

Software Engineering Project Report

Requirement Document for AutoDiagScan (ADS) Mobile Application

Prepared by

**Grant Moy, Onome Ovedje, Annie To, Muthiah Muthu Palaniappan
University of Illinois Chicago**

February 2015

Table of Contents

I	Project Description	7
1	Project Overview	7
2	The Purpose of the Project	7
2a	The User Business or Background of the Project Effort.....	7
2b	Goals of the Project.....	7
2c	Measurement.....	7
3	The Scope of the Work.....	7
3a	The Current Situation.....	8
3b	The Context of the Work.....	8
3c	Work Partitioning.....	9
3d	Competing Products	10
4	The Scope of the Product	10
4a	Scenario Diagram(s)	12
4b	Product Scenario List	13
4c	Individual Product Scenarios	13
5	Stakeholders	16
5a	The Client.....	16
5b	The Customer	16
5c	Hands-On Users of the Product	16
5d	Priorities Assigned to Users	17
5e	User Participation.....	17
5f	Maintenance Users and Service Technicians.....	17
5g	Other Stakeholders	17
6	Mandated Constraints.....	18
6a	Solution Constraints.....	18
6b	Implementation Environment of the Current System	18
6c	Partner or Collaborative Applications	18
6d	Off-the-Shelf Software.....	19
6e	Anticipated Workplace Environment	19
6f	Schedule Constraints.....	19
6g	Budget Constraints	19
7	Naming Conventions and Definitions	19
7a	Definitions of Key Terms	19
7b	UML and Other Notation Used in This Document	20
7c	Data Dictionary for Any Included Models	20
8	Relevant Facts and Assumptions.....	20

8a	Facts	20
8b	Assumptions	20
II	Requirements	21
9	Product Use Cases	21
9a	Use Case Diagrams	21
9b	Product Use Case List	21
9c	Individual Product Use Cases	22
10	Functional Requirements	26
11	Data Requirements	30
12	Performance Requirements	30
12a	Speed and Latency Requirements	30
12b	Precision or Accuracy Requirements	31
12c	Capacity Requirements	31
13	Dependability Requirements	32
13a	Reliability Requirements	32
13b	Availability Requirements	32
13c	Robustness or Fault-Tolerance Requirements	32
13d	Safety-Critical Requirements	33
14	Maintainability and Supportability Requirements	33
14a	Maintenance Requirements	33
14b	Supportability Requirements	33
14c	Adaptability Requirements	33
14d	Scalability or Extensibility Requirements	34
14e	Longevity Requirements	34
15	Security Requirements	34
15a	Access Requirements	34
15b	Integrity Requirements	34
15c	Privacy Requirements	35
15d	Audit Requirements	35
15e	Immunity Requirements	35
16	Usability and Humanity Requirements	35
16a	Ease of Use Requirements	35
16b	Personalization and Internationalization Requirements	35
16c	Learning Requirements	35
16d	Understandability and Politeness Requirements	36
16e	Accessibility Requirements	36
16f	User Documentation Requirements	36
16g	Training Requirements	36

17	Look and Feel Requirements	36
17a	Appearance Requirements.....	36
17b	Style Requirements	36
18	Operational and Environmental Requirements	36
18a	Expected Physical Environment.....	36
18b	Requirements for Interfacing with Adjacent Systems.....	37
18c	Productization Requirements	37
18d	Release Requirements	37
19	Cultural and Political Requirements.....	37
19a	Cultural Requirements.....	37
19b	Political Requirements	37
20	Legal Requirements.....	38
20a	Compliance Requirements	38
20b	Standards Requirements.....	38
III	Design.....	38
21	System Design	38
21a	Design goals	38
22	Current Software Architecture	39
23	Proposed Software Architecture.....	39
23a	Overview	39
23b	Class Diagrams.....	40
23c	Dynamic Model.....	42
23d	Subsystem Decomposition	50
23e	Hardware / software mapping	50
23f	Data Dictionary	51
23g	Persistent Data management	52
23h	Access control and security.....	52
23i	Global software control.....	52
23j	Boundary conditions	53
24	Subsystem services	53
25	User Interface	53
26	Object Design	60
26a	Object Design trade-offs	60
26b	Interface Documentation guidelines.....	60
26c	Packages	61
26d	Class Interfaces	61

IV	Test Plans.....	62
27	Features to be tested / not to be tested	62
28	Pass / Fail Criteria	63
29	Approach	63
30	Suspension and Resumption.....	63
31	Testing Materials (hardware / software requirements).....	64
32	Test Cases	65
33	Testing Schedule	71
V	Project Issues	71
34	Open Issues.....	71
35	Off-The-Shelf Solutions	72
35a	Ready-Made Products	72
35b	Reusable Components	72
35c	Products That Can Be Copied	72
36	New Problems	72
36a	Effects on Current Environment	72
36b	Effects on the Installed Systems.....	72
36c	Potential User Problems.....	73
36d	Limitations in the Anticipated Implementation Environment that may inhibit the New Product	73
36e	Follow-Up Problems	73
37	Tasks.....	73
37a	Project Planning	73
37b	Planning of the Development Phase.....	74
38	Migration to the New Product	75
38a	Requirements for Migration to the New Product	75
38b	Data That Has to Be Modified or Translated for the New System	75
39	Risks	75
40	Costs	76
41	Waiting Room	76
42	Ideas for Solution	76

43	Project Retrospective	77
VI	Glossary	77
VII	References/Bibliography	77
VIII	Index	78

I Project Description

1 Project Overview

Autodiagscan is a combination of a mobile application with a hardware device. Its goal is to scan, detect, and report malfunction issues or problems of a car. The mobile application is installed on any mobile phone device and all tablets. The hardware is connected to the mobile device and to the car's computer via the OBD 2. (On board Diagnostic)

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Many car owners do not know how to fix their car until they have to go to a mechanic. Our product aims to equip car owners with a device that allows them to function independent of a mechanic, if necessary. The device provides the car user with preventive measures. The device scans the car and notifies the car owner of the problem, the part that needs to be changed, and directs you to a mechanic that can fix it, if necessary, or provide a step by step tutorial if you want to fix the vehicle yourself. The business will generate revenue by allowing auto part retailers and auto retailers to advertise their product and services on the site.

2b Goals of the Project

The goal of this product is to have the user save their time and money from going to a mechanic or auto shop. They can detect problems on their own. The notifications from the device allow the user to take more proactive preventive of their car, thereby saving the user money and time.

2c Measurement

The goals of the Autodiagscan are measurable. The first measurement that we will use is a survey online to access the customer satisfaction of the application. We will ascertain whether the application allows customers to know more about their vehicle and whether it helps them figure out minor car issues that they can correct themselves. We will also know the direction that we should go in, in order to help us improve the product. The second measurement that we will use is the amount of retailers, auto service providers, and car manufacturers that want to advertise their products and services on our site. Finally, we will measure the percentage of new app users daily.

3 The Scope of the Work

The product is a connector device that can connect your car's computer system with your smart phone to find out the exact problem that your vehicle's computer system is trying to report, when it triggers the maintenance sign in your dashboard. The product also includes a mobile Application that will help in finding and displaying the exact

problem. Apart from this the Mobile Application will have some additional features to help in the real time monitoring of your vehicle.

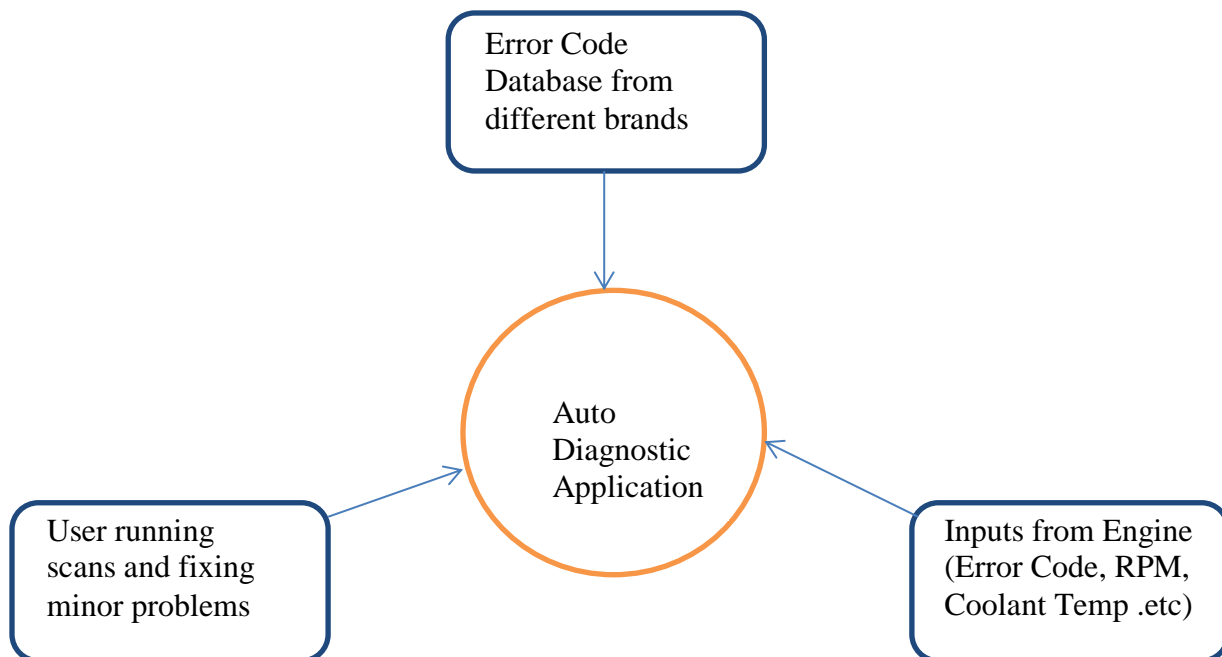
3a The Current Situation

Today car is the most important means of transport for our daily commutation between home and office. Everyone knows how to drive but not all of them know the mechanics that's involved in the car. Every car that's manufactured today has an inbuilt computer in it, which controls the overall functioning of the car, and monitors the engine and other tightly coupled systems in it. The computer sends out a signal to the driver when it finds something that's above or below a specified limit. The very next thing that the owner has to do is to take the vehicle to a mechanic to get it fixed, irrespective of the problem being small or big. The mechanics will plug in a Diagnostic Scanner into the vehicle's computer system and will find issue by reading the error codes that the system sends. Having a mechanic to fix a small problem is very expensive. This new device will help in handling the problem more effectively.

3b The Context of the Work

In order to develop an application that will let us know the vehicle's problem we will have to briefly study the following,

- The behavior of the Computers installed in the cars is to be studied to get a better understanding of how they work and how they are fixed in case of failure.
- At present there is an instrument named the Diagnostic Scanner that's used to read the error codes from the cars computer. This instrument has to be analyzed to build the connector for our application, which will read the error codes from the inbuilt computers.
- The nature of the failure that usually occurs in a vehicle and the severity of it have to be studied so that the suggested solution in the application will be more effective.
- The error codes for different car brands and models are to be retrieved and analyzed for implementing the same in the application that is to be developed.
- The basic recovery techniques should be studied for the problems that can be fixed without a mechanic.
- The basic working of a car has to be studied.



3c Work Partitioning

The following are the input and output feeds of the system. The input and output are represented using the notation in and out.

Event Name	Input and Output	Summary
When there is a problem with the car, the warning sign will be displayed on the dashboard.	<p>The user will have to connect the device to the vehicle's computer system.</p> <p>The error code from the vehicle's computer system will be pulled into the phone (in)</p> <p>The error code is then analyzed and the problem description is given to the user (out)</p>	The user or the driver on seeing the description will know what went wrong with the vehicle and will also be informed about the severity of the problem.
Checking the health of the vehicle's engine	<p>The user will have to connect the device to the vehicle's computer system.</p> <p>The application reads the live data from the vehicle's computer system (in)</p> <p>The application projects the data that's read onto the application screen for the user</p>	This will give the user or the driver with a brief description of the overall performance of the vehicle thereby helping him to find any potential problem that would occur in the near future.

	to view (out)	
Error code database update	<p>The latest error code database will be downloaded in the mobile upon the user's permission. (In)</p> <p>This will be used later when the user analyses the problem with the vehicle based on the error code. (out)</p>	The user will be notified when the error code database is updated. When the user chooses to update then all the error codes are updated with the latest data and are used for diagnosis later on.
Retrieving saved data	The user will open the application and open a previously saved report (out)	The user can open a previously saved report to check the occurrence of a certain problem over a period of time. And it can also be used to give a heads up to the Mechanic.

3d Competing Products

There are many Auto Diagnostic Scanners that's available in the market today for analyzing the errors given out by the inbuilt computer in the car, but there isn't a mobile device that can analyze the error codes given out by the inbuilt computer system and keep the user informed about the problem. It just doesn't stop there, it also tells the user about the severity of the problem. If the problem can be fixed without visiting a mechanic then the user will be briefed with the instruction to get the problem fixed without having to visit a Mechanic for a minor issue.

All the operations that this application can perform will be implemented by analyzing the existing fail safe procedures that being used. This application will come in handy especially when we are on a road trip all on our own without much of knowledge about the cars functioning.

4 The Scope of the Product

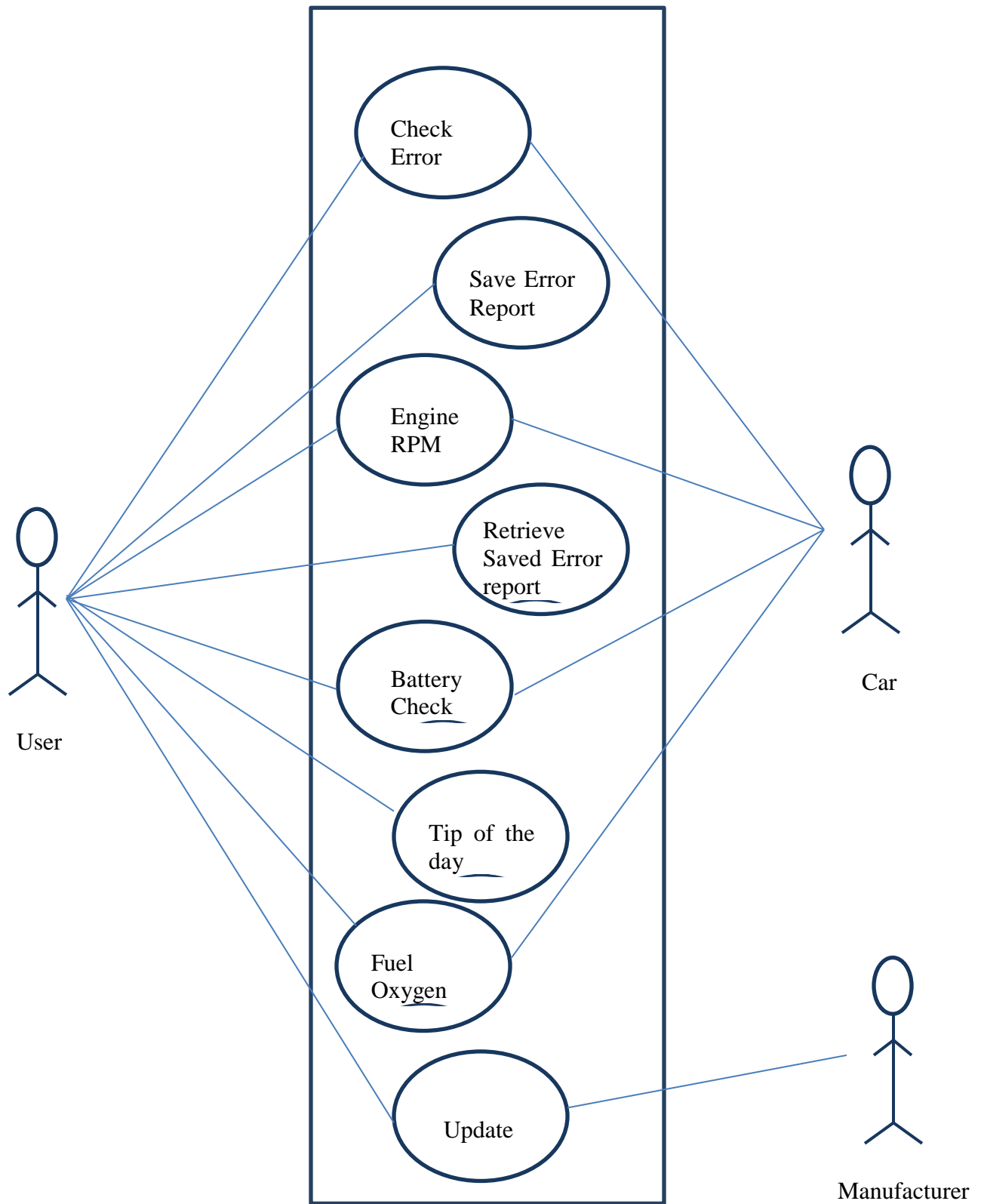
The mobile application for Auto Diagnostic Scanning along with the hardware will have the following functionality,

1. The connector plug, which is a piece of hardware, that acts as an interface between the application and the vehicle's computer system.
2. The 'Check Code' option is used to analyze the error code that's given by the vehicle's inbuilt computer system. The application displays the appropriate error message corresponding to the error code.
3. The 'Live Info' option is used to get the live running details of the vehicle when the device is plugged into the vehicle's computer system. Live data such as

Engine RPM, Coolant Temp, and Desired Idle, etc. are displayed in the mobile application.

