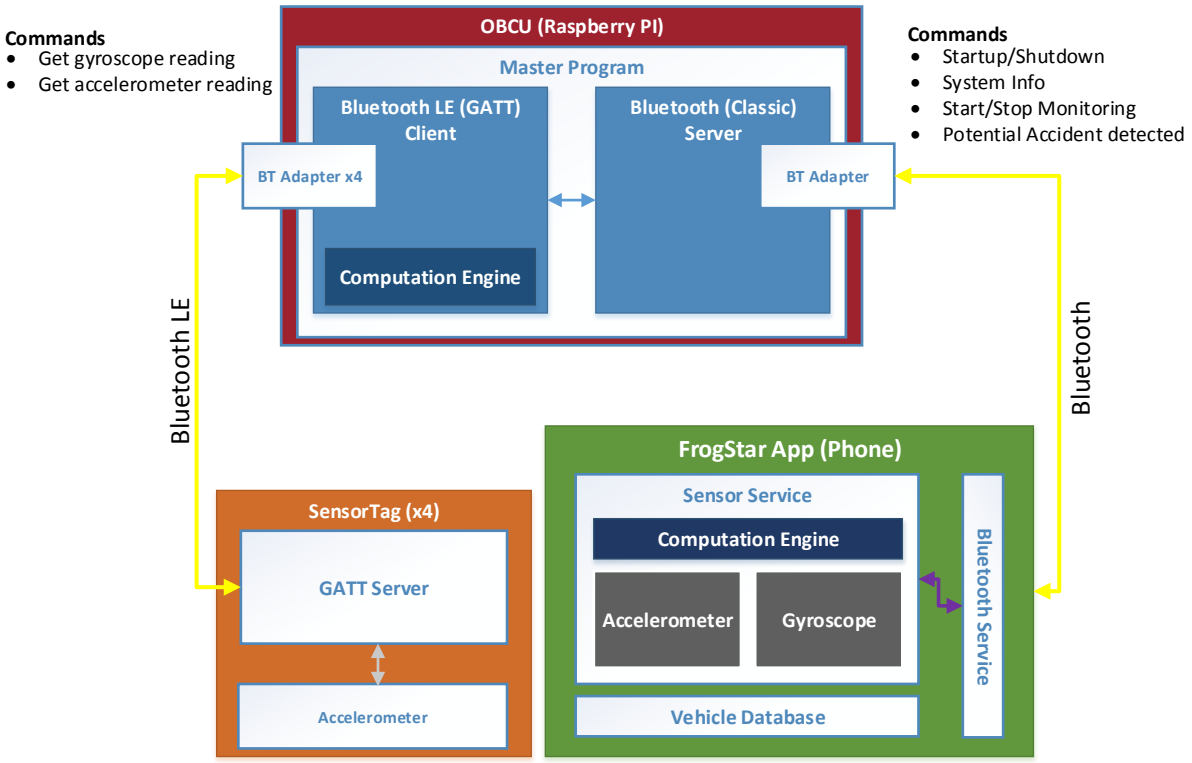
2.5. Operating Environment

FrogStar is a software application installed on smartphones for use by vehicle owners. An OBCU will monitor the SensorTags, and correlate with the smartphone. The smartphone must be able to connect to the OBCU via Bluetooth.

2.6. Assumptions and Dependencies

The system will assume that the user will swipe an NFC tag to start and shutdown the system. The user must have a charged smartphone and the OBCU must be powered by a vehicle's power source. Additionally the system must depend on the smartphone and at least one of the sensors to remain operational in the event of an accident.

3. System Architecture



4. External Interface Requirements

4.1. User Interfaces

The user interface will be intuitive to any smartphone user. Screenshots are displayed in Appendix G.

4.2. Hardware Interfaces

The user must manually swipe the NFC tags with their smartphone in order to start the system. Before user exits the vehicle, the user must swipe the NFC tag for the system to initiate a shutdown sequence.

4.3. Communication Interfaces

The system shall communicate through cellular service providers to alert the proper authorities in the event of an accident.

5. Functional Requirements

5.1. General Requirements

- An NFC tag shall be used to store network information – this tag may be scanned by a user to toggle the system on and off.
- The smartphone shall temporarily store sensor readings within the past 1 minute.
- The system shall back up data in the event of an accident.

