On-Board Diagnostics Monitoring System

Sample High Level Design Document

Table of Contents

Introduction and Executive summary

This project is composed of hardware and software components. The stakeholder's ultimate goal of the project is to send information from a car to a cloud server, so that the collected data can be analyzed to improve the driver's safety. Information about the car such as the driver's speed, seat belts, acceleration, breaks, will be gathered from the OBD-II (On Board Diagnostics) system in the car.

This system will be sending the information from car to server for a company to analysis collected data from the fleet of the cars. We will be using the Raspberry Pi in order to collect the data from OBD-II (On Board diagnostics) and transferring data over Wi-Fi. The data will be transferred to backend server for analysis. The Raspberry Pi 3 has an Ethernet network adapter, Wi-Fi adapter, and SD card adapter available. These features of device will help us to divide the problem in small modules. The main task for the developers is to manipulate the Raspberry Pi to connect to the OBDII. Then, the incoming bits of information will be converted to human readable form. This is followed by storing this information on a SD card and lastly, send the invaluable information to the backend server via wireless streams. The data should be sent/received when there is an available Wi-Fi connection and device is connected. Otherwise it will be stored on the SD card for future transfer. The OBDII collects important data and information about vehicle and driver's driving behavior. The main focus of this project is *collecting and analyzing the information*, which will be a valuable asset to various stakeholders within the industry

In order to accomplish this task a few alternate solutions were discussed. While doing some research, some existing solutions seemed to exist, however those solutions do not satisfy the stakeholder's requirements or are not fully functional. Based on the research and discussions done by the group, the final choice was to implement a solution involving the Raspberry pi 3. The Raspberry Pi provides many advantageous characteristics for this project. If monetary cost is solely considered, the RPi is clearly the most cost-effective alternative.

The RPi also provides a flexible and adaptable environment for development. It is compatible with many useful ports, such as USB, Micro SD, HDMI, while having various options for its power source. This allows the team to create a custom product that more closely meets the functional and non-functional requirements of the project, with fewer limitations. Since the RPi has very large compatibility with various interfaces, the device is seen as the most optimal solution.