4. The user will be notified whenever the error code database for a certain brand and model is updated or modified. The user will then have to click on the 'Update' option to update the application's database updated with the latest error code and the corresponding error message.
5. The user can save the Error Code that he had performed on a vehicle for future reference using the 'Save' button in the analysis screen.
6. The user as can pull the saved analysis report and when selecting the 'Saved Info' option in the applications menu needs it.
7. The 'TIP' menu is used to get safety tip of the day for proper maintenance of the vehicle.

4a Scenario Diagram(s)



4b Product Scenario List

1. *Connecting the application and the Vehicle's inbuilt computer:* The user can connect the vehicle's inbuilt computer and the application in the mobile phone using the AutoDiagScan connector which bridges the vehicle's computer and the mobile application.
2. *Launch AutoDiagScan Application:* The user can launch the application by clicking on the application icon in his/her mobile phone.
3. *Error Code Check:* The user can run an error code check in the vehicle to get the details of the problem with the vehicle.
4. *Solution for the Problem:* By clicking on 'FIX' button the user can get info about fixing the problem that are quite small and which doesn't need a mechanic to fix it.
5. *Fixing and Rescanning:* The user will have to rescan for error codes after fixing some problems with the vehicle to check if the problems are actually fixed or if they still exist.
6. *Saving the Error Code Report:* The generated error code report can be saved for future reference by clicking on the 'Save' button in the error code report screen of the application.
7. *Viewing Saved Error Code Report:* The user can pull out the saved reports by clicking on the 'Saved Info' option from the main menu.
8. *Checking the Live Running Status of the vehicle:* The user can get the vehicle's live running status when the application is connected to the vehicle's computer via the connector. The user will have to click on the 'Live Info' from the main menu.
9. *Updating the Error Code Database:* The user can choose to update the error code database by clicking on the 'Update' option in the main menu.
10. *Tip of the Day:* The user can view the Tip of the day by clicking on the 'Tip' option from the main menu of the application.

4c Individual Product Scenarios

Connecting the Application and the Vehicle's Inbuilt Computer

While driving back home in the evening from office, Adam notices the warning signal in his car's dashboard indicating that there would be a problem with the engine. Adam decides to check the problem soon after reaching

home. As soon as he reaches home he would pull out the connector and plug one end of it into the vehicle's inbuilt computer and the other end to his smart phone. Once the connection is properly done Adam can start using the application to trouble shoot his car.

Launch AutoDiagScan Application

Having made the connection between the phone and his vehicle's computer properly, Adam can now launch the application in his phone. For this Adam has to open the application list in his smart phone and search for the AutoDiagScan application and click on it. Soon after launching the application, Adam will land in the Main Menu of the application, which displays a list of things that can be performed using the application.

Error Code Check

Adam was notified in the dashboard that there is a problem with the vehicle, so Adam will first want to know about the problem(s). Having launched the application Adam will select 'Check Error Code' button to get the details of the problem in his vehicle. The application will take Adam to another screen where he has to enter his vehicle's Make, Model and Model Year. After hitting on the 'Go' button the application will get all the error codes from Adam's vehicle and look up the error code against the inbuilt database to get the problem description along with the severity of the problem. Once done, the application displays the problem description of all the error code that was sent out by the vehicle. The application also gives suggestions for fixing some minor issues.

Getting solution for minor problems

There are problems are quite common and doesn't need a mechanic to fix it. If the application encounters such a problem then it will show up a button 'FIX' along with the error description. Adam click's on this button and will land in Google's result page showing result for getting the problem solved without visiting a mechanic. Adam can choose a viable option amongst the ones that displayed to get the instructions on fixing the problem with his car.

Fixing and Rescanning

Adam can find the tips for fixing simple issues without visiting the Mechanic by clicking on the 'Fix' button, which will take him to various web pages that would give him step by step instruction on how to solve the problem. Once after fixing all the possible issues, Adam will press 'Fix and Rescan' to erase the issue codes that he had fixed and also to run the scan again to check for the ones that are still there and which needs a mechanic to get it fixed. If the error code that was fixed doesn't come up again then it means that the problem was fixed.

Saving the Error Code Report

Once Adam has completed fixing all possible issues, he will be left with the problems that need a mechanic to fix it. Adam can now choose to save the Diagnostic Report that the application has created, so that Adam can show it to the Mechanic. Adam will click on the 'Save' button in the report screen to save the generated report.

Viewing Saved Error Code Report

Adam can choose to view all the error code that was saved in his smart phone by the application by selecting the 'Saved Info' option from the Main Menu. This will list Adam with all report that was saved based on Time, Make, Model and Model Year. Adam can pull out different reports to compare the type of problems that his car has been getting over a period of time or he can pull the last report to show it to the mechanic to get them fixed.

Checking Live Running Status of the Vehicle

When phone is connected to the vehicle's computer system, Adam can monitor the live running status of his car by selecting the option 'Live Info' from the Main Menu. He will be displayed with a page that lists the basic status like Engine RPM, Battery, Intake Air Temperature, etc. Adam can use this for the periodic maintenance of his vehicle.

Updating the Error Code Database

The Error Code list differs between the Brand, Model and Model Year. Every time when a manufacturer launches a new model, the error codes of that model is published and the same has to be updated in the application. Whenever there is an update, Adam will be notified about it in the application. Adam has to select the 'Update' option in the Main Menu to get his application's database updated.

Tip of the Day

This option gives a maintenance tip every day. Adam should have his phone connected to the Internet to get a new tip every day. Adam has to click on the 'Tip' option from the main menu. The application will then display the tip of the day.

5 Stakeholders

5a The Client

The client for this software will be the makers/producers of automotive diagnostics devices (OBD Scanners). These manufacturers will be able to open the market to people outside of mechanics. Ease of use and helpful features will be the main selling point of the software. Large tool related companies (ex. TTI, Snap-On, Stanley Black and Decker) are candidates for this software.

5b The Customer

The customer is the buyer of the portable diagnostics tool. They will be able to download the software onto their device (smartphone/laptop). Different versions of the download will be available depending on the product key code given upon buying the hardware. The reason for different versions is because there will be a basic version as well as an advanced version. The basic version is aimed towards everyday people who own a car and know little or close to nothing about cars. The advanced version is geared towards enthusiast and mechanics that requires an abundance of detailed information for a proper diagnostics.

Once again this is aimed towards ease of use for any customer. Basic version customers will be able to see basic codes with tutorials of the method to fix such problem codes along with a list of directions to the closest auto repair shop if the severity level is high. The advanced version will provide not only basic codes but specific manufacturer codes which basic scanners do not have access to (recorded but not shown).

5c Hands-On Users of the Product

User name/category: Car Owners (basic user)

- *User role:* The app does not automatically tell the user there is an issue. Customer must know what the lights that show up on the dashboard of a car indicate (check engine light, tire pressure light, etc.). User starts the scan after a light has been indicated.
- *Subject matter experience:* Novice about cars. The basic user does not need to know what certain codes mean. A list of codes and summaries of what each code stands for will be available for the basic user. Tutorials for simple fixes will also be present for the user to view.
- *Technological experience:* Novice to Intermediate knowledge about smartphones. Basic users need to know the minimal workings of a smartphone. Help files can guide new users knowledge of the products features. Data can be transferred via NFC so an understanding on where the phone's NFC chip is located (transfer of resent code scans from user to mechanic) is expected.
- *User name/category:* Car Enthusiast and Mechanics (advance user)

- *User role:* The advance user should know the jargon and workings of a car. With this knowledge their user of the application is to aid them in fixing whatever problem exists and then scanning again to clarify the issue is solved.
- *Subject matter experience:* Skilled or Expert about cars. The advance user receives more than the basic user. The list of common codes and summaries still exist but advanced users should be able to know the jargon of the trade. What certain parts look like and how most of them work.
- *Technological experience:* Intermediate knowledge about smartphones and have used an ODB scanner before (helps to know where certain features will be located). As with the basic user, an understanding on where the phone's NFC chip is located (transfer of resent code scans from user to mechanic) is expected.

5d Priorities Assigned to Users

- Key users: Daily drivers, car owners, and local auto shop mechanics
- Secondary users: Dealership mechanics and car manufacturers
- unimportant users:

5e User Participation

To improve the quality of the software, feedback from both the basic and advance users are required. Feedback from basic users will further improve the ease of use and quality of information given. We do not want basic users to feel the need to know a mechanic's jargon. Feedback from advance users will cover the needs of mechanics. Additional submenus can be added if enough mechanics request for it. This will further advance the software with more detailed information (such as wiring diagrams for electrical diagnostics or drivetrain/belt diagrams).

5f Maintenance Users and Service Technicians

Since most of the content will be available in the program's initial download (there is no need to be online), there will not be a need for a team to maintain a database. An update team will be required when sufficient amount of changes are made to the software. The role of the update team is to sift through the feedback given by both the basic and advance users to determine additional features that will be added. There will also be a need of communication between the update team and legal specialists. There can be a lift on legal rights of certain manufacturer codes allowing them to be released in the following update.

5g Other Stakeholders

- Car Manufacturers – They are the building block for car specific information for the mechanics to work on. Since there are manufacturer specific codes, they are the ones we need to count on to provide the necessary data and information for the advance users.

- Testers – They are the initial people who will test the product for bugs. This will be a tough task, as they have to make sure certain codes do show up and end up displayed and well as clearing fault codes.
- OBD Scanner Manufacturers – They will provide information on the limitations of the hardware, which the software is to be made for.
- Legal experts – Rights are required to output manufacturer codes. If these legal rights are not met then certain manufacturer codes should not be placed on the list of read data.
- Usability experts – Provide information on the usability and functionality of the design through surveys.

6 Mandated Constraints

6a Solution Constraints

1. Description: The product will be supported on multiple platforms

Rationale: The client will not want to buy into a specific platform to use the software. There should not be a need to carry multiple devices.

Fit criterion: The product should run on mobile devices (iOS / Android / Windows Phone).

2. Description: The product will have options for transferring scanned data.

Rationale: The client will not want to buy a new device, which supports NFC for quick transfers. Fit criterion: The product should have an option to either email/text or use NFC to share the list of old scan data.

6b Implementation Environment of the Current System

User buys the product. Within the product is a key used to download the application (basic or advance). The device now has the application. A connection between the hardware and the device, which holds the software, is to be made. This connection is made by hooking up the hardware device to the phones data port (micro USB port or apple dock/lightning port). When the device detects a stable connection, diagnostic scanning and live scanning is allowed. Users will be prompted with updates when new version are made.

Option of transferring data through 2 phone devices is present with the use of NFC is any saved scan data exists. This is often use to transfer data from a car owner to a mechanic. This relaying of data is confirmation of the diagnostic between car owner and mechanic.

6c Partner or Collaborative Applications

Collaboration with Google Maps will be required for the map feature. The map feature helps basic users find the closest auto shop. Depending on how useful and big the device becomes, further collaboration with Google may add an icon to certain shops that use the same application (allows data sharing).

Collaborations with streaming sites, which host tutorials for basic knowledge for simple troubleshooting, will be required.

6d Off-the-Shelf Software

No outside software is required to run this application. A browser application is required to access the links when a fix is required. This browser should come with whatever device the user already has.

6e Anticipated Workplace Environment

The mechanic's hands might be tied and/or dirty however they wish to acquire information. Allow keywords to activate speech to text for searching on the advance user application.

The implication that basic users have no knowledge of cars should be taken into consideration when applying ease of use (tutorials and help files).

If the workplace is loud the user should wait before continuing to use any audio related option of the software (ex. pausing a tutorial video, using voice commands).

The workplace environment can be dark or bright. Users should be able to adjust the brightness on their devices.

6f Schedule Constraints

The completion of the software should be done before the holiday seasons. The sooner the completion is, the better it will be for the manufacturer of the hardware. It is the best interest to allow the manufacturer's advertising and marketing sections to promote the hardware. Thus boosting the sales of the hardware.

6g Budget Constraints

The budget is calculated based on the different types of people working in it. We will have a set of architects who would design the basic architecture of the system followed by a bunch of developers who are going to implement the design into a working product.

The architects will be designing the system for a month followed by the developers coding the system for 5 months. Its going to be a 6 month project costing \$100000 for 2000 man hours at \$50 per hour.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

ADS: AutoDiagScan—name of the application

OBD: On Board Diagnostic

RPM: Revolutions per Minute

NFC: Near Field Communication

USB: Universal Serial Bus

7b UML and Other Notation Used in This Document

7c Data Dictionary for Any Included Models

8 Relevant Facts and Assumptions

8a Facts

The app should work in all mobile platforms including iOS, Android, and Windows Phone. The user interface and features of the application should look and work alike across all these platforms.

8b Assumptions

Users should have no issues in installing the application.

The users should have Internet connection in order to use all features of the application.

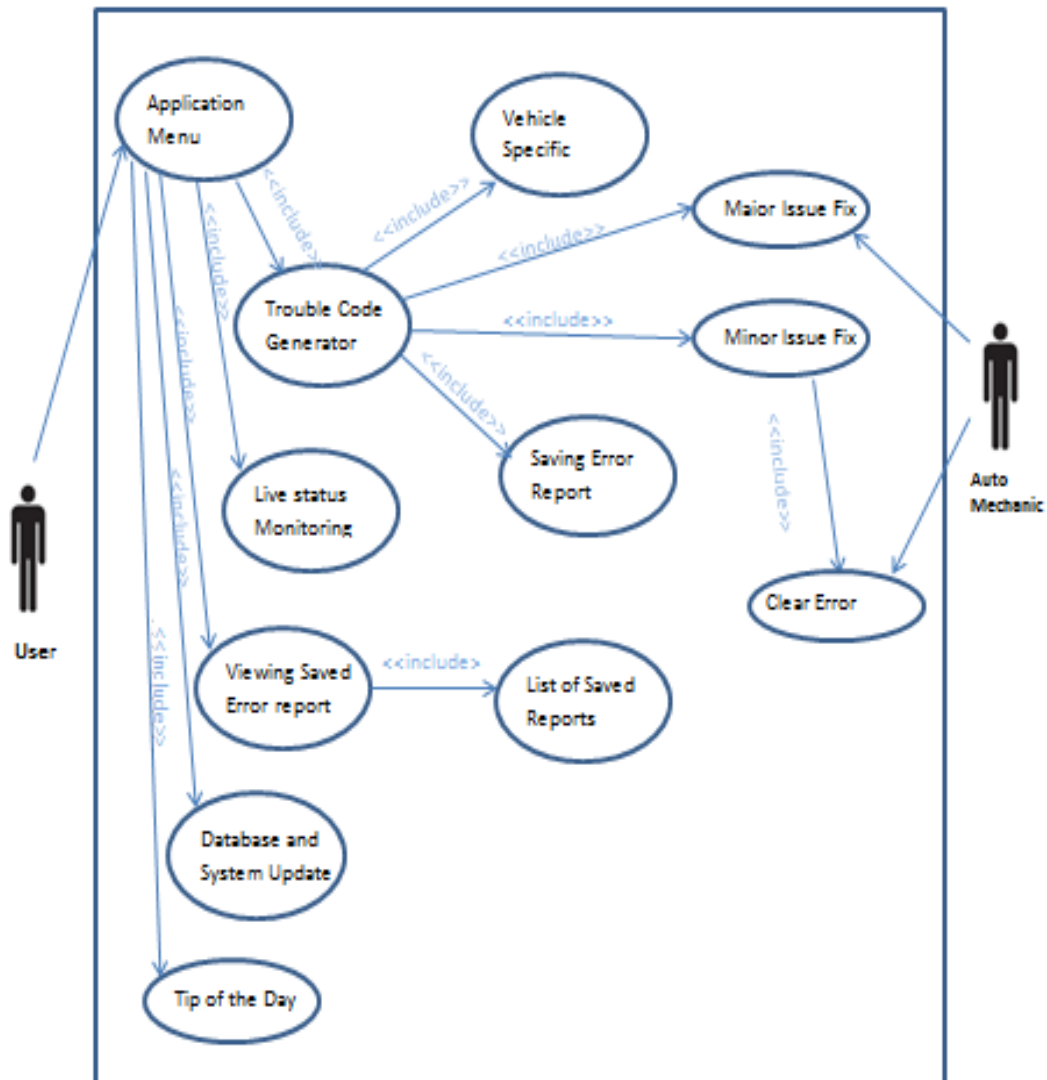
There should be a prototype of the hardware add-on available prior to the beginning of the development.

The engine diagnostic fault trouble code database will be updated daily to keep the application up to date.

II Requirements

9 Product Use Cases

9a Use Case Diagrams



9b Product Use Case List

Below is the list of use cases that are further developed in the next section.