5.2. Smartphone Requirements

- A user shall be able to manage user and car profiles through a smartphone application.
- A user shall be able to view the system information and health diagnostics through a smartphone application.
- The smartphone application shall collect its own acceleration and gyroscope sensor readings to determine if there has been an accident.
- The smartphone shall respond to OBCU requests for accident detection. The OBCU shall respond to smartphone requests pertaining to accident detection.
- Upon accident detection, the smartphone application shall give the user an option to confirm whether an accident has occurred.
- A smartphone shall alert the proper authorities in the event of an accident if the user of the smartphone does not prevent this action from taking place.

5.3. OBCU Requirements

- The OBCU shall query the four sensors by a scheduling algorithm.
- The OBCU shall halt querying sensors in the event of an accident.
- The OBCU shall collect the TI SensorTags readings to determine if there has been an accident.

6. Non-Functional Requirements

6.1. Performance Requirements

- The querying of TI CC2541 SensorTags shall be performed by a method that provides for accident detection.
- A minimal amount of storage space shall be used to accommodate data storage.
- The CC2541 SensorTags shall be queried at or above the minimum possible sampling interval.

6.2. External Requirements

- 911 emergency respondents shall only be called in the case of an actual emergency.

7. Definition of Terms

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

Accelerometer - An instrument for measuring acceleration.

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

Client – Program that is capable of obtaining a service provided by another program.

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

GPS – Global Positioning System to find exact location of user on a map.

Gyroscope – Measure 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.

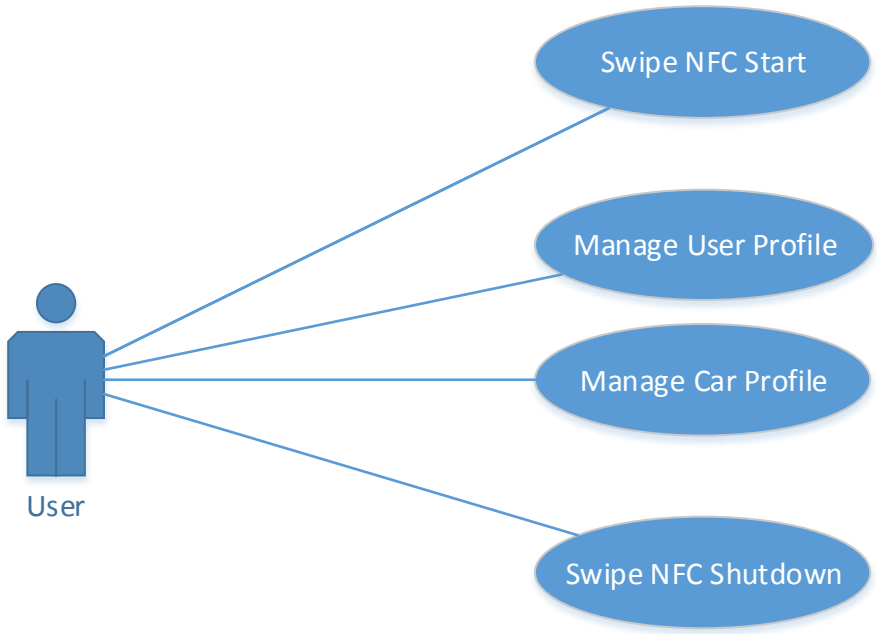
OBCU – On Board Control Unit, also known as Raspberry Pi.

Raspberry Pi – An inexpensive, single-board computer that can support an operating system and user data held on a secure digital (SD) card. Bluetooth and Wi-Fi capability can be added via adapters plugged into its USB ports.