Stakeholder's User Stories

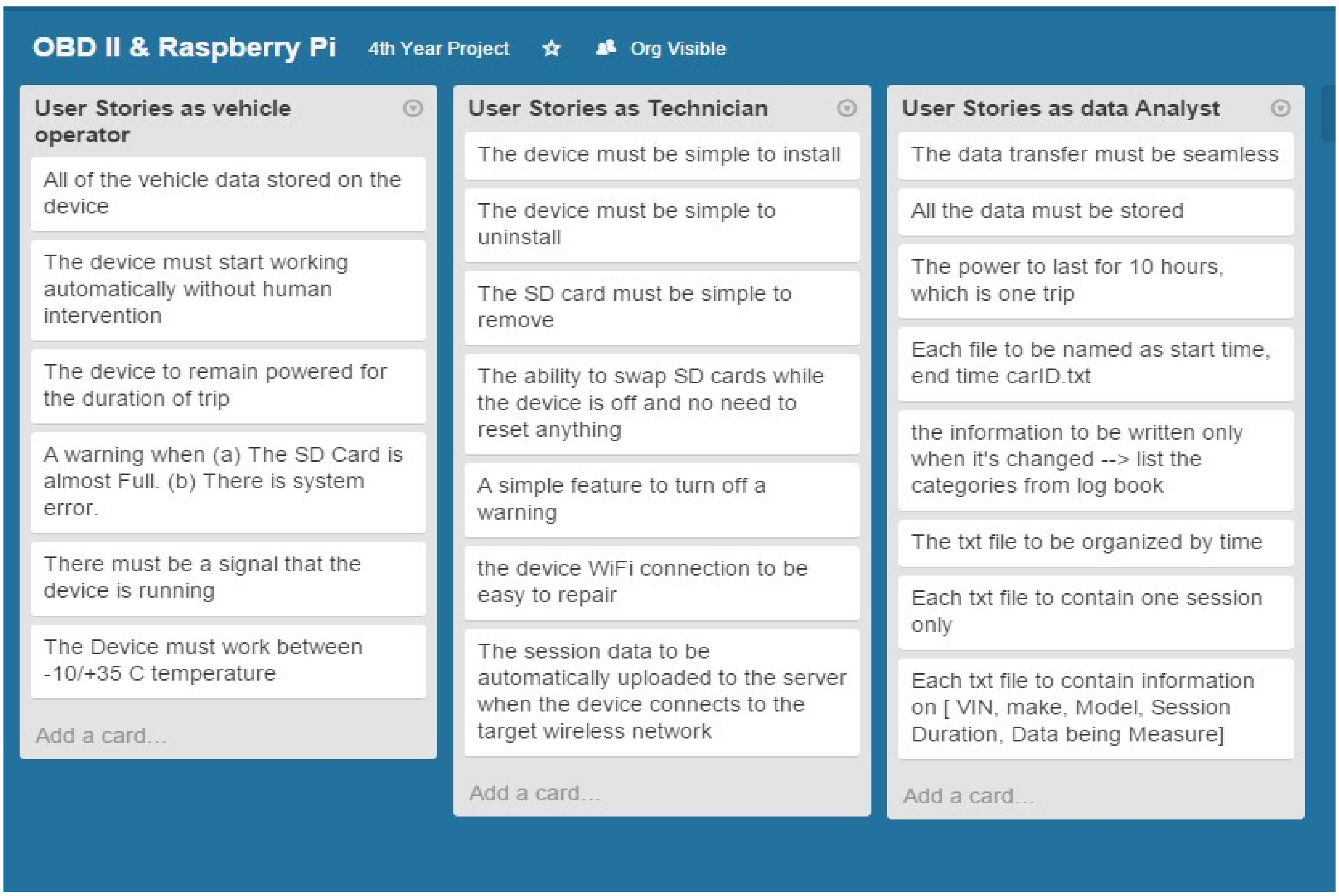


Figure 1: User Stories

Detailed User Stories

As a vehicle operator, I want	As a technician I want	As a Data Analyst I want
1. all of the vehicle data stored on the device 2. the device to automatically start working without intervention 3. the device to remain powered for the duration of the trip 4. a warning when the sd a. card is almost full there b. is a system error a signal 5. that the device is running the device to work between 6. 10/+35C temperatures	1. the device to be simple to install 2. the device to be simple to uninstall 3. the sd card to be simple to remove 4. the ability to swap SD cards while the device is off and not need to reset anything 5. a simple feature to turn off a warning 6. the device to connect automatically when it is within 30m of the target wireless network 7. the device wifi connection to be easy to repair 8. the session data to automatically upload to the server when the device connects to the target wireless network	1. the data transfer to be seamless 2. all the data to be stored 3. the power to last for 10 hours, which is one trip. 4. each file to be named start time, end time carID.txt 5. information to be written only when it’s changed -> list the category from log book 6. the txt file to be organized by time 7. each txt file to contain one session only 8. each txt file to contain information on: i. VIN ii. make iii. model iv. session duration v. data being measure (example, RPM)