- 1 Trouble Code Generator
- 2 Troubleshooting (Minor)
- 3 Troubleshooting (Major)

- 4 Saving error code report
- 5 View report history
- 6 Live status of the vehicle
- 7 Update error code database
- 8 Tip of the day
- 9 Clear error code

9c Individual Product Use Cases

Use case name	Trouble Code Generator
Participating actors	Initiated by the user
Flow of events	<ol style="list-style-type: none"> 1. User opens the application from the menu tray of his phone 2. The application opens up and the menu screen is displayed to the user 3. The user selects the “Scan” option from the menu to launch a full error code scan of his car 4. The application prompts the user for the details like Brand, Model, Year and Engine Type. 5. The user enters the required information and presses the “Ok” button. 6. The application starts the error scanning through the ODB2 connector and displays the list of errors with their severity on screen to the user. 7. After viewing the error list the user closes the application. <ol style="list-style-type: none"> 7.1 The user may choose to save the error report and then close the application 7.2 The user may attempt to fix simple error before closing the application
Entry condition	User starts the error code scan on seeing the warning alert in his vehicles dash board.
Exit condition	User will be listed with errors that are read from the vehicles inbuilt computer system. The severities of the errors are also listed along with it.
Quality requirements	The user must be listed with the exact set of errors that are produced in the vehicles inbuilt computer system. The severity provided must be correct.
Use case name	Troubleshooting (Minor)

Participating actors	User (primary) Browser
Flow of events	<ol style="list-style-type: none"> 1. User gets a list of Error Codes after a scan 2. User presses “Fix” Button located next to a specific Error <ol style="list-style-type: none"> 2.1. System determines difficulty level of Error to display relevant information 2.2. System recognizes that the difficulty level is minor 2.3. System opens Browser 2.4. Browser displays information regarding how to fix the specific Error (video & text links) 3. User returns to Error Code list and may view another Fix if requested.
Entry condition	After a scan has been completed the User presses the “Fix” button found on the side of an Error Code
Exit condition	User is done viewing the required information and decides to go back to Error List.
Quality requirements	The user will not be able to access other functionality while Browser is open with troubleshooting information. Other functionality will be given upon going back to the Error Code List.
Use case name	Troubleshooting (Major)
Participating actors	User (primary) Google Maps
Flow of events	<ol style="list-style-type: none"> 1. User gets a list of Error Codes after a scan 2. User presses “Fix” Button located next to a specific Error 3. System determines difficulty level of Error to display relevant information 4. System recognizes that the difficulty level is major 5. System checks to see if GPS/Location is turned on <ol style="list-style-type: none"> 5.1 If GPS is not turned on, System tells the user of possible location inconsistency 5.2 If GPS is on, System opens Google Maps 6. System hands Google Maps current location and searches for the closest Auto Repair Shop 7. User returns to Error Code list and may view another Fix if requested.
Entry condition	After a scan has been completed the User presses the “Fix” button found on the side of an Error Code
Exit condition	User is done viewing the required information and decides to go back to Error List.
Quality requirements	The user will not be able to access other functionality while Google Maps is open with closest shop information. Other functionality will be given upon returning to the Error Code List.
Use case name	Saving Error Code Report

Participating actors	User (primary) Memory/Application's data log
Flow of events	<ol style="list-style-type: none"> 4. User gets a list of Error Codes after a scan 5. User presses the "Save" Button located at the bottom bar below the Error Code List <ol style="list-style-type: none"> 5.1. System saves the current list of Error Codes into a log file located somewhere specific in memory. Metadata such as Year, Make, Model, Date and Time are also saved. 5.2. System displays a message box indicating that the Error Code List has been saved/logged. 6. User returns to Error Code list and proceed with further application functionality at choice of the user.
Entry condition	After a scan has been completed, the User presses the "Save" button.
Exit condition	System is done saving the current Error Code List and returns to the Error List screen.
Quality requirements	The user will not be able to access other functionality while the saving of data is undergoing. The lock is released after necessary data has been saved.

Use case name	View Report History
Participating actors	Initiated by the user
Flow of events	<ol style="list-style-type: none"> 1. User opens the application from the menu tray of his phone 2. The application opens up and the menu screen is displayed to the user 3. The user selects the "Saved Info" option from the menu to get the history of error report 4. The application lists all the error reports that were saved in the application. The list is ordered based on the Model and Make and Model Year. 5. The user selects a report from the list of error report that's listed by the application. 6. The application pulls all the error details that were saved corresponding to the user selection and displays the same. 7. After viewing the saved error list the user closes the application. <ol style="list-style-type: none"> 7.1 The user may choose to carry on other tasks in the application before closing it.
Entry condition	User views the Error report history to check if there is a recurring problem with his vehicle or if he wants to show the last scan to the Mechanic.
Exit condition	User will be listed with the errors that were saved in the selected scan.

Quality requirements	The errors that are listed in the saved report must be exact copy of what was viewed when the scan was actually run on the vehicle.
----------------------	---

Use case name	Live status of the vehicle
Participating actors	Initiated by User
Flow of events	<ol style="list-style-type: none"> 1. User presses the Live Info button 2. ADS starts reading from the built-in computer of the vehicle for the live status of the vehicle 3. After done reading, ADS displays a list of live information of the vehicle
Entry condition	The phone must be connected to the built-in computer via ODB connector
Exit condition	User can see the live information of the vehicle
Quality requirements	ADS should not take more than 1 minute to read and display the live status of the vehicle

Use case name	Update error code database
Participating actors	Initiated by the system
Flow of events	<ol style="list-style-type: none"> 1. User launch the app with an Internet connection 2. ADS detects that there is a new code update available 3. ADS automatically updates its error code database 4. ADS displays a notification saying that the update is successful.
Entry condition	The phone must be connected to the Internet, and there is a new code update.
Exit condition	The code database is up to date.
Quality requirements	ADS display a message saying that the update was successful, and it should not take more than 5 minutes to update its database.

Use case name	Tip of the day
Participating actors	Initiated by User
Flow of events	<ol style="list-style-type: none"> 1. User presses the Tip of the day button 2. ADS displays a randomized tip
Entry condition	The app is installed and works correctly
Exit condition	User can see the randomized tip
Quality requirements	ADS should not take more than 10 seconds to display a random tip

Use case name	Clear Error Code
---------------	------------------

Participating actors	User (primary) Car's Computer
Flow of events	<ol style="list-style-type: none"> 1. User gets a list of Error Codes after a scan 2. User presses the "Clear" Button located at the bottom bar below the Error Code List <ol style="list-style-type: none"> 2.1 System "saves" current error report before communicating with the car's computer to clear errors. <ol style="list-style-type: none"> 2.1.1 Car's computer resets the current list of lights/flags that were triggering the errors. 2.2 System displays a message box indicating that the Error Code List has been cleared and issues a rescan. 2.3 System starts a scan once again. 3. User returns to Error Code list and proceed with further application functionality at choice of the user.
Entry condition	After a scan has been completed, the User presses the "Clear" button.
Exit condition	System is done clearing the current Error Code List and returns to the Error List screen.
Quality requirements	The user will not be able to access other functionality while the errors/flags are being cleared by the car's computer. A copy of the error report must be saved before clearing is to be done. A rescan must follow after errors have been cleared. The lock is released after the scan has been done.

10 Functional Requirements

Requirement # 1

Description: The application must recognize that a connection is made when it's connected to the vehicles computer system with the ODB connector.

Rationale: This would confirm the user that the application had been successfully connected to the vehicle's computer system.

Fit Criterion: The application having successfully connected to the vehicle's computer system, shows up an icon indicating the connection.

Requirement # 2

Description: The application must compile a list of error codes along with the corresponding error messages and severity after completing the scan.

Rationale: This would let the user know about the errors that's in their vehicle.

Fit Criterion: The application will display the list of error codes, messages and the corresponding severity soon after it completes the scan on the vehicle's computer system.

Requirement # 3

Description: When connected the application must display data periodically corresponding to the sensors.

Rationale: This lets the user to monitor the performance of the vehicle with live data from the sensors being displayed on screen.

Fit Criterion: A menu labelled "Live Info" will let the users to monitor the live info of the vehicle from various sensors in the vehicle.

Requirement # 4

Description: The system must force an update when new versions of Error Codes are released.

Rationale: Updating the application with the latest error codes helps it to better scanning and in identifying the corresponding error messages.

Fit Criterion: The application must notify the user about the update for it, and if the user chooses to update the application then the user can do so by clicking on the 'Update' button in the menu screen.

Requirement # 5

Description: The application should show instructions for minor issues that can be fixed by the owner.

Rationale: This would let the user to fix minor problems themselves, which they would have to take it to a mechanic otherwise.

Fit Criterion: The screen that displays the error list will include a fix button for those issues that are identified to be minor issues by the application. Clicking the on the button will take the user to the instructions page for getting the issue fixed.

Requirement # 6

Description: The application must show a list of Auto Repair shops close by the user's location.

Rationale: This lets the user to get the vehicle to the nearest auto repair shop to get the major issues fixed.

Fit Criterion: The system will include a button in the error list screen if there happens to be a major issue which requires a mechanics attention. The user will have to click on the button to get the list of Auto Repair shops that's close to his/her location.

Requirement # 7

Description: The application must let the user to save the scanned error report.

Rationale: This would let the user to view the error report at a later point in time and also to identify recurring problems based on the history of error reports.

Fit Criterion: the system will include a button in the error list screen, clicking on it will save the current report into the applications database for future reference.

Requirement # 8

Description: The application must present the user with a list of saved error reports to choose from, for viewing.

Rationale: This would enable the user to view the error list that was generated in the previous scans.

Fit Criterion: When the user selects the "Saved Info" option from the application menu, the list of saved error reports will be presented for the user to choose from.

Requirement # 9

Description: The application must display a general tip every day in such a way that it doesn't intrude the users' normal activity.

Rationale: The users who are interested in daily tips for car maintenance can get them from the application and for the rest it wouldn't show up letting them carry on with their usual tasks.

Fit Criterion: The application menu will include a "Tip" button, clicking on which will display the tip of the day for proper car maintenance.

Requirement # 10

Description: The user must be able to provide information like Model, Model Year, Make and Engine Type before proceeding with the scan.

Rationale: This set of information will enable the application to adjust itself to scan the vehicle and to compile the error list faster with the corresponding error messages specific to the selected vehicle.

Fit Criterion: When the user selects the “Scan” option from the menu list, the application must get the required vehicle information from the user before starting to scan the vehicle.

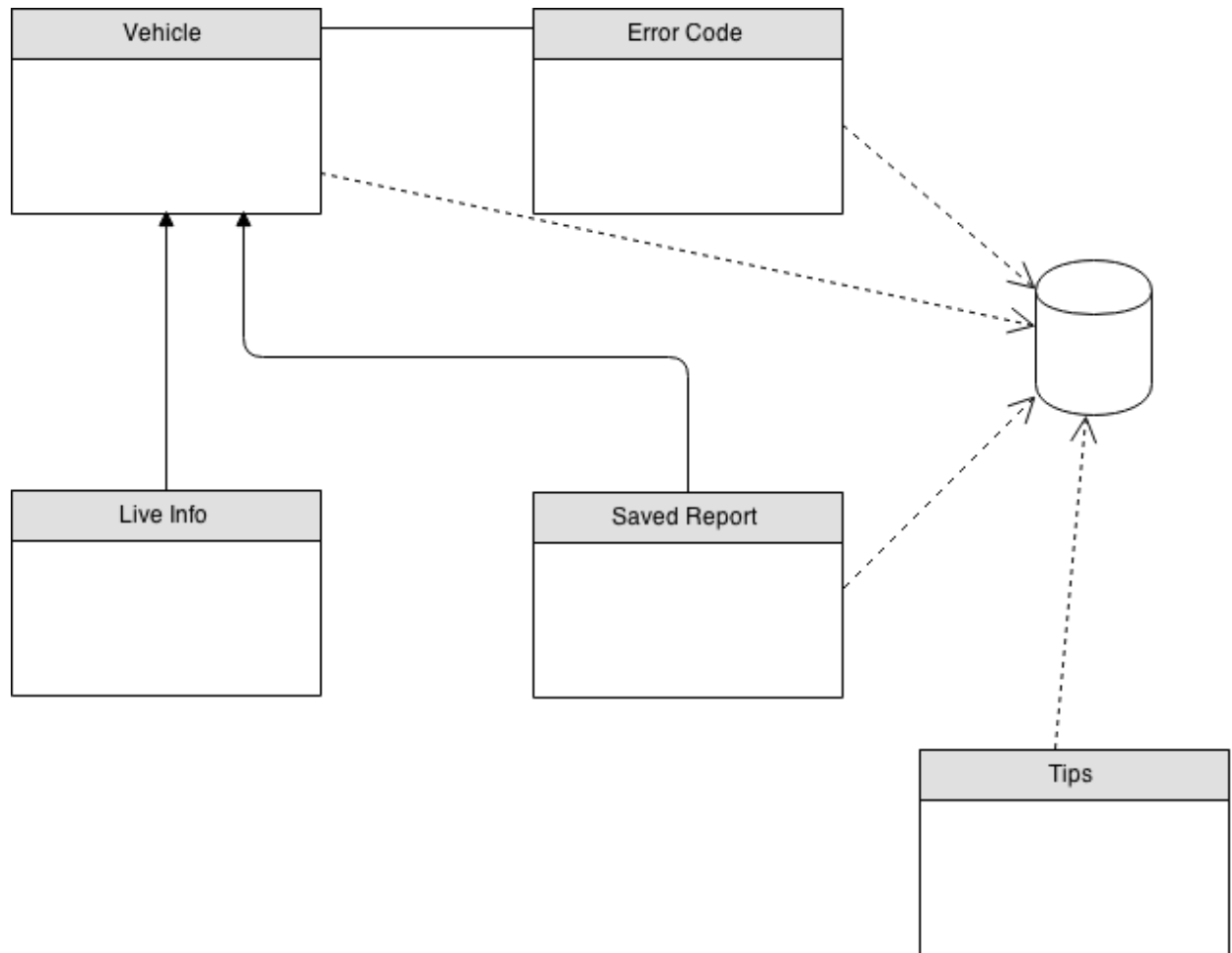
Requirement # 11

Description: The application must be able to clear error codes which have been identified by the vehicles inbuilt computer system after the user fixes them.

Rationale: Clearing the error codes will stop the flashing of Engine Alert sign in the Vehicle’s dashboard.

Fit Criterion: The application will have a fix button in the error list screen corresponding to each error code that can be fixed by the users themselves. On fixing the error the user will have to press this button to get the error code cleared from the Vehicle’s computer system.

11 Data Requirements



12 Performance Requirements

12a Speed and Latency Requirements

- The application should be loaded in 3 seconds or less after the user open the application on any mobile phone device or tablet.
- If the user completes a scan, the application shall load and display the list of all generated error codes within 2 seconds.
- When connected, the application shall display the live status data of the vehicle on mobile phone device or tablet screen less than 10 seconds.
- The response time of the application must not be longer than 3 seconds.
- Overall, the application must feel very responsive.

- The game shall respond in less than 2 seconds for 85 percent of the interrogations.
- The application response must be fast enough to avoid interrupting the user during use or scanning.
- At any point if request takes longer than 3 seconds, the application must display a graphic similar to the graphic below as to indicate that the application is still processing the request.



12b Precision or Accuracy Requirements

- The user shall be able to save all trouble code generated by the application and view them when necessary.
- Trouble or error codes generated by the application must be as accurate as the code generated by a conventional OBD scanner.
- Live status data displayed by the application must be as accurate as the one displayed by a conventional OBD scanner.

12c Capacity Requirements

- The application must be able to store a total of 100,000-trouble or error code it generated for users.
- The application database must able to store all error code for all make and model of vehicle made from 1996.

13 Dependability Requirements

13a Reliability Requirements

- In case the mobile phone device or tablet loss power and shuts down, the trouble or error code generated by the application must be saved.
- The application must not have glitches or crashes more than once every week.
- In case the application is on maintenances, users of the application must be able to scan and ready error code from the application.
- Should the application fail, it should only take a few seconds for it to reload and get all the information or error code form the scan in progress.
- Users data must not be lost or damage while error code is been saved.
- The must be able to retrieve all error code scan history of any particular vehicle.
- All live status data of vehicle must be save in case the Internet service is out during scans.

13b Availability Requirements

- All users must be able to access to application at all times.
- The application new updates such as bugs fixes must be made available every month.
- New updates of application must be available every year when auto manufacturers are released new yearly model of vehicle.
- The application must be functioning properly for 99% of the time.

13c Robustness or Fault-Tolerance Requirements

- In the event of a lost in connection with the database during a scan, the application will quickly notify the user.
- Application will scan and must allow user to access database for error code for at last four minutes during a lost in connection with the database.
- During database connection lost, application must be able to display live status data of a scan.

13d Safety-Critical Requirements

- The application hardware shall not cause any direct physical damage to the users.
- The application shall not display changing or flashing light to cause damage to eyes of the user.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

- Maintenance updates will be made for the application quarterly every year.
- All shall be maintained by developer who are not the creators of the application
- Only the developer will have access to the source code since users will not be required to maintain the application (software).
- Users must have the option to report any bugs or email the application support term for any assistances.
- The bugs reported by the users will be developed and tested two weeks before the release date for the updated version of the application.