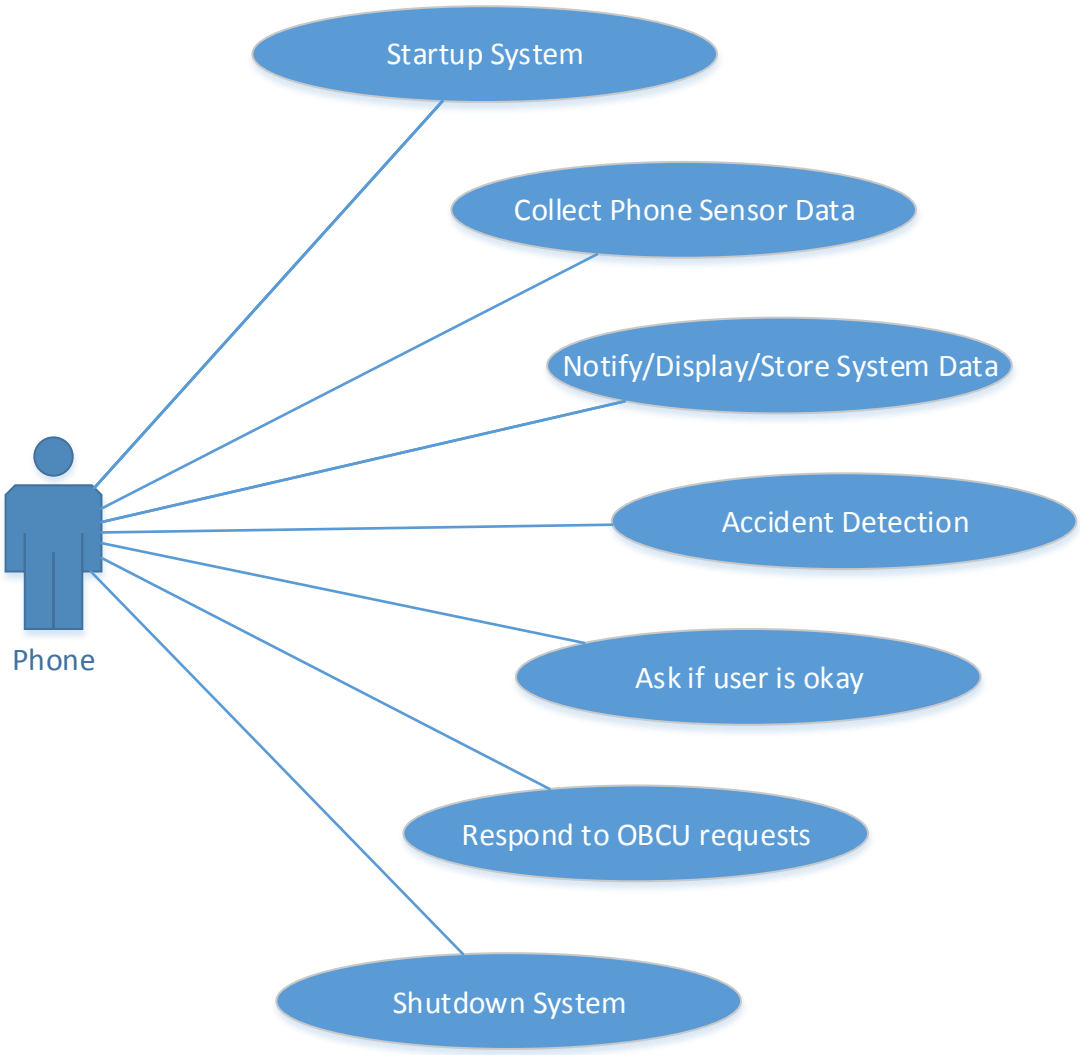
Server – Computer programs running to serve the requests of other programs, the clients.