Table 3: detailed user stories

Use Case Diagram

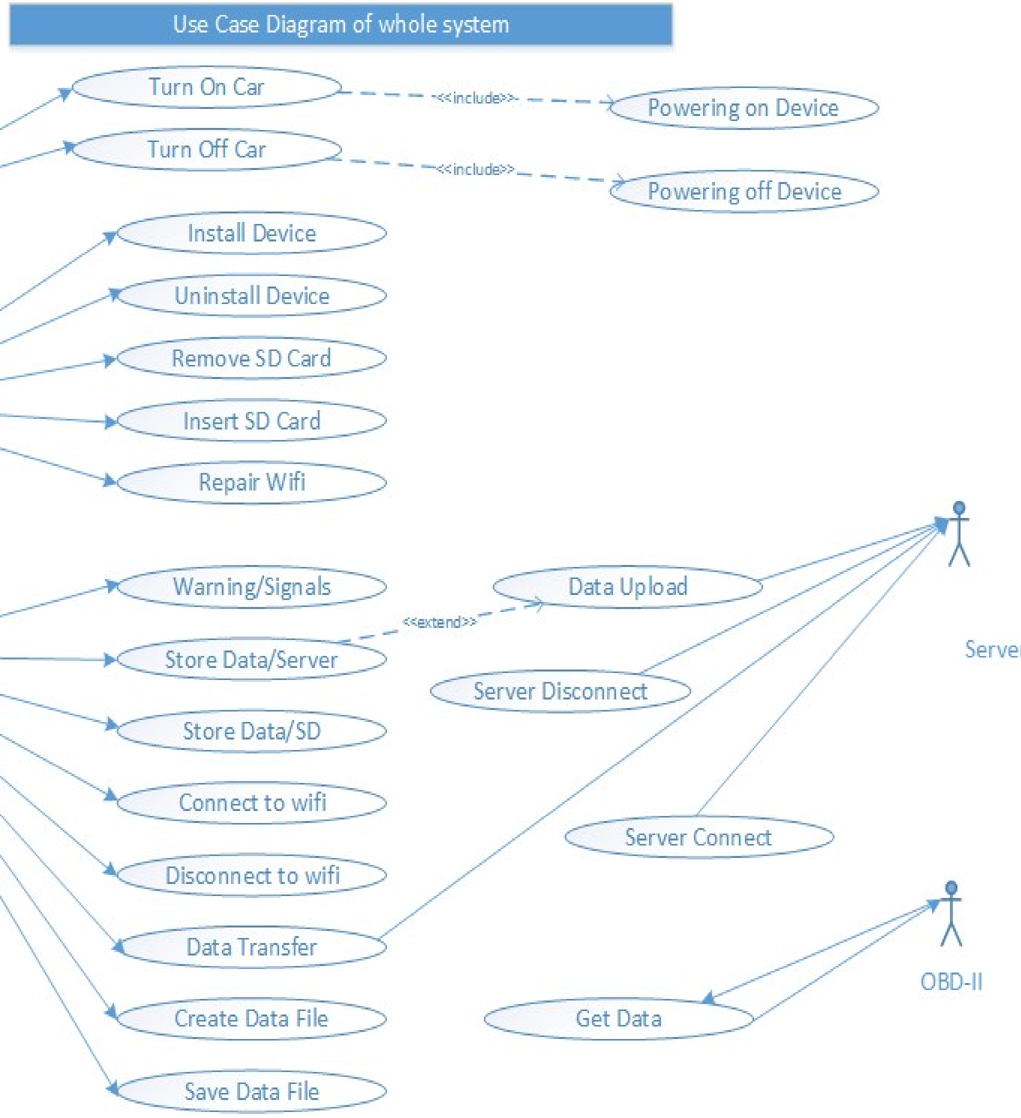


Figure 2: use case diagram of system

Data Flow Diagram of the system