14b Supportability Requirements

- Initially when the application is been lunched, the maintenance term will be in charge of support.
- Help desk must be available by telephone from Monday to Friday 6:00am to 6:00pm.
- During weekends and non business hours, support will be provided through email from users.

14c Adaptability Requirements

- Upon the first release, the application must be compatible with android, blackberry, window and iPhone mobile devices.
- Upon the first release, the application shall include support for English only, since it is been released in the United States. Every quarter of the year, the application must incorporate a new language. Examples of languages are Spanish, Russian and so on, so that it can be used internationally.

- New and updated version of the application must be back compatible with older mobile devices.

14d Scalability or Extensibility Requirements

- Upon release, the application shall be able to 100,000 users in the first month of release.
- After the first year, the application will be able to support 1,000,000 users here in the United States.

14e Longevity Requirements

- The features of the application shall continue to improve over time. Any bug that been discovered shall be removed and better features shall be added in order to improve user experience.
- Application shall be expected to operate within the maximum maintenance budget.

15 Security Requirements

15a Access Requirements

- Access to source code and data of application shall be granted to only developers and the testers managers.
- Database for all error code of the application must only be accessible to managers in the maintenance team.
- All new-updated version of application must release only with the authorization of the owners of the company.

15b Integrity Requirements

- All data in the users database shall not be modified in any form or shape. Users must not have access to this data at any time.
- All user data and error code must be stored confidential and protected.
- The application itself must be protected from hacker or non-users.
- Application maintenance team shall conduct a daily check to ensure source code and database is not been manipulated.

15c Privacy Requirements

- Privacy policy shall be made available to new user on the main page of the application.
- The users must get notified immediately if any changes are made to the privacy policy of the application.
- The application shall protect private information in accordance with the relevant privacy laws of the state.
- The application shall send emails to the user notifying them of the company's information policy.

15d Audit Requirements

- User private information shall be kept on application database for six years.
- In accordance to the state laws, audit checks will require the company maintain users data for sometime.

15e Immunity Requirements

- All information sent to the database shall be protected.
- Application will be scanned periodically for any malicious software or spyware that would be dangerous for business. It will be removed right away.
- Application shall be designed in a way to be protected from any unforeseen problems.

16 Usability and Humanity Requirements

16a Ease of Use Requirements

- The user is expected to know how to read English.
- The application should be usable for users with little to no knowledge and users that have never used the app before.
- The app shall be appropriate for any age.

16b Personalization and Internationalization Requirements

- The app shall be available in English.

16c Learning Requirements

- The learning curve should be very low.

- The user is expected to possess basic knowledge of vehicles.

16d Understandability and Politeness Requirements

- Any videos or images used by the application shall not be offensive to any user.

16e Accessibility Requirements

- The app shall be accessible by all individuals with common disability which does not restrict them from operating a phone.
- The app shall avoid using red or green colors to make it easier for people with red-green colorblind.
- The app shall be further developed to support people with disabilities.

16f User Documentation Requirements

- The app shall include a user manual in electronic format. All features of the app shall be documented.

16g Training Requirements

- The app shall be easy to use that no trainings are needed.

17 Look and Feel Requirements

17a Appearance Requirements

- The app's user interface shall be both professional and simple.
- The app shall use visually appealing colors.

17b Style Requirements

- The Main Menu of the app shall look professional and avoid clustering a lot of materials together.

18 Operational and Environmental Requirements

18a Expected Physical Environment

- To use the app, the user must have access to a smartphone running on iOS, Android, or Window Phones.
- The app shall need internet access for fully functioning.
- The app shall not be loud in the environment.

18b Requirements for Interfacing with Adjacent Systems

- The app shall need internet access in order to get new updates of the app as well as new updates of the code database.
- The app shall support old and new versions of smartphones that meet all the requirements of the app.

18c Productization Requirements

- The app shall be distributed on the app market of each Mobile platform that it supports (Apple App Store, Google Play Store, and Microsoft Windows Store).
- The app shall be able to be installed by an untrained user.

18d Release Requirements

- There shall be at least one major release per year. Each major release focuses on introducing new features, enhancing existing features, and improving visual features.
- Maintenance releases that shall be offered whenever bugs reported during the period are fixed.
- There shall be database update releases whenever a new versions of the OBD2 database is released.
- All of the existing features of the app shall not fail on any of the new releases
- Maintenance team shall fix any bugs reported within 10 days the bug is identified.

19 Cultural and Political Requirements

19a Cultural Requirements

- The app shall not be offensive to religious or ethnic groups.
- The app shall use American English for all the spellings and numeric formats.
- The app shall not use any terms, images, or videos that offend individuals or any particular culture.

19b Political Requirements

- The app shall adhere by all laws laid out by the constitution of the United States of America.

20 Legal Requirements

20a Compliance Requirements

- Both the application and the new ODB 2 connector shall avoid all copyright violations.
- The application will be available for free online in Google store for android users and in the Apple Store for IOS users. This application shall not be redistributed by independent individuals and companies seeking personal gains from it.
- The ODB2 will be made available at all the Leading Mobile Accessory store. This connector shall not be duplicated or redistributed by independent individuals or companies seeking personal gains from it.
- This application will not read or transfer any of the information in the installed device. So there are no confidentiality requirements for this application.
- In order to avoid Intellectual Property lawsuits, developer who has seen code by direct competitors will not be allowed to work on this application. The same holds for the Engineers who design the connector for the application.

20b Standards Requirements

- There is an appointed Standard Keeper to verify the standards met by both the application and the connector.
- Both the application and the connector should obtain approval from the Standard Keeper before every release. Failing to do so, will delay the release of a particular version.
- The application will be developed using the Agile Development Process.
- The connector will be designed and manufactured following the conventional manufacturing process.

III Design

21 System Design

21a Design goals

The design goals represent the desired qualities of the application AutoDiagScan and provide a consistent set of criteria that must be considered when making design decisions. Design of the application will put simplicity or user friendly design. It should be the easy to use by any user with looking at the help menu. The main objective is to design robust, maintainable and reusable software with Object-Oriented analysis and design.

The design goals identified in details are as follows:

- **Adaptability:** Java will be one of the language uses for creating the application. This is because of the portability of java; it is a cross platform language.
- **Efficiency:** The application is going to be responsive and able to run with high performance. The application must response to search in less than 3 seconds displaying all the error code.
- **Reliability:** System will be bug-free and consistent in the boundary conditions. The system should not crash. To achieve this goal, the testing procedures will continue simultaneously with each stage of the development. Besides, boundary conditions will be evaluated very carefully not to miss any unconsidered situation, which may crash the system.
- **Usability:** The easier it is to use, the better. This is important design goal for this application. This will attracts a lot of user it is easy to user. The design will make it easy for first time users to use.

22 Current Software Architecture

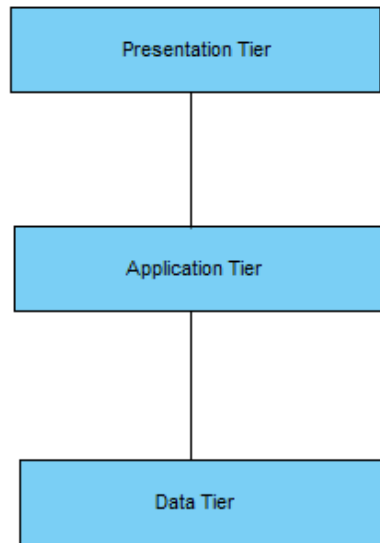
This is a new and a unique application is being designed and developed. There is no application out there that has the same concept of the AutoDiagScan. Therefore, there is no prior architecture that can be used to compare the app.

23 Proposed Software Architecture

23aOverview

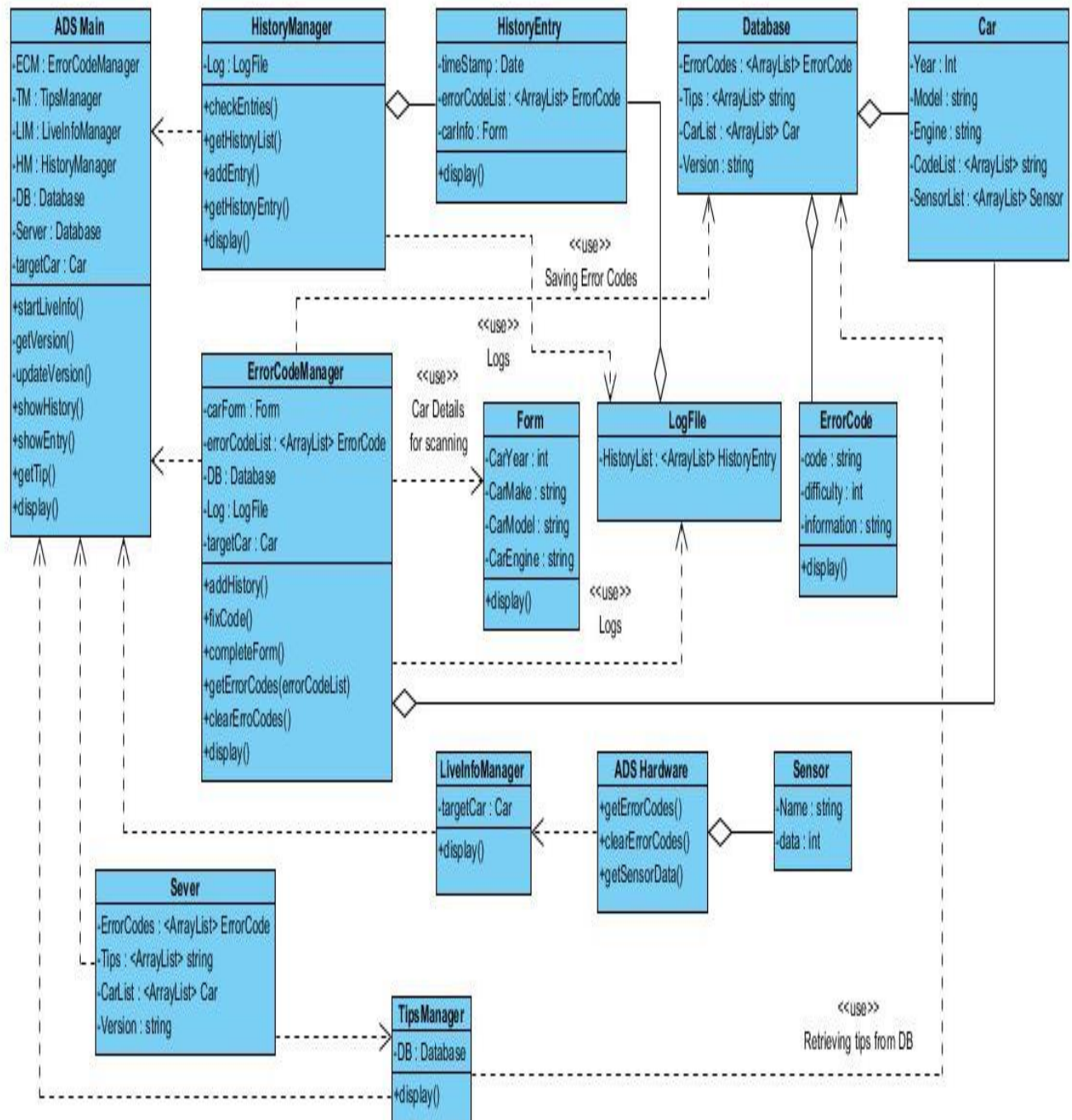
The application is designed based on the three tier architecture model. Starting with the presentation tier which presents data to the user and captures user's commands. The action associated with the users command are delegated to the application tier which takes care of the logic and data processing along with the interaction with the vehicles computer system. Finally the data tier is used for storing all the data into the applications inbuilt database.

Breaking up the tiers based on task, all the screens and buttons for user interaction will fall under the presentation tier. All the major activities which involves interacting with the vehicles computer system are handled by the application tier. All the error codes and saved error report are taken care by the database tier.



23b Class Diagrams

The class diagram below shows the relationship between various classes that are used in the application along with the various design patterns that's implemented in achieving the design.

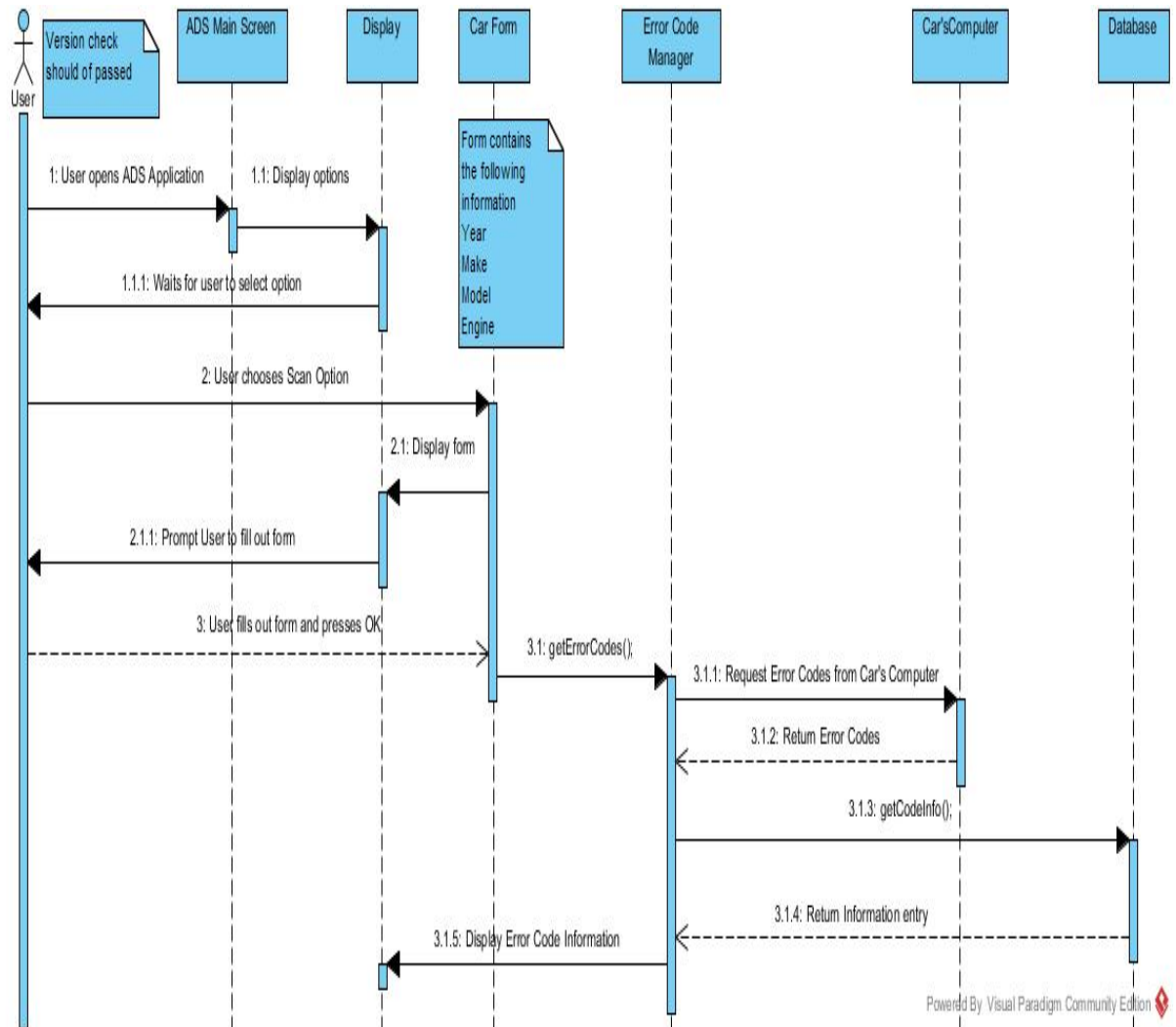


23cDynamic Model

The sequence diagrams that are to follow will illustrate the steps in performing the corresponding activity in the application.