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

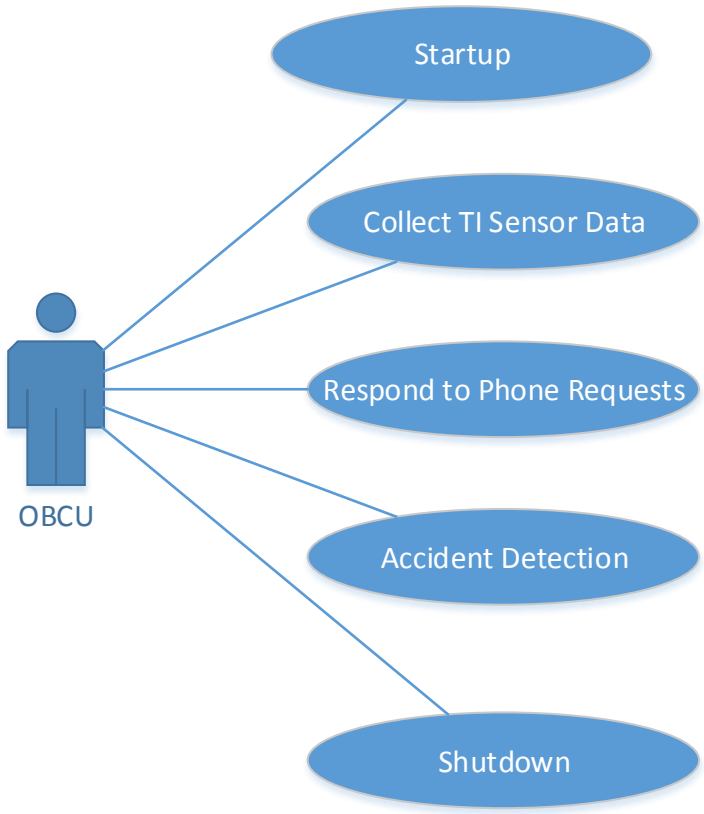
Appendix A: User Use-Case Diagram



Appendix B: Smartphone Use-Case Diagram



Appendix C: OBCU Use-Case Diagram



Appendix D: Use-Case Scenarios

Swipe NFC Start	
Actors	User
Description	Once user enters the vehicle and turns the car on, they shall hold their smartphone within the appropriate distance of the NFC tag. Once this occurs the smartphone application will trigger a startup.
Data	Bluetooth and OBCU MAC addresses.
Pre-Conditions	The NFC tag shall be preprogrammed with the appropriate data.
Triggers	User swipes the smartphone over the NFC tag.
Events	1. Application triggers system startup command to the OBCU. 2. Home screen on application will display. 3. If 1 and 2 fail a prompt will display requesting the user to swipe the NFC tag again.

Manage User Profile	
Actors	User
Description	The user can input and edit their general information.
Data	The data the user inputs will be stored will be his name, birthday, address, E-mail, Emergency Contact, and telephone number.
Pre-Conditions	The user must have the application installed on their smartphone.
Triggers	First app start or an edit button click.
Events	1. Edit screen for user data appears. If data has already been entered the data fields will be populated. 2. User interacts to edit the fields. 3. User clicks the save button, user profile is saved into the application database and screen closes. 4. If user input is invalid a notification will display prompting the user to reenter information.

Manage Vehicle Profile	
Actors	User
Description	The user can input and edit their general information about their vehicle.
Data	The data the user inputs will be stored will be his vehicle identification number (VIN), license plate number, state, make and model.
Pre-Conditions	The user must have the application installed on their smartphone.
Triggers	First app start after user profile management or an edit button click.
Events	1. Edit screen for vehicle data appears. If data has already been entered the data fields will be populated. 2. User interacts to edit the fields. 3. User clicks the save button, vehicle profile is saved, and screen closes. 4. If vehicle input is invalid a notification will display will prompt the user to reenter information.

Swipe NFC Shutdown	
Actors	User
Description	Before user exits the vehicle and turns the car off, they shall hold their smartphone within the appropriate distance of the NFC tag. Once this occurs the smartphone application will trigger a shutdown.
Data	There is no data.
Pre-Conditions	The application and system are currently running.
Triggers	When the user swipes the smartphone over the NFC tag.
Events	1. Application triggers system shutdown command to the OBCU. 2. Android Application shuts down.

Appendix E: Smartphone Use-Case Scenarios

Startup System	
Actors	Smartphone
Description	After the NFC tag has been swiped, smartphone will connect to OBCU. Sends necessary data to OBCU along with a startup command.
Data	Sends Bluetooth information to OBCU.
Pre-Conditions	The smartphone application has been started.
Triggers	NFC tag is swiped.
Events	1. Smartphone connects to OBCU. 2. Smartphone sends data and startup command to OBCU. 3. If 1 and 2 fail a prompt will display requesting the user to swipe the NFC tag again.

Collect Phone Sensor Data	
Actors	Smartphone
Description	The smartphone shall query its accelerometer and gyroscope information. Data is stored.
Data	Accelerometer and gyroscope readings.
Pre-Conditions	The application and system are running.
Triggers	System startup.
Events	1. Smartphone queries accelerometer and gyroscope information. 2. Data is stored in a flat file in the event of an accident.

Notify/Display System Data	
Actors	Smartphone
Description	If there is an inconsistency between smartphone and OBCU acceleration and gyroscope readings a notification will prompt the user to interact with the application. Additionally, if the system is deemed unhealthy then a notification will appear with a diagnosis.
Data	Sensor readings from both smartphone and OBCU.
Pre-Conditions	The application is running and the system is functioning.
Triggers	Unhealthy system and inconsistent data.
Events	1. Inconsistent data or a malfunction is determined. 2. A message is displayed with pertinent data.

Respond to OBCU Requests	
Actors	Smartphone
Description	Smartphone will be queried if the OBCU detects an accident. Smartphone will either confirm or deny that an accident has occurred.
Data	There is no data.
Pre-Conditions	The system is running.
Triggers	When the OBCU detects an accident.
Events	1. Smartphone receives a query from the OBCU. 2. If the smartphone also detects an accident, the Accident Detection event will take place.