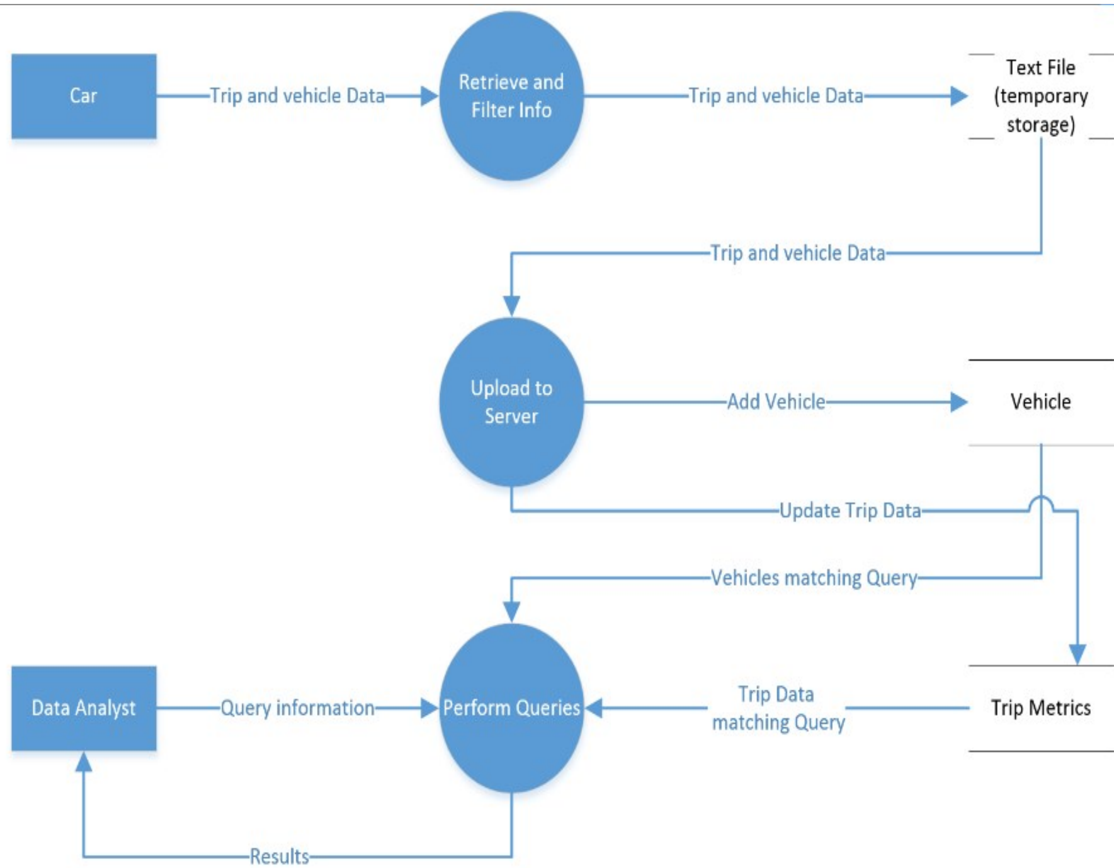


Figure 3: Data flow diagram

High Fidelity Prototype

Architectural Diagram of the whole system