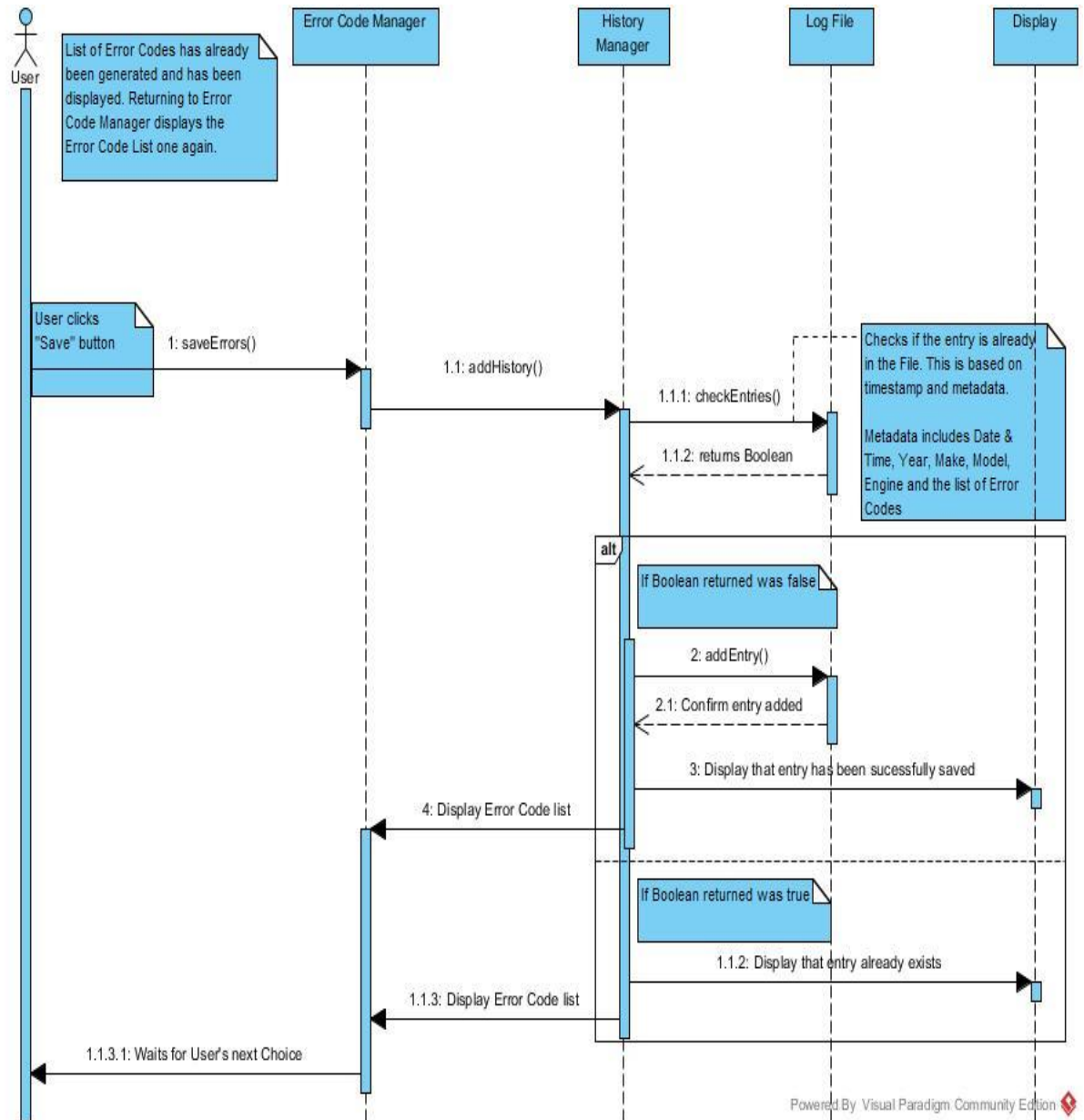
Trouble Code Generator

This is the most vital functionality of the application. This involves the steps involved in using the application to scan for error codes in the vehicle built-in computer system.



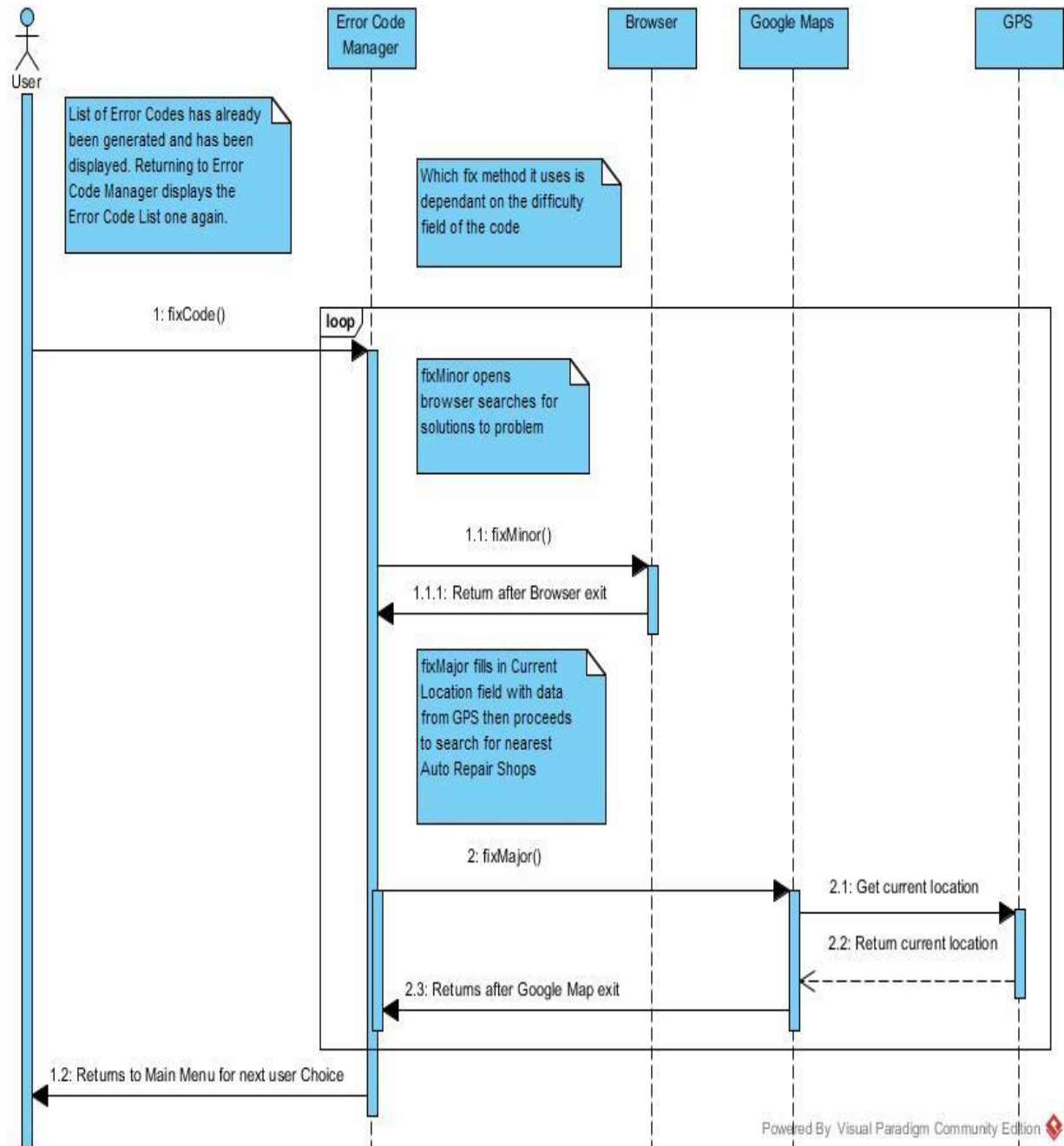
Saving Error Code

This involves the steps in saving the generated error code into the application database for future reference



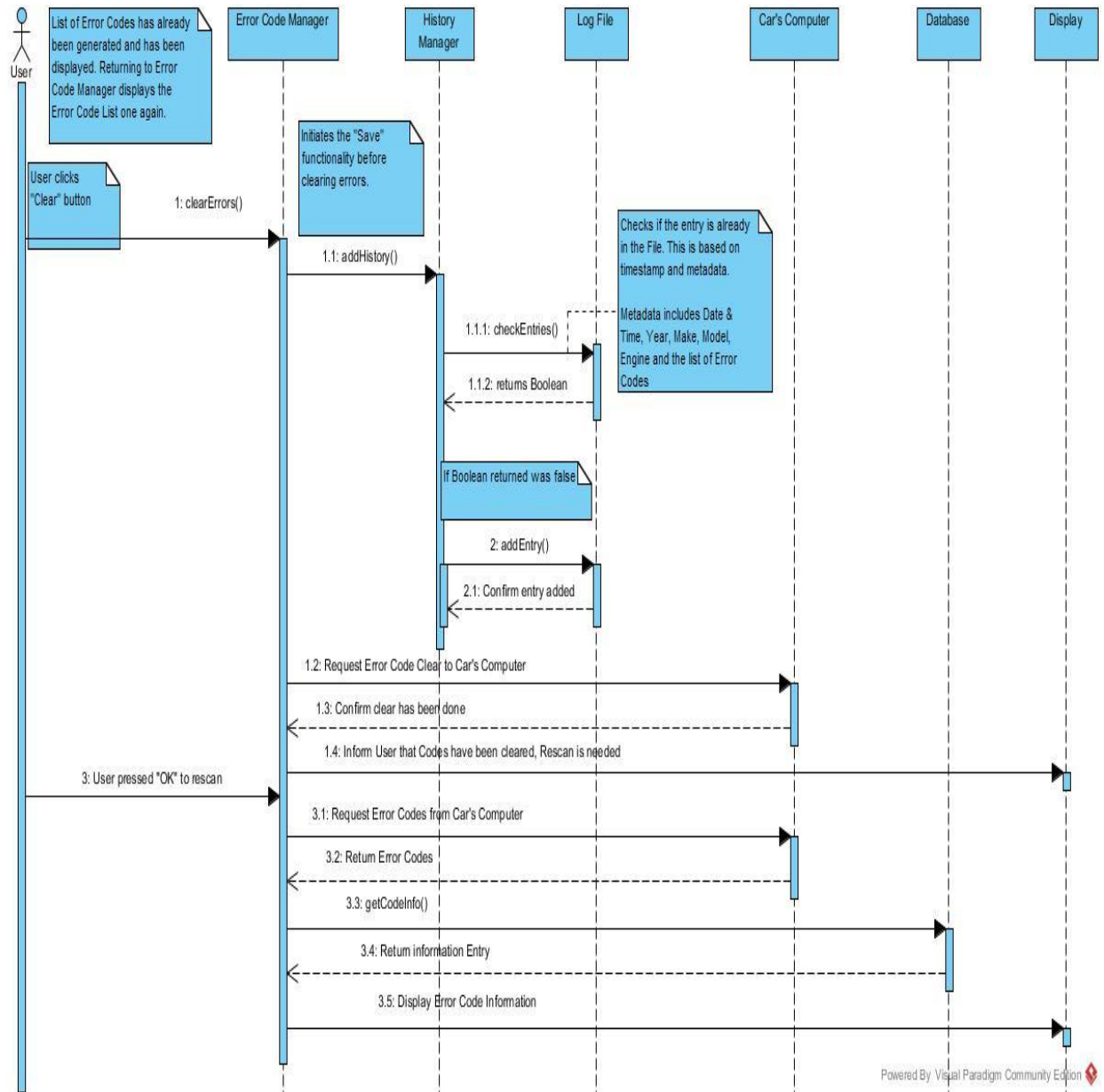
Troubleshooting

This shows the steps involved in the troubleshooting the vehicle based on the generated error codes.



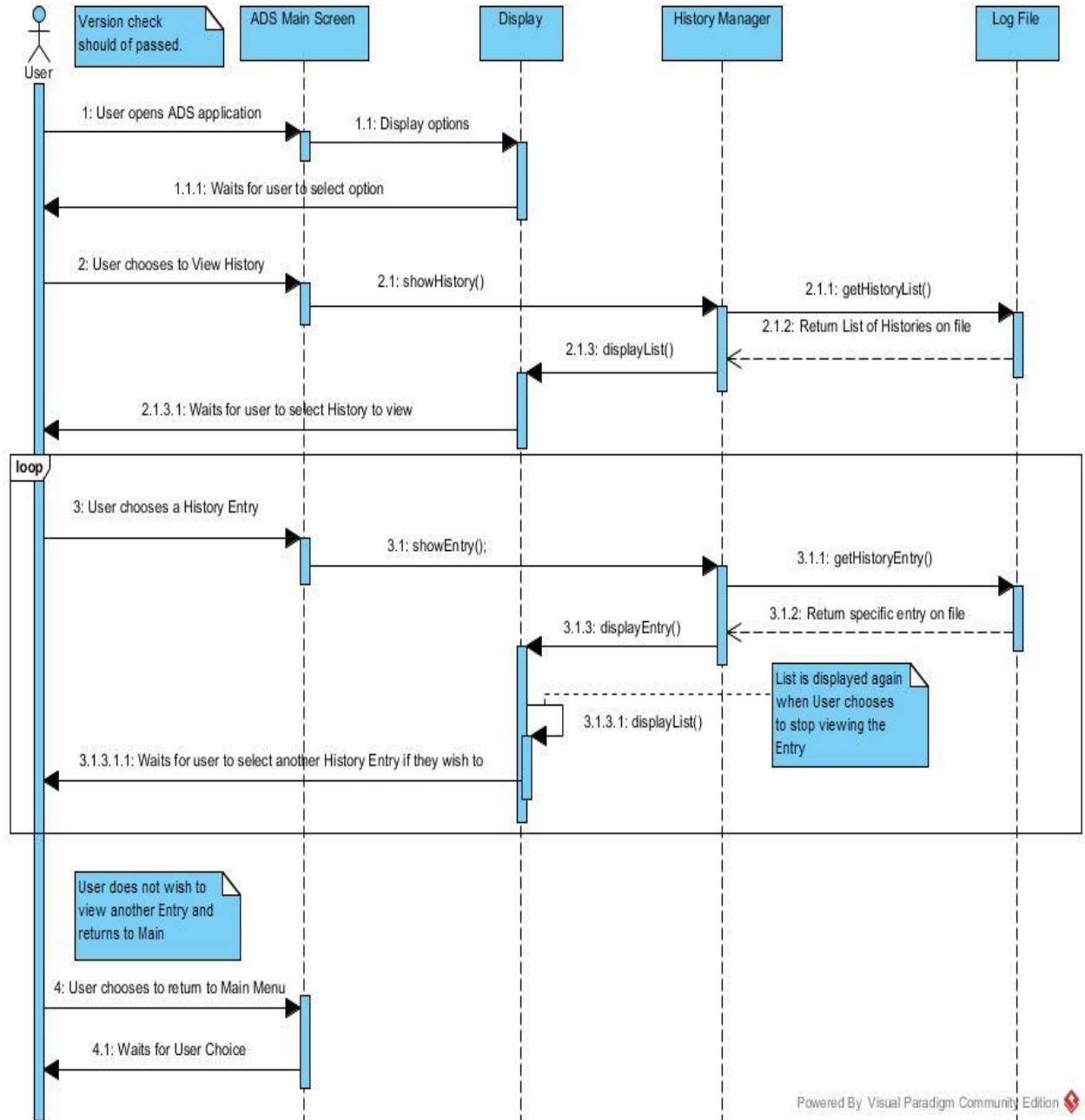
Clear Error Codes

This shows the steps involved in clearing the error codes from the vehicles inbuilt computer system soon after fixing minor issues.



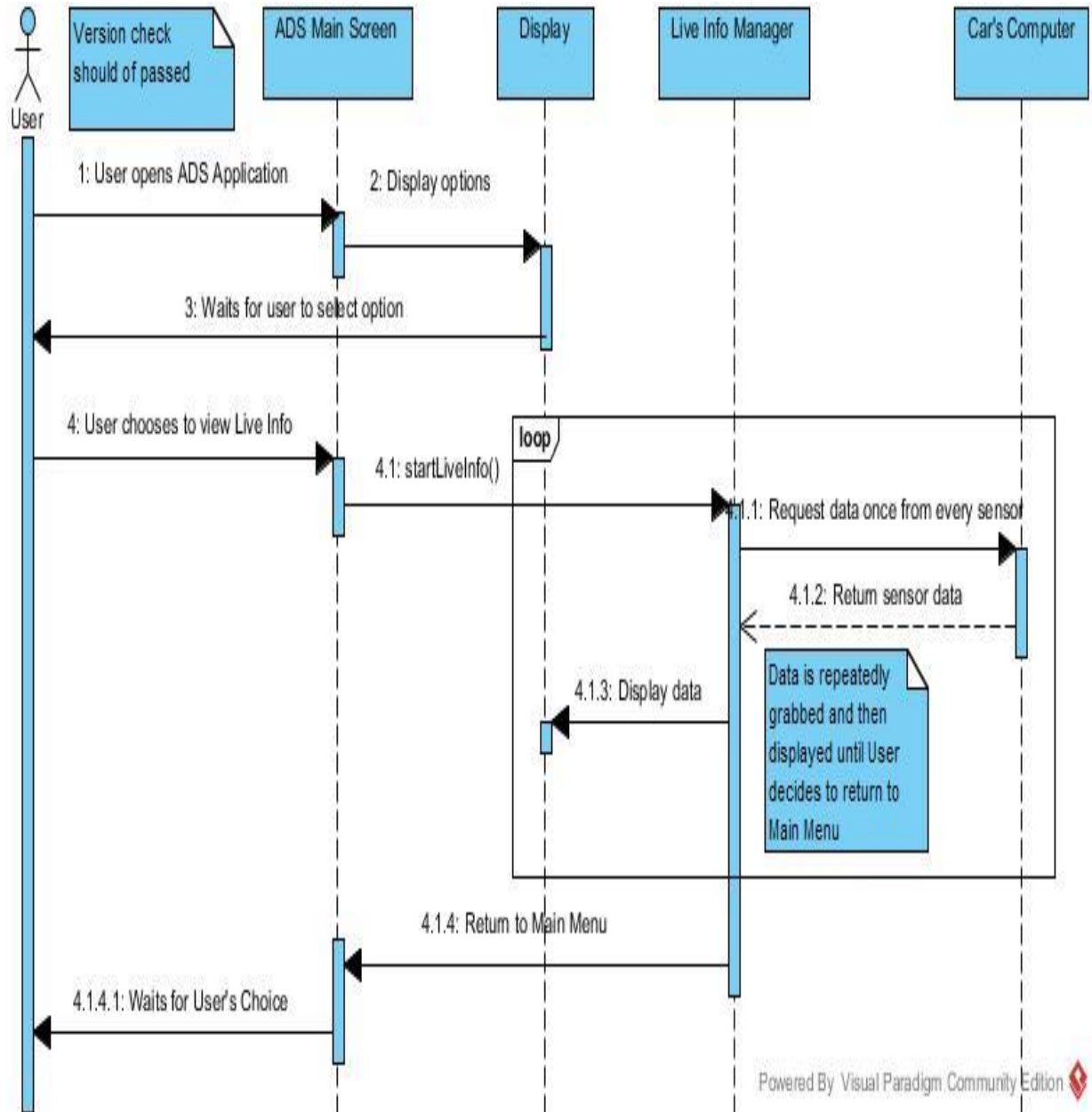
View Report History

The below diagram details the steps involved in retrieving a saved report from the applications inbuilt database