Accident Detection	
Actors	Smartphone
Description	When the smartphone detects an accelerometer or gyroscope reading above the threshold for an accident, then it will communicate with the OBCU to confirm similar readings. If the OBCU confirms similar readings, then an accident has occurred. Finally the user is prompted to confirm an accident has occurred.
Data	Accelerometer and gyroscope readings.
Pre-Conditions	Application is running.
Triggers	Through processing data, the smartphone has determined an accident has occurred.
Events	1. Smartphone communicates with OBCU to confirm an accident has occurred. 2. A prompt will the sent to the user if an accident has occurred. 3. If an accident has not occurred, data collection will resume.

Ask if User is Okay	
Actors	Smartphone
Description	Prompts the user to check if they need emergency assistance in the event of an accident. If the user does not respond or confirms that they need assistance, emergency personnel will be notified.
Data	There is no data.
Pre-Conditions	An accident has been confirmed.
Triggers	An accident has been confirmed.
Events	1. User is prompted for emergency assistance. 2. If there is no response or the user needs assistance, emergency personnel will be notified. 3. If the user declines assistance system will shut down.

Appendix F: OBCU Use-Case Scenarios

Startup	
Actors	OBCU
Description	OBCU boots, initializes networking, and waits to receive startup command from smartphone.
Data	MAC addresses of the SensorTags.
Pre-Conditions	None.
Triggers	OBCU receives power from car.
Events	1. OBCU boots OS. 2. Initializes networking. 3. Turns on SensorTags. 4. Receives startup command. 5. Begins Collect TI Sensor Data use-case.

Collect TI Sensor Data	
Actors	OBCU
Description	The OBCU shall query the SensorTags' accelerometer and gyroscope information. Data is stored.
Data	Accelerometer and gyroscope readings.
Pre-Conditions	The system is currently running.
Triggers	OBCU Startup events have completed.
Events	1. OBCU queries SensorTags 2. Stores data readings gathered in the last minute.

Respond to Phone Requests	
Actors	OBCU
Description	When receiving a query from the smartphone, the OBCU will process the query and respond with the appropriate results.
Data	Query data from the smartphone and result.
Pre-Conditions	The system is currently running.
Triggers	OBCU receives a query from the smartphone.
Events	1. OBCU responds with appropriate data.

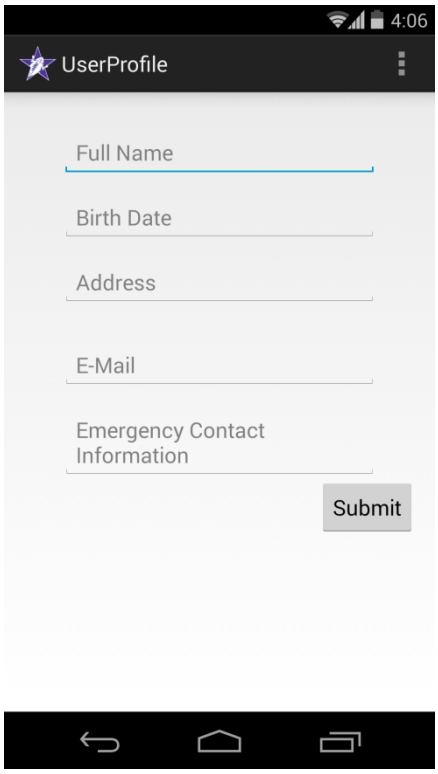
Accident Detection	
Actors	OBCU
Description	When the OBCU detects an accelerometer or gyroscope reading above the threshold for an accident, then it will communicate with the smartphone to confirm similar readings.
Data	Accelerometer and gyroscope readings.
Pre-Conditions	The system is currently running.
Triggers	OBCU detects an accident.
Events	1. Communicates with smartphone to confirm similar readings.

Shutdown	
Actors	OBCU
Description	After receiving shutdown command from the smartphone, OBCU halts data collection, and powers down.
Data	None.
Pre-Conditions	The system is currently running.
Triggers	The OBCU receives a shutdown command from the smartphone.
Events	1. The OBCU receives a shutdown command from the smartphone. 2. OBCU halts data collection. 3. OBCU powers down.