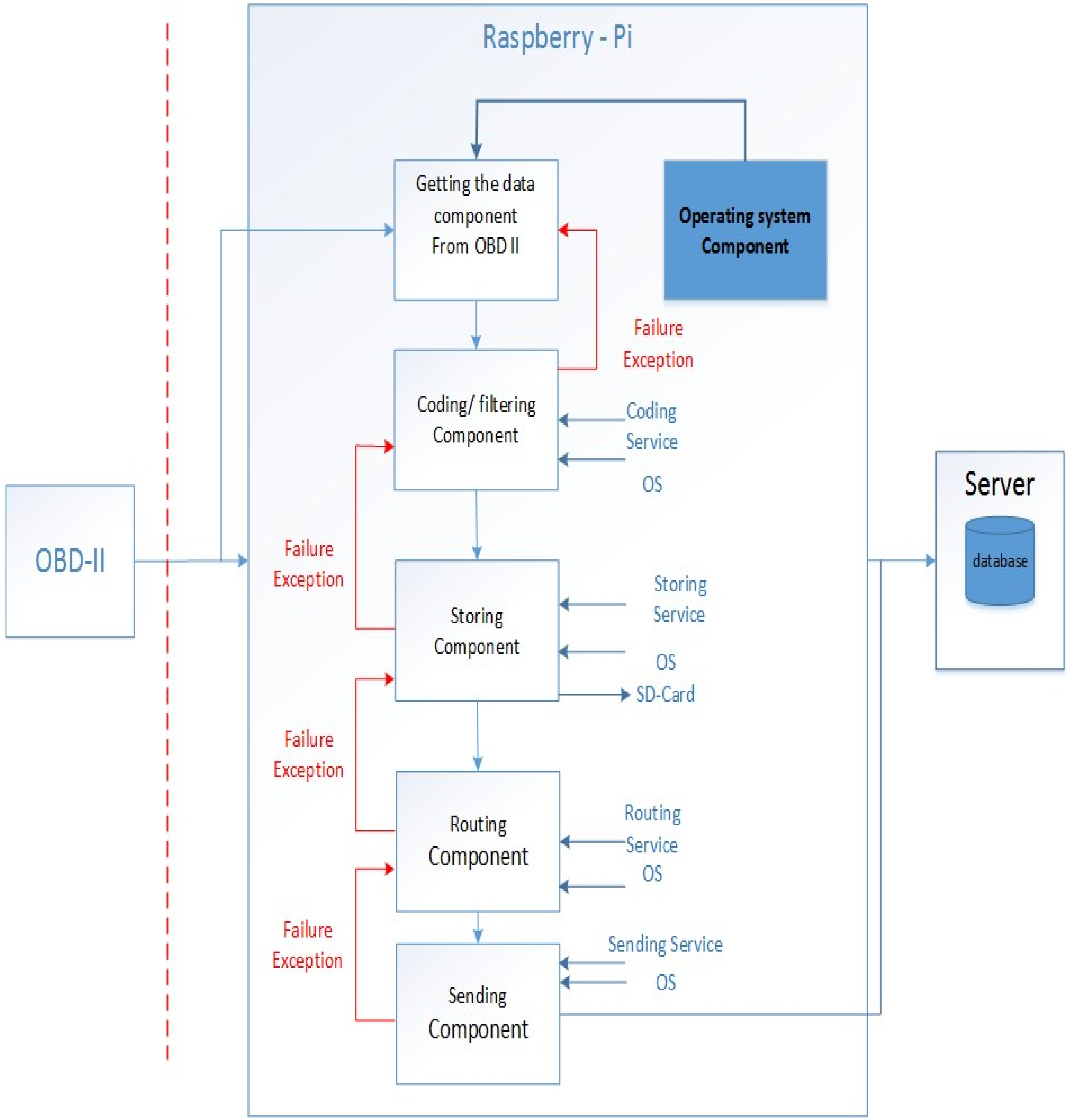
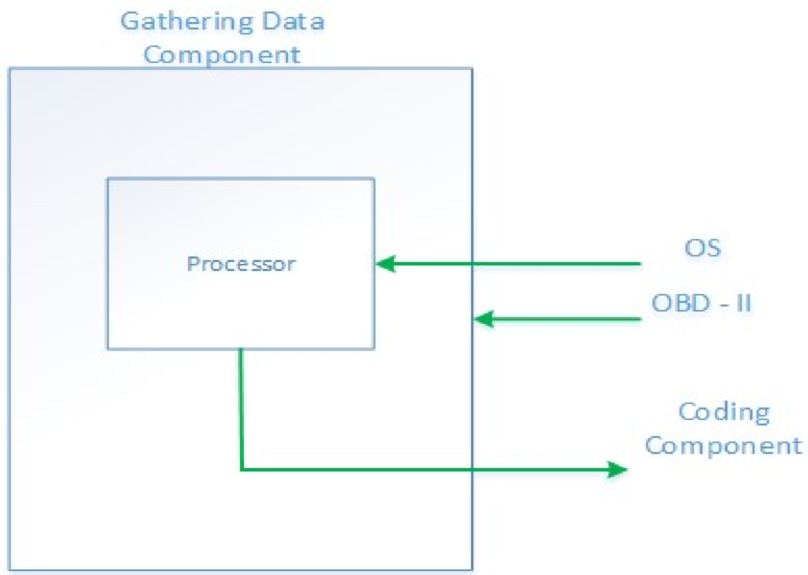
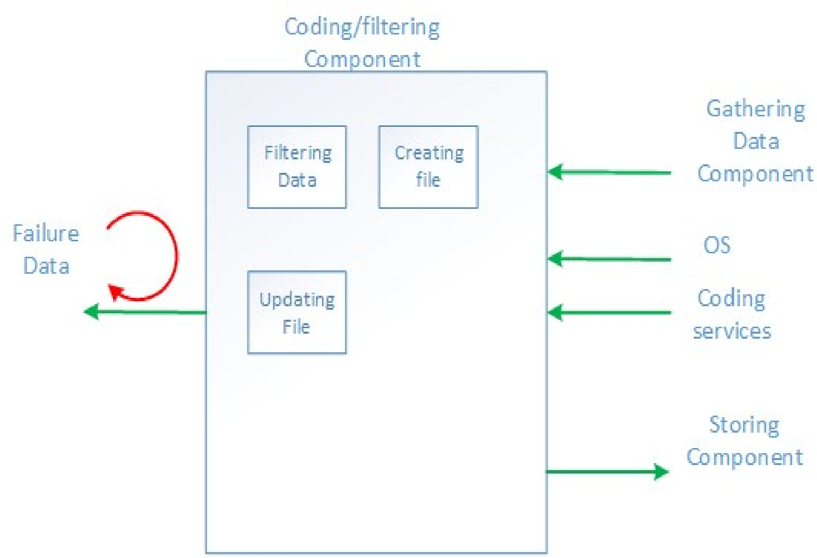
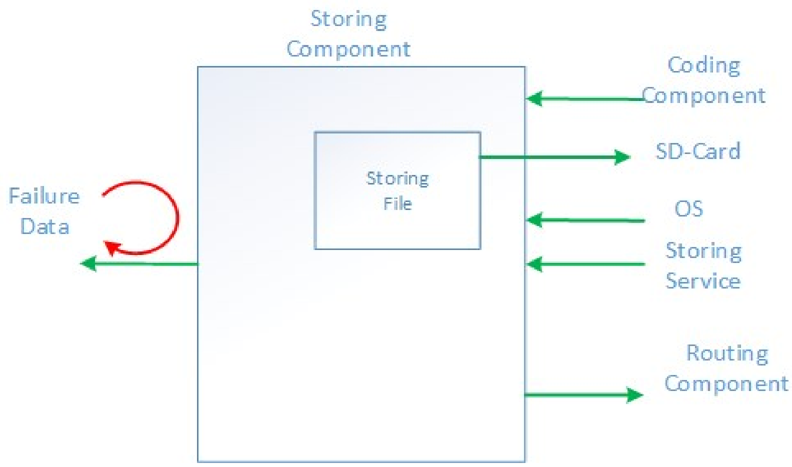