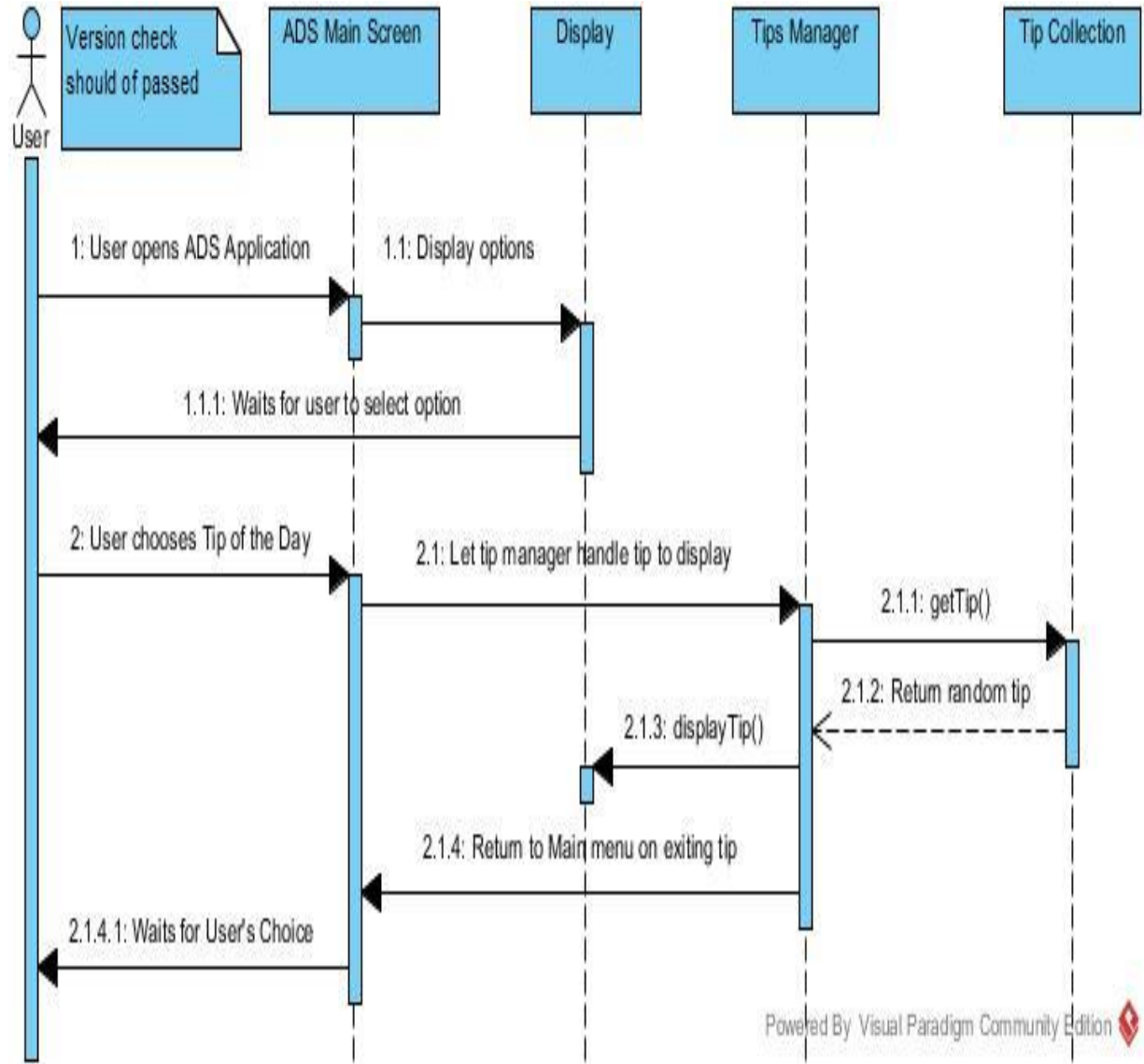
Live Vehicle Information

This lists the steps in monitoring the live information from various sensors positioned at different places in the vehicle, which sends the live information on vehicles performance.



Tip of the Day

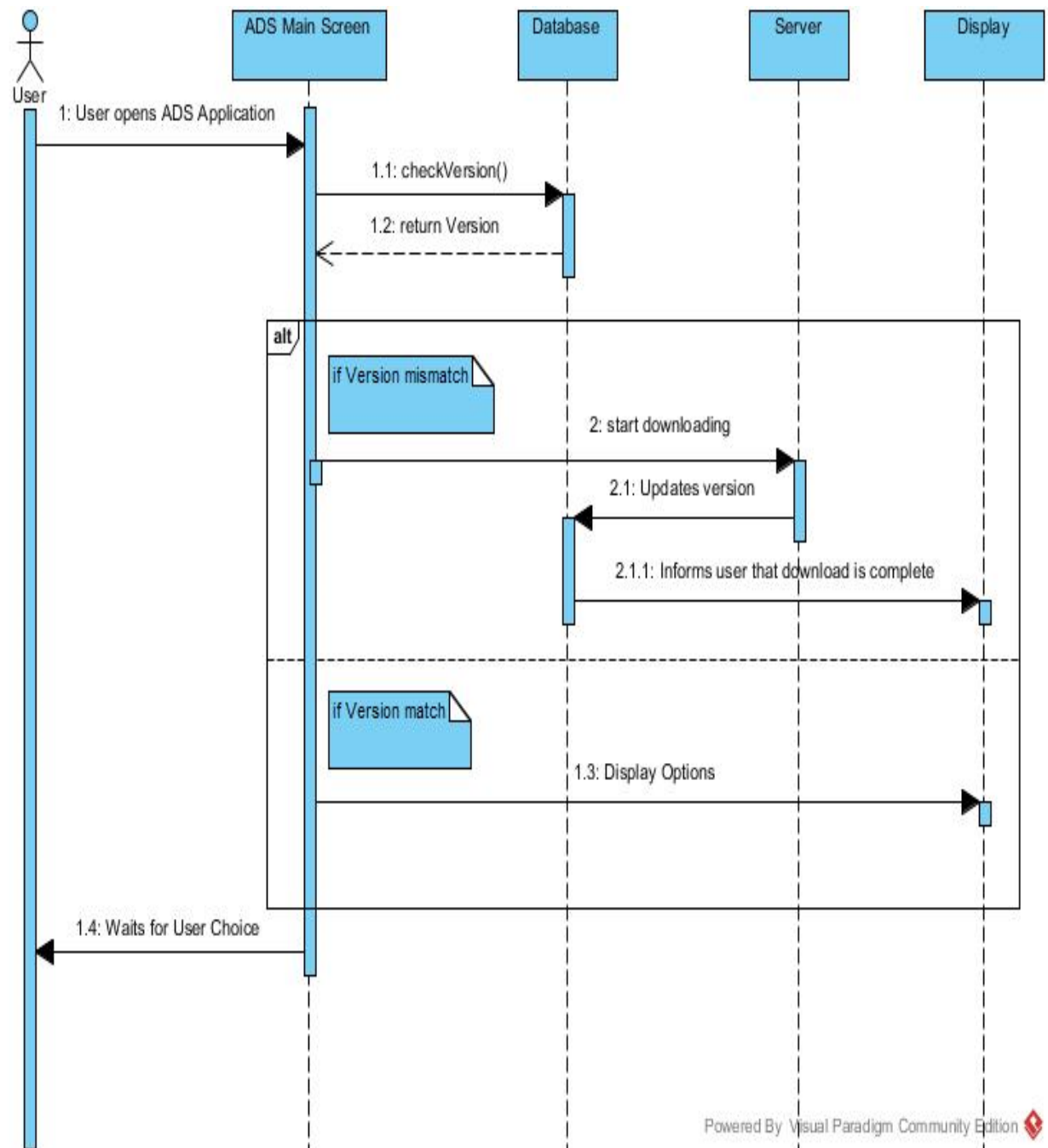
This involves the steps in getting safety tips on a daily basis from the application server where the tip gets updated on a daily basis.



Powered By Visual Paradigm Community Edition

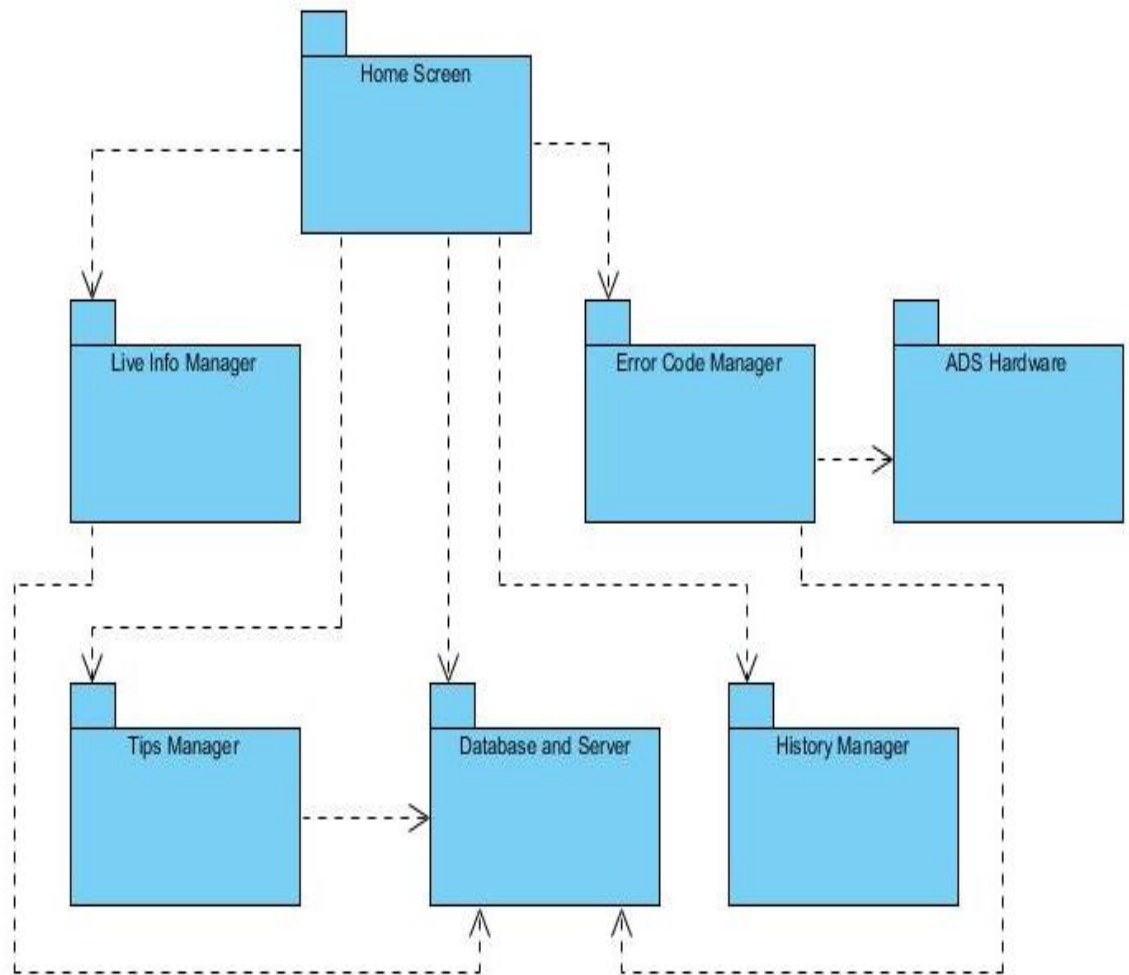
Update Application Database

The application will have to update the error code database as and when a new version of it released from an automobile manufacturer. The diagram below displays the steps for the same.



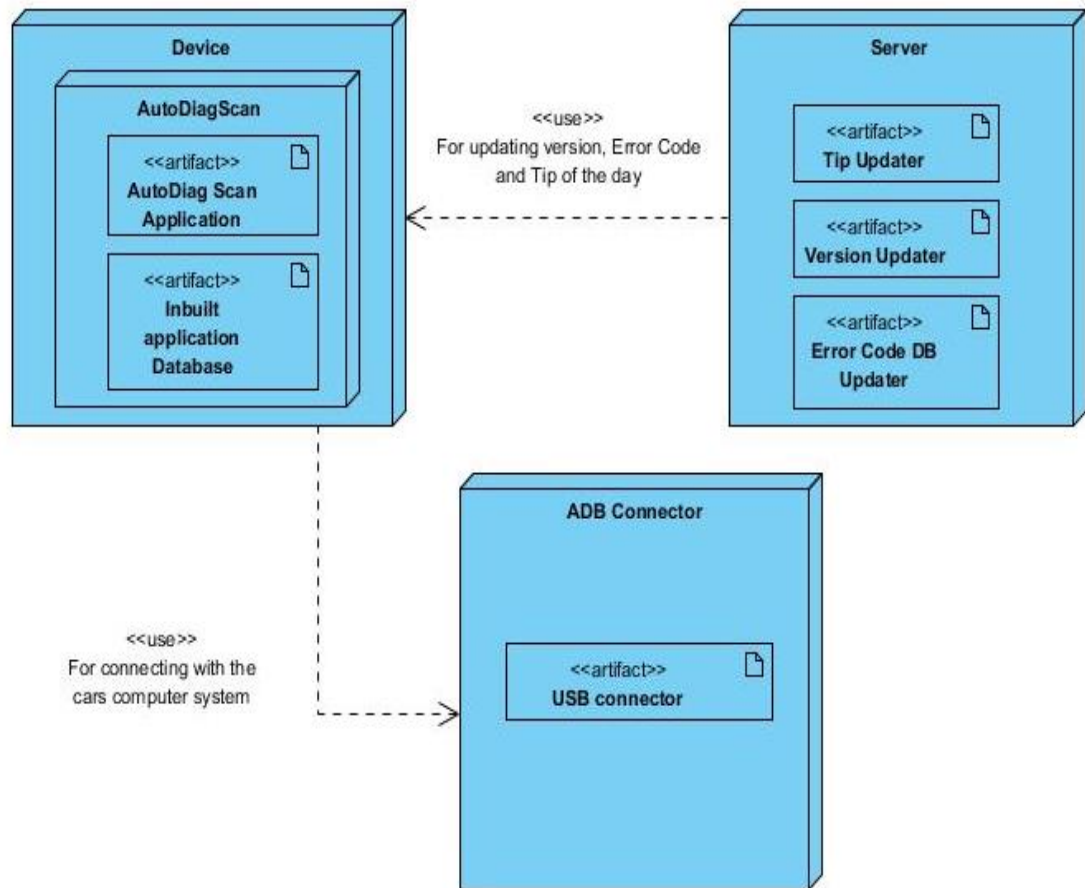
23d Subsystem Decomposition

The below diagram gives an insight into the sub components that together forms the complete application.



23e Hardware / software mapping

The ADS application is installed in the user's mobile phone hosting its own inbuilt database for data storage. Some of the information like daily tips, error code updates are fetched from server.



23f Data Dictionary

This being a mobile application, all the data that the application uses are to be stored in the application itself. The most important tables that the application will be in need of are the Error Codes table, Saved Report table and Daily Tips table. All the tables are stored in the applications inbuilt database for ease of access by the application.

Error Codes Table:

Table Name: ErrorCodes

Fields: ID, Brand, Model, Year, Engine Type, Error_Code, Error_Description, Error_Severity, Fix_Flag, Fix_Instructions, Last_Updated_Date

Saved Report Table:

Table Name: Saved_Report_Data

Fields: ID, Date, Time, Brand, Model, Year, Engine_type, Error_Code, Error_Description, Error_Severity

Daily Tip Table:

Table Name: Daily_Tip

Fields: ID, Date, Tip, Source, Usefulness

23g Persistent Data management

One of the main advantages of the mobile application is portability. This ADS application can be used to troubleshoot the vehicle as and when there is a problem in it irrespective of the place. This cannot be achieved if the application relies on data from a company hosted server as we cannot expect a good internet connection at all places which leads to the use of inbuilt database for the applications working.

There are three main tables that are to be maintained in the application. Starting with the Error Codes table which will hold all the error details based on Make, Manufacturer, Model and Engine type. Followed by a table for saving the user generated report. Finally we will need a table for storing the daily tip that will be extracted from the application server on a daily basis. This is stored in the local database so that it can be showed to the user without having to connect to the internet.

23h Access control and security

The application doesn't store any of the user information which is the reason why it doesn't exercise major security practice. All data that the application stores are non-sensitive data. and since this being a mobile application the owner will be the sole user of the phone and the application.

The basic security practice of the phone will hold good for this application. Since this being non-sensitive data, access to it other than the intended user will not have any major implications.

23i Global software control

Event Driven control flow design is employed in designing of this application. All the buttons in the main screen of the application corresponds to a manager. All the functions that each button corresponds to are delegated to an implementation manager which does the required task.

The best example would be button for scanning the vehicle and getting the error codes from vehicles inbuilt computer system. When the button is clicked the action is delegated to the error code manager which takes over the process from then on and finally displays all the error code retrieved from vehicle.

23j Boundary conditions

The application can be started by clicking on the application's icon from the Application tray. This will launch the applications main screen with various tasks that can be done with the application.

The application can be closed by just pressing the back button or the home button depending on the model of the phone. The application will prompt the user to save the data if the user tries to quit soon after generating an error report. If otherwise the application will close itself without any prompt to the user.

When either the phone or the application crashes during the middle of the error scan, then the system will not store any of the error code information that was extracted until the time of crash. This is done with purpose as there will be a lots a data and if we are to continue scanning from the place where the scan was interrupted then we will end up missing some of the errors. The application is designed in such a way that the scab restarts soon after recovering from a crash.

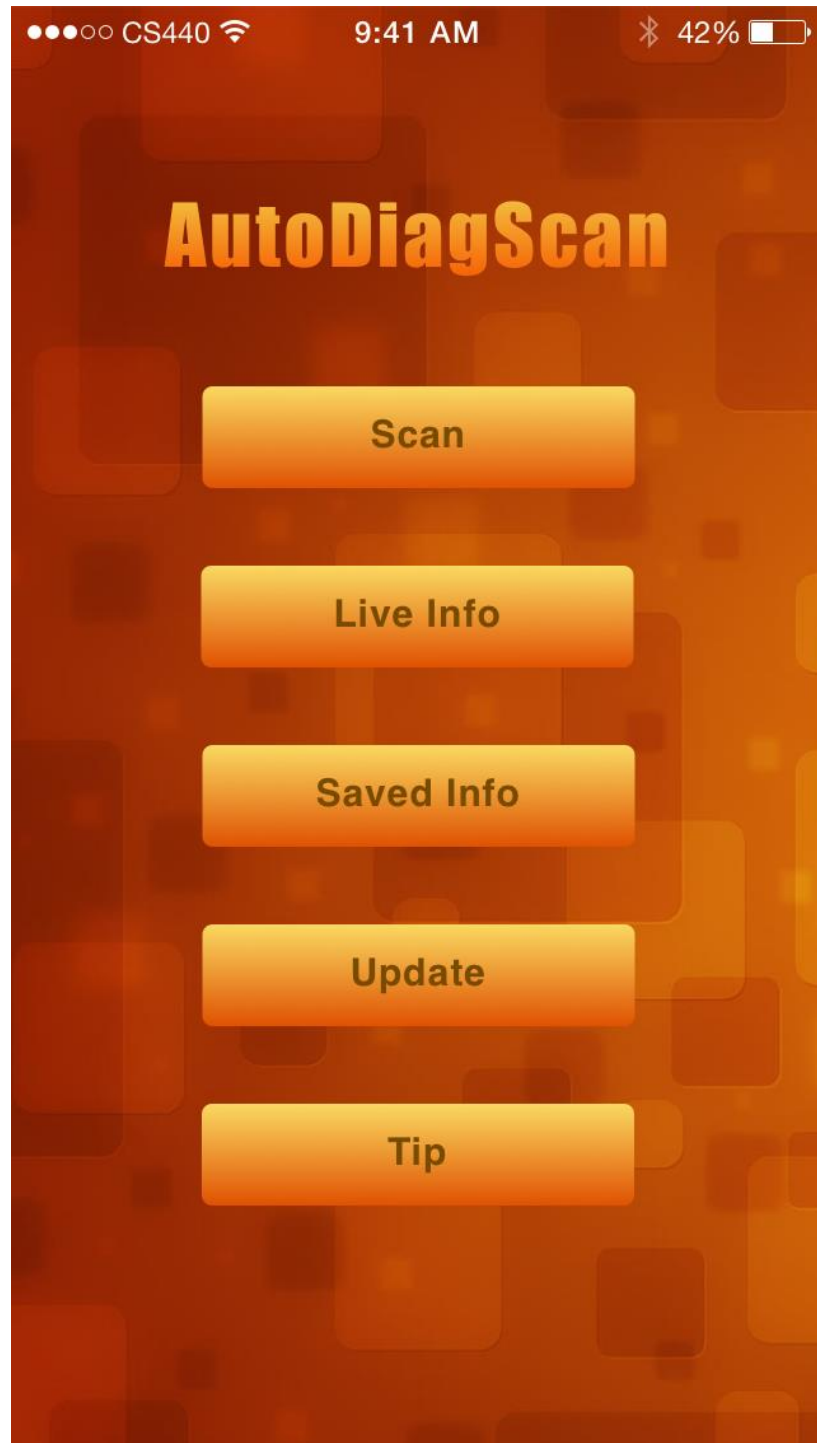
24 Subsystem services

- *Home Screen:* This screen is the Main Menu screen. It shall contain the Scan, Live Info, Saved Info, Update, and Tip options.
- *Error Code Manager:* All of the error codes that are read from the vehicle shall be managed by the Error Code Manager.
- *History Manager:* History Manager shall log all of the information of the scans performed by ADS. The user shall be able to see a list of saved information of all of the cars that has been scanned by ADS. When a car is chosen, the user shall see a list of saved scan results of this car.
- *Lived Info Manager:* The user shall be able to see the live information that ADS reads from the car.
- *Tips Manager:* All of the daily tips shall be stored in the database of the app. The user shall be able to see one daily tip per day.
- *Database and Server:* The database and server shall store all of the error codes up-to-date, the tips, and information of the cars. The app shall be able to retrieve error codes, tips, and car information from its database. When there is a new version of the error code database available, the app shall force update the error code database.

25 User Interface

Below is the design of the user interface of the application on iOS.

- Main Menu



- Scan Input Screen

●●●○○ CS440

9:41 AM

42%

☰

Scan

Brand

Model

Year

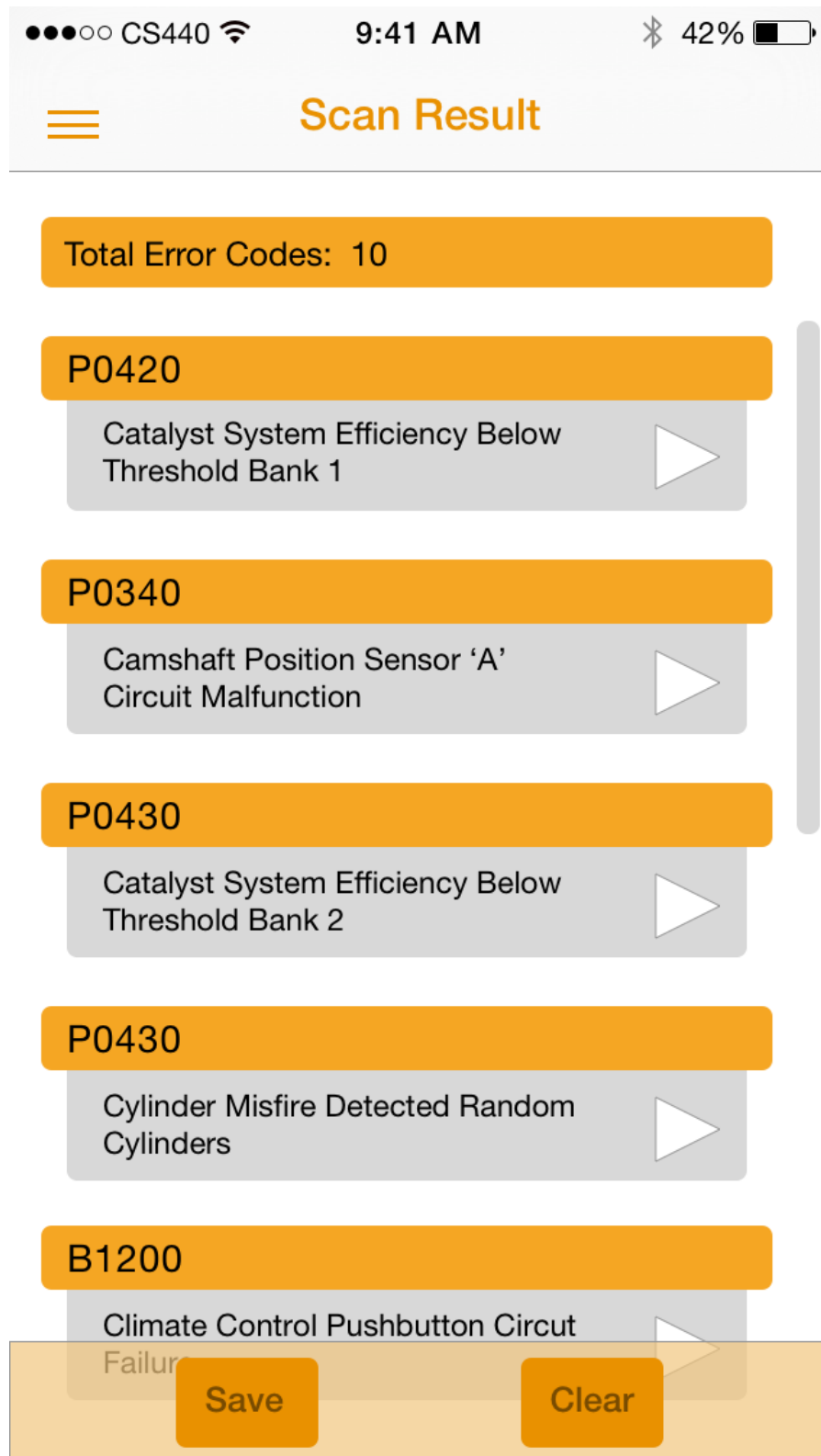
Engine Type

Cancel

Ok

The app asks for information of the vehicle before it starts to scan for error codes.

- Scan Result Screen




This is a list of error codes ADS read from the built-in computer of the vehicle.

- Code Detail

●●●○○ CS440

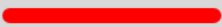
9:41 AM

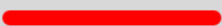
42%

 **P0011 Detail**

P0011

**Intake Valve Timing Control
Performance Bank 1**

Repair Importance Level: 3 

Repair Difficulty Level: 3 

1=Easy 2=Moderate 3=Hard

Recommend: Expert repair
recommended. Problem may require
special tools and experienced technician
to diagnose and repair

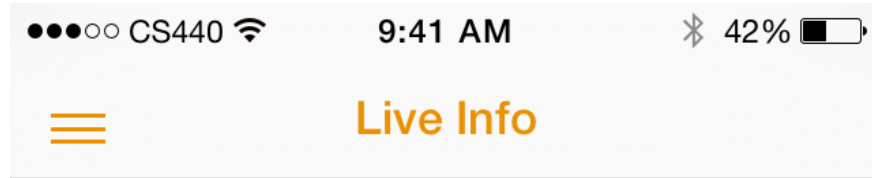
Possible causes:

- Faulty Intake Valve Timing Control Solenoid Valve
- Intake Valve Timing Control Solenoid Valve harness is open or shorted
- Intake Valve Timing Control Solenoid Valve circuit poor electrical connection
- Faulty Camshaft Position Sensor
- Camshaft Position Sensor harness is open or shorted
- Camshaft Position Sensor circuit poor electrical connection
- Faulty Crankshaft Position Sensor (POS)

Find Closest Shop

The image above is the details of an error code that requires a mechanic to fix. It displays the code's information and difficulty level.

- Live Info Screen

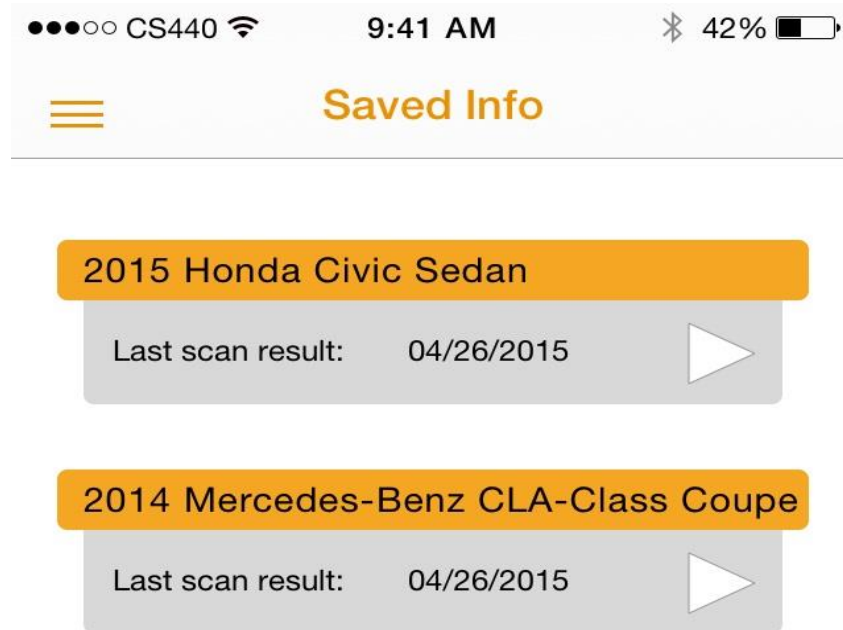


Engine Datastream Recording

ENGINE SPEED	0739 RPM
COOLANT TEMP	70°C
INTAKE TEMP	060°C
VEHICLE SPEED	0KM/H
THROTTLE	001°
BATTERY VOLT	11.4V
OXYGEN	0513MV
IDLE MOTOR	45STEP

ADS reads the information from the car and display all of the information in the live info screen.

- Saved Info Screen



The image above shows a list of saved info in the app. The list contains a list of the vehicles that has been scanned by ADS.

26 Object Design

26a Object Design trade-offs

- Performance:

The application shall focus on performance. It shall not take it more than 1 minute to read the error codes from the car's built-in computer, retrieve all of the information from the database, and display the results. As a result, error codes information shall be cached to boost up the performance of the application.

- Adaptability without loss of Functionality:

Periodic updates and bug fixes shall not affect existing functionalities of the application. The existing app shall contain user data and saved scan results. The updates shall not affect the user data.

- Functionality without loss of Usability:

The app focuses on functionality, yet maintains the ease of use for its users. The app's interface shall be simple and easy to use, and at the same time, it shall provide a rich set of functionalities that are useful for the users.

- Expandability:

More functionalities shall be added through major updates. The design of the app should provide this type of growths and make it easy for improvements and extensions.

26b Interface Documentation guidelines

The following general guidelines should be followed to maintain a uniform and clean interface:

- Package names shall be all in lower case.
- Classes and Interface should be noun phrases, with the first character of each word capitalized. (Eg: HistoryManager, Car, etc.)
- Methods should be verb phrases, with the first character of each word except the first word capitalized. (Eg: addHistory(), getHistoryList(), etc.)
- Constant should be in all capital letters.

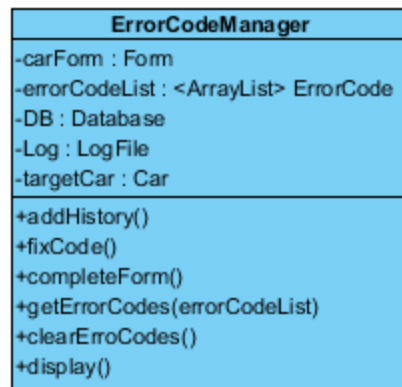
26c Packages

The application system is divided into 5 packages:

- The **hardware** package contains classes that manage hardware related operations and hardware connections.
- The **historymanager** package contains HistoryManager, HistoryEntry, and LogFile classes.
- The **errorcode** package contains ErrorCodeManager and ErrorCode classes. It
- The **storage** package contains Database and Server classes.
- The **app** package contains all remaining classes.

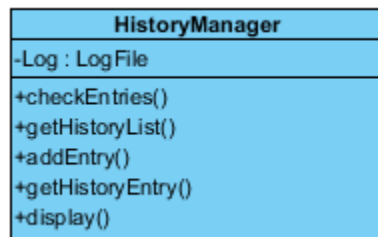
26d Class Interfaces

- **ErrorCodeManager**



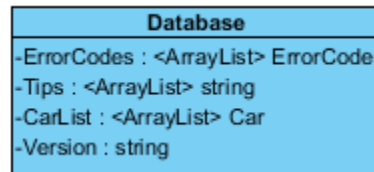
ErrorCodeManager is a part of the errorcode package. This class is responsible for managing the error codes.