Appendix G: Prototypes

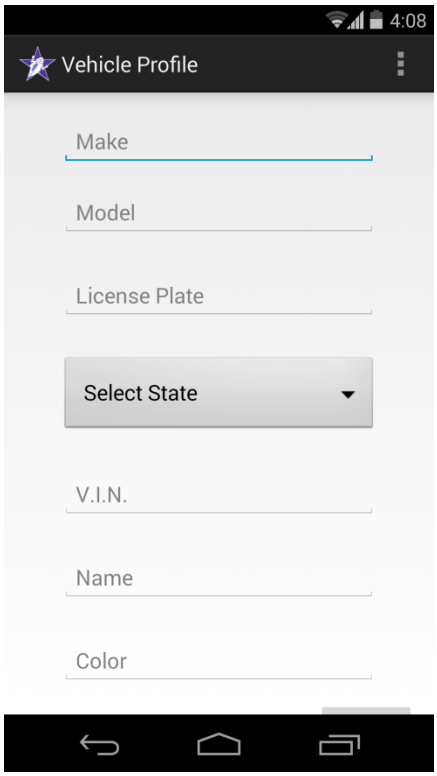
G.1. User Profile

This is the User Profile screen of the smartphone application that 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.



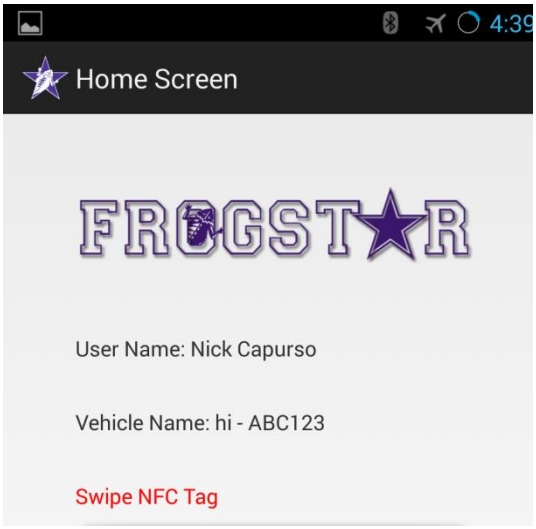
G.2. Vehicle Profile

This is the Vehicle Profile screen of the application that 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.



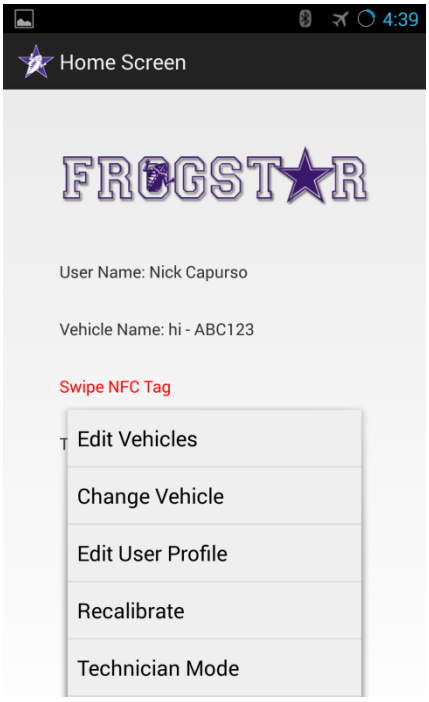
G.3. Home Screen

This is the main screen of the app. This 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.



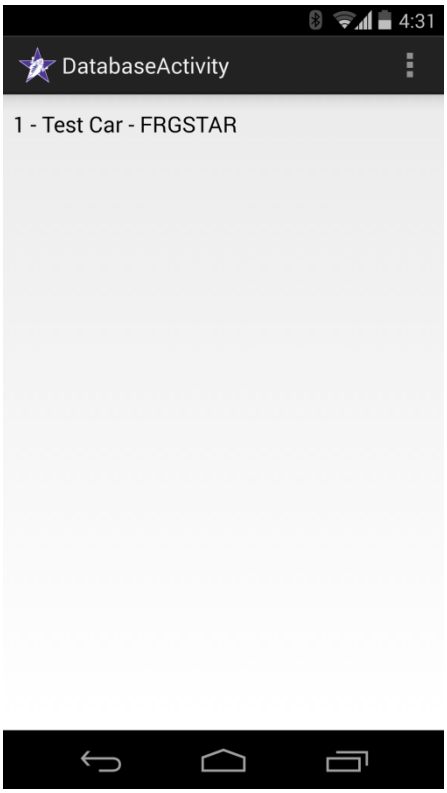
G.4. Home Screen Menu

This is the menus of the home screen. This shows where the user can go from the main screen. From here the user may select 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.



G.5. Database

This is a view of the database for the app. The user can select which vehicle he is using at the time, 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.



G.6. Technician Mode

This is the Technician mode home screen. From here, FrogStar Technicians may edit the MAC address 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.