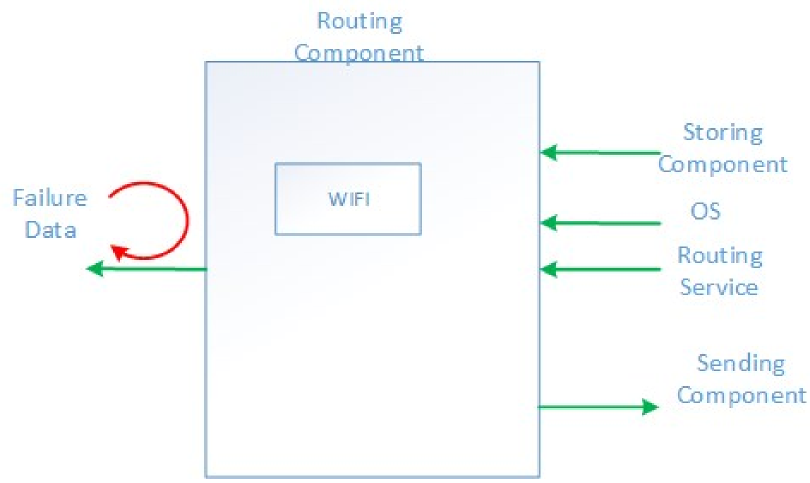
Figure 4: High fidelity diagram

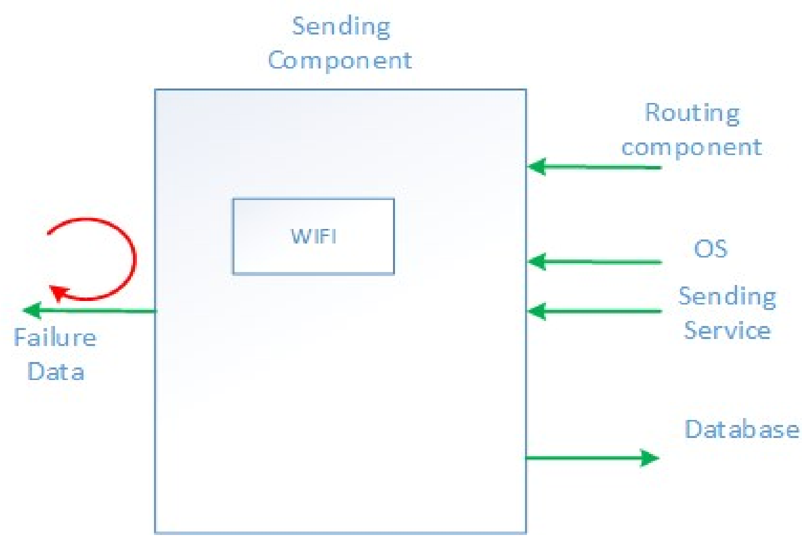
Modules of the system

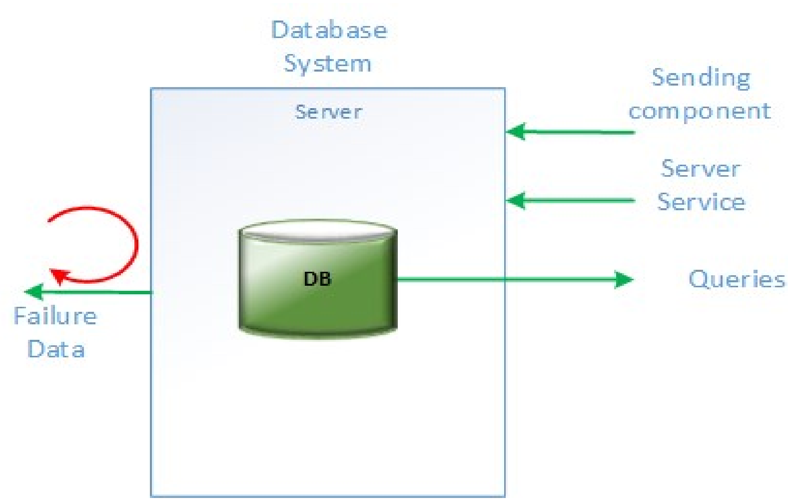
Modules	Architectural Diagram of Modules