- **HistoryManager**



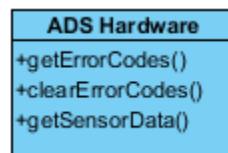
HistoryManager is a part of the historymanager package. It is responsible for managing all the logs the application.

- **Database**



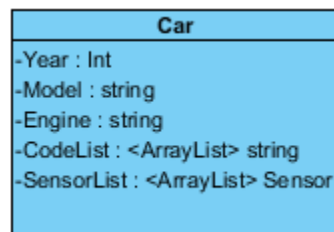
Database is a class in the storage package. Database class is responsible for managing all of data of the application including error codes, tips, and cars.

- **ADSHardware**



ADSHardware is a part of the hardware package. This class manages everything related to hardware. It is responsible for getting data from the car's sensors.

- **Car**



Car class is in the app package. This class contains information of the car that uses ADS application to scan for error codes.

IV Test Plans

27 Features to be tested / not to be tested

- The AutoDiagScan application and the ODB connector shall be thoroughly tested to ensure that it operates at a high standard. The following is a list of general features that will be fully tested before the final release,
 - Application Menu
 - Trouble code generator
 - Live Status Monitoring
 - Database and System Update

- Trouble shooting issues
- Saving the generated error report
- Retrieving saved report
- Tip of the day feature
- Error checking for different models and manufactures from a single device
- The ODB connector is tested for proper communication with the application
- Exit
- More detailed test plans can be found in the Test Cases section
- Features not to be tested are those that reside outside of our control, namely to do with the failure/interrupt caused by the vehicle's computer in which the application is interacting with.

28 Pass / Fail Criteria

- After executing a test, the decision is defined according to the following rules. The test result is set to "PASS" when all steps of the test case are executed and no errors were found. The test result is set to "FAIL" when some of the test steps did not match the requirement or caused error while executing the test case. Test Results that partially satisfy the expected result shall be considered as "FAIL".

29 Approach

- The most efficient means of testing shall be applied throughout the development of this application and for designing the hardware connector.
- Both white box and black box testing will be applied for this application.
- Unit testing is to be performed for every module that's being developed. The unit testing should be completed before integrating the module with the bulk of the product.
- The code must be reviewed before pushing it for further testing.
- Integration testing must be done soon after Unit Testing a module. If the testing is successful the code base should be base lined before testing the next module.
- The hardware connector must be tested across all the industrial standards.
- Design Prototype of the application for every module must be tested for Usability with a sample set of users from and outside the company.

30 Suspension and Resumption

- The testing of the system should stop when:
- At any point during the testing phase an error has occurred. Since certain tests rely on other modules, those modules will have to also be tested before a suspension occurs. The reason is to find out which modules were the cause of the test failing. A log of potential problem modules will be handed to the Development team.

- The testing of the system should resume when:
 - The Development team resolves and fix these issues and give the okay to resume tests. Afterwards the Testing team will restart the tests from the beginning. This process will be repeated until all tests pass and all display results are correct.

31 Testing Materials (hardware / software requirements)

Hardware:

1. Car / Car's computer unit: ADS Hardware is to be plugged into this socket. This computer unit is the major source of data used within the ADS application. It is of top priority to have a computer unit to test if our software application is running correctly.
2. OBD Scanner: A working device that fits the standard when comes to finding out error codes. This item will be used in comparison tests between the scanners results against our application results.
3. OBD Y-splitter Cable: This item is a cable splitter which allows two OBD devices to be connected into one socket. The Y cable will aid in comparison tests between the conventional scanner and our unit. The "Live Data" functionally will use this cable to test for correctness. Both the OBD Scanner and ADS will run live scan simultaneously during the test.
4. ADS Hardware: The product which will be used in conjunction with the ADS application. This hardware is the middle man for receiving data from a car's computer. All test that require information from or to the car's computer will use the hardware.

Hardware + Software:

Smartphone: The mobile application is to work on multiple platforms (Android / iOS). This means that a smartphone for each platform is needed during the tests. Earlier versions of the phone's OS might be required as well to ensure that older versions not be left out. The smartphone should also be equipped with GPS to test the troubleshooting functionality.

Software:

Google Maps: This application is available to both Android and iOS. The application will be used when testing troubleshooting functionality with error codes that are considered "major" issues.

Documentation:

1. Testing status reports
2. Phase completion reports
3. Final report (confirmation that all tests have been passed)

32 Test Cases

Number	TC-01
Module	Port Connection
Title	Test that a connection to the car's computer has been established
Functional Requirement	The application must recognize a connection to the car's computer before a scan can be started.
Test Steps	<ol style="list-style-type: none"> 1. Start the application with the port plugged into the car. 2. Attempt to start a scan. A message box indicating there no connection has been established should be displayed. 3. Attach the Adapter into the ODB port of the car. An icon indicating there is a connection is displayed in the corner. 4. Press the Scan Button. A scan should start promptly after the button is pressed. 5. Disconnect the wire in the middle of a scan. Indicator icon should disappear. An error box explaining that the scan has been interrupted due to a connection issue should be displayed. 6. Reconnect the wire. Indicator Icon should appear again. A message box with a "Start Scan" button informs the user that the scan must start over again should be displayed. 7. Start Scan button is pressed and scan is started. 8. Similar connection tests are ran with "Live Data".
Expected Results	The system should be able to tell if there is a connection established at any point when data transfer from the car to the application is needed.

Number	TC-02
Module	Trouble Code Generator
Title	Test if the application lists the correct set of error codes from the vehicles inbuilt computer system

Functional Requirement	The application must compile a list of error codes along with the corresponding error messages after completing the scan.
Test Steps	<ol style="list-style-type: none"> 1. When the vehicle flashes the Check Engine Light, connect the phone to the vehicle with the help of the ODB2 connector and launch the application. 2. On the Main Menu, application will list the following options “Scan”, “Live Info”, “Saved Info”, “Update” and “Tip”. 3. Select the “Scan” option to launch an error code check. 4. The system will prompt for Model, Model Year, Make and Engine Type. 5. Provide the prompted information according to the vehicle and press on the “Ok” button to start the scan 6. The application will list all the error codes that were read from the vehicle’s inbuilt computer system. The error messages corresponding to the error code are also listed with it.
Expected Results	The error code list and the error message corresponding to each of it must exactly match with the error code list generated by a conventional ODB2 scanner.

Number	TC-03
Module	Troubleshooting (minor)
Title	Test if the “Fix” button opens Browser with relevant information upon access
Functional Requirement	The application must show instructions for minor issues that can be fixed by the owner through a Browser.
Test Steps	<ol style="list-style-type: none"> 1. Input a special case Error Code which is marked “minor” difficulty. 2. Conduct a scan to get the Error List. 3. Click the “Fix” button for the special case error. 4. Ensure that a Browser pops up when the button is pressed. 5. Ensure the Error List is shown again when returning from the Browser. 6. Repeat test for correctness on all cases marked “minor”.

Expected Results	The User gets troubleshooting information via Browser for a “minor” error when the “Fix” button is pressed. The User returns to the Error List upon exiting from the Browser.

Number	TC-04
Module	Troubleshooting (major)
Title	Test if the “Fix” button opens Google Maps with relevant information upon access
Functional Requirement	The application must show the closest Auto Repair Shops relative to user’s GPS location.
Test Steps	<ol style="list-style-type: none"> 1. Input a special case Error Code which is marked “major” difficulty. 2. Conduct a scan to get the Error List. 3. Click the “Fix” button for the special case error. 4. Ensure that Google Maps launches when the button is pressed. 5. Ensure the Error List is shown again when returning from Google Maps. 6. Test correctness of information relayed to Google Maps by pressing the “Fix” button on-site of a few Auto Repair Shops. 7. Test that all “major” flagged errors open Google Maps.
Expected Results	The user gets a list of closest Auto Repair Shops relative to devices GPS location via Google Maps when the “Fix” button is pressed for a major error. The User returns to the Error List upon exiting from Google Maps.

Number	TC-05
Module	Save Error Report
Title	Test that Error Reports have been Saved.
Functional Requirement	The application must let the user save the scanned error report. There should not be duplicates with back to back saves.
Test Steps	<ol style="list-style-type: none"> 1. Complete a scan and get to the Error Code list screen. 2. Press the “Save” button twice for the same Error report (one after another). 3. View old reports with the implemented (correctly working) “View History” option. Confirm that there isn’t two of the same error report listed in the history log. 4. Repeat step 1. 5. Press the “Save” button and save a copy of the current Error Report. 6. Fix an issue and use the “Clear” and rescan option. This will cause one of the errors to disappear when another scan is done. 7. Repeat steps 4 and 5. 8. View the old history log and confirm that there are two different entries in the log.
Expected Results	The user is able to save a unique copy of the current Error Report when the “Save” button is pressed. This Error Report can be viewed with the “View History” functionality.

Number	TC-06
Module	Pulling out saved report
Title	Test if the app displays a saved report when it is retrieved

Functional Requirement	The system must display a list of saved reports from which the user can select
Test Steps	<ol style="list-style-type: none"> 1. Launch the AutoDiagScan application. 2. Complete a scan and save the report. 3. Go to the app Main Menu, select “Saved Report” button 4. ADS displays a list of saved reports. 5. Find and select the report previously saved in step 2. 6. See if ADS displays the report with the correct information of the car and error codes detected in step 2.
Expected Results	The app must display the saved report with all of the information including Year, Make, Model, Date and Time, and a list of error codes.

Number	TC-07
Module	Database update
Title	Test if the app does a forced database update if there is a database update available
Functional Requirement	The system must force an update when a new version of error codes are release
Test Steps	<ol style="list-style-type: none"> 1. There is a new database update available 2. Launch the AutoDiagScan application. 3. Ensure that the phone is connected to the Internet 4. See if the app forces a database update 5. See if the app display a success message 6. Check if the time it takes the app to update its database is less than 5 minutes.
Expected Results	The app must forces an update if there is a new database update released by the manufacturer.

Number	TC-08
Module	Tips
Title	Test if the tips are randomized when the user presses Tip of the day button

Functional Requirement	The system must display a general tip every day.
Test Steps	<ol style="list-style-type: none"> 1. Install and Launch the AutoDiagScan application. 2. On the Main Menu, press the Tip button. 3. See if the system displays a randomized tip. 4. Exit the application. 5. Change the date of the phone to the next day. 6. Launch the app again and press the Tip button once more time. 7. See if the app displays a different tip this time.
Expected Results	The app must displays a different tip every day when the user press the “Tip” button.

Number	TC-09
Module	Clear Errors
Title	Test that errors are cleared
Functional Requirement	The application must be able to clear error codes which have been identified by the car’s computer.
Test Steps	<ol style="list-style-type: none"> 1. Complete a scan and get to the Error Code list screen. 2. Press the “Clear” button. 3. Ensure that the lights that were triggered before (Check Engine Light) has been turned off after clearing is done. 4. A display screen should appear indicating a rescan is to be done after the lights are cleared. 5. View old reports with the implemented (correctly working) “View History” option. Confirm that there is a saved/logged report for the previous error report.
Expected Results	The user is able to clear the error codes on the car’s computer that have been fixed. The second scan is to make sure those errors that were not fixed are listed for viewing. An error report is saved in case there is a need to look at past issues.

33 Testing Schedule

Type of Testing	Comments
Static Testing	Duration: 3 days. Begins at the end of each Sprint period. Tests include reviewing, walkthrough, and inspection of code without execution of code.
Unit Testing	Duration: 1 week. Begins 5 weeks before each release
Component Testing	Duration 3 days. Begins after Unit Testing is complete.
Integration Testing	Duration: 1 week. Begins after testable parts are confirmed from component tests. Ideal integration testing method is “Big Bang” This testing period also includes interface testing.
Usability Testing	Duration: 4 days. Begins 2 weeks before each release. A/B tests against working scanners will also be done. Usability testing method will be “Hallway testing” along with “Expert review”.
Regression Testing	Duration: 2 days. Begins after each major code-based change following integration testing.
System Testing	Duration: 3 days. Begins 2 weeks before the final release
Release Acceptance Testing	Duration 5 days. Begins 1 week before the final release

V Project Issues

34 Open Issues

There is no major issue with the application as of now. There may implementation issues when developing the application across different platforms. Programming features are not the same across Android and I Phone which would lead to development problems. The design will have to be tweaked a little when faced with implementation issues especially between Android and I Phone, and the changes should be handled in such a way that the interface is same for the most part and consistent throughout.

Apart from the foresaid development issue we may be faced with integration issues when the development is being developed in modules. This can be handled easily as long as the implementation is done based on the chalked design. There are no major design flaws in the application which would stall the development process as such.

35 Off-The-Shelf Solutions

35a Ready-Made Products

The applications concept is totally new because of which there are no ready more components that can aid in the application development. Lot of program specific components like the SQLite database can be used for storing data in the application. IDE's like Eclipse, Android Studio and Xcode can be used for application development in Android and iPhone.

35b Reusable Components

Reusing the developed code across the application will save us a lot of development time and will let us concentrate on more trivial tasks. The application interaction with the ODB 2 device to read data from vehicles inbuilt computer system is going to remain the same across all the features that the application offers. This piece of code can be reused once after it has been successfully implemented for the very first functionality that is developed for the application.

The application layout and design is common across all the features that application offers. The base layout and design of the application can be used across the application with added little features according to the feature that's being implemented.

35c Products That Can Be Copied

This application is one-of-a-kind without any known players in the market especially in this segment of auto industry. This imparts a major reasonability on this application to set the bench mark for this kind of applications. The very reason rules out the option of getting design ideas from other applications.

36 New Problems

36a Effects on Current Environment

- The application will not cause any harm both to the people using it and the environment in which it being used.
- Protective measures are considered in designing the ODB2 connector which helps the application communicate with the vehicles computer system, given the fact that the vehicles internal temperature is high if the scanning is done soon after the stopping vehicle after a long drive
- The application is designed in such a way that, the user's saved information are neither distributed nor shared with other organizations at any cost.
- Other basic security measures that is vital for an application are also implemented in this application

36b Effects on the Installed Systems

- The application is intended for Android and iPhone running phones and it will not modify or optimize the systems performance for running this application.

- The application is designed to run with the basic system configuration

36c Potential User Problems

- Cars are prone to problems after a long run which is when the application is to be used for finding the problem. Users are advised to wait for a while and let the engine cool down before plugging in the device with the vehicles computer system.
- Care to be taken by the users when fixing minor issues. They are highly advised to contact a mechanic if in case they are completely confident about themselves fixing the vehicle.
- The mobile phone should be handled with care when testing for error in the application as the heat generated from the vehicle may damage the phone when in close contact.

36d Limitations in the Anticipated Implementation Environment that may inhibit the New Product

- The mobile platform is prone to lots of changes in a very short period of time. It gets upgraded almost every year. Given this fact the application can be modified to run without any problems for the newer system platforms but the backward compatibility cannot be provided for all the previous installments expect for a few that was released in the recent past.
- When installed in a phone that doesn't meet the basic hardware requirements, the application will run on a slower pace with frequent interruptions.

36e Follow-Up Problems

- During the live monitoring of the vehicle, data in feed only from a maximum of five sensors can be shown in the mobile owing to the size and performance of the phone.

37 Tasks

37a Project Planning

Since scrum is a well-known iterative and incremental agile software development method or framework, it will be used in the developing the application AutoDiagScan.

It defines a flexible product development strategy where the development team works as a unit to reach common goal. Assuming challenges of the traditional sequential approach to product development, this enable the team to have a close online collaboration of all team member as well as daily face to face communication among all team members and also, disciplines in the project. This framework help's to complete deliverable product in a timely manner and testing phase is used to clean out the bugs.

- This is a fast pace, cutting edge development, which is quickly coded. The testing is done right away to fix bugs and mistakes.

- Using scrum, specific features can be figured out and can be assigned to specific sprints and then release plans.
- Scrum is highly controlled, so it entails frequent updates of progress in the work through regular meetings. Everyone is on board with all the progress made and going forward.
- The developer in the team branch out and work on the code separately there creating separate instance for the features of the product. Once the scrum master approves code, it is the merge into main branch.
- With daily meeting on schedule, productivity of individual developer are been measured and this lead s to improvement of the team in general.
- Changes and issues are identified in advances due to meetings of the teams and there are resolved.
- The overhead cost is minimal, so this will to a quicker and cheaper cost of production.

In Scrum implementation of the application AutoDiagScan, various features have been separated and are being forwarded from the product backlog to their respective sprint.

Once the Team commit to the work, the project or product owner cannot add more work, alter course mid-sprint or micromanage. This spring should be about two weeks long.

Due to short sprints and constant feedback, it becomes easier to cope with the changes. Also, constant small sprints are being in each development phase.

Just as every sprint begins with the sprint-planning meeting, the sprint concludes with the sprint review meeting, in which the team presents its work to the Product Owner. During this meeting, the Product Owner determines if the team's work has met its acceptance criteria.

After completion of all sprints a working product would have be created on which various modes of testing helps in removal of bugs and improving performance.

37b Planning of the Development Phase

The development of the development of the application AutoDiagScan is planned in the following stages:

- First of all, the requirement gathering stage will start. This will include gathering functional and non-functional requirement. Like requirement for functionality, usability, reliability, performance and supportability.

- This will then follow by