Module 1 - Gathering Data	 Gathering the data from car through OBD-II connector. The data in binary form will sent through the 6 pin cable to the Raspberry Pi. There is one translator which will help with managing the different voltages. The Raspberry Pi recognizes that the connection has been made and it would be ready to send the data to the next module.
Module 2 - Coding/Filtering component	 Data filtering will be happen in the raspberry pi it self. In this step, the data that has been collected in the first module is filtered through to extract the desired information such as as

	driving speed, RPM, seat belt, over breaking will get pulled out of the input from the OBD-II, and the data will be transferred to next module.
Module 3 - Storing Component	 Data storage module will saving the filtered data onto the SD Card.This is made possible by using the SD card adapter that is supplied by the Raspberry Pi. The SD card is accessed, a text file is created on the SD card, and the filtered data is saved to the file. The data is therefore backed up on the device, however the filtered data will be transferred to next module whenever the wifi connection is available.

Module 4 - Routing Component	 Connection with backend server database. This task is going to be handled by two options available on the Raspberry Pi, an Ethernet adapter, and WiFi adapter. In this module, a connection is made to the backend server through which ever RPi adapter has current activity. Once a connection to the server is established, the data will be transferred to sending component.

<table>
<tr>
<td>Module 5 - Sending Component</td>
<td>


Data sending component within Raspberry Pi, which will help the filter data to send to the database. Since the connection to the server has already been established in the previous module, the data file that is saved on the SD card is simply uploaded to the distributed database which would be on the server
</td>
</tr>
<tr>
<td>Module 6 - Database</td>
<td>


Cassandra is an open source, distributed, decentralized, elastically scalable, highly available, fault tolerant, and column oriented database. It is capable of running on multiple machines and there is no single point of failure. It also has peer to peer architecture and no master-slave issue. There is no single node that holds the entire data, but instead each holds a small part of it.
</td>
</tr>
</table>

Table 4: Modules of system

Sequence Diagram/Event Tree Diagram

For Technician:

Figure 11: Sequence diagram for technician

For Vehicle driver:

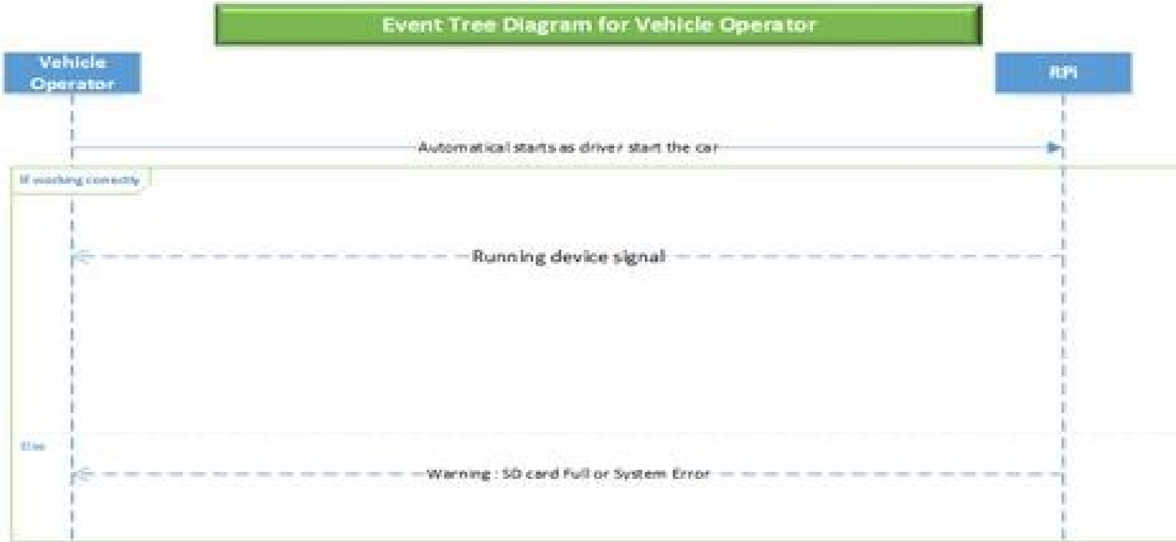


Figure 12: sequence diagram for driver

For Data Analyst:

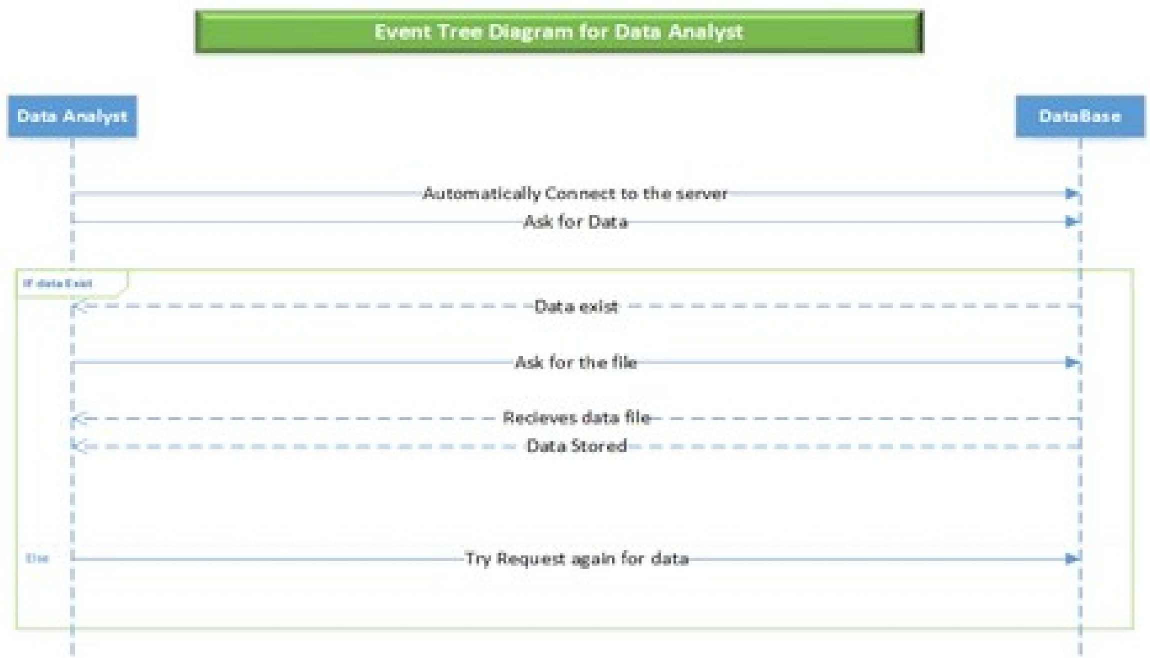


Figure 13: sequence diagram for data analyst

Glossary

OBD-II: On Board Diagnostics system. Information regarding the driver's actions will be drawn from here.

Raspberry Pi: "Raspberry Pi is a low-cost, basic computer that was originally intended to help spur interest in computing among school-aged children. The Raspberry Pi is contained on a single circuit board and features ports for:

- HDMI
- USB 2.0
- Composite video
- Analog audio
- Power
- Internet
- SD Card

The computer runs entirely on open-source software and gives students the ability to mix and match software according to the work they wish to do [1] .

User stories: captures the 'who', 'what' and 'why' of a requirement in a simple and concise way

Cassandra database: "Apache **Cassandra** is an open source distributed database management system designed to handle large amounts of **data** across many commodity servers, providing high availability with no single point of failure. [2]

Use case diagram: is a representation of a user's interaction with the system.

Data Flow diagram: is a graphical representation of the "flow" of data through the whole system.

Sequence diagram: is an interaction diagram that shows how processes operate with one another and in what order

References

[1] Buyapi.ca, 'Raspberry Pi', 2014. [Online]. Available: http://www.buyapi.ca/productcategory/raspberry-pi/.

[2] Cassandra, 'The Apache Cassandra Project', 2014. [Online]. Available: http://cassandra.apache.org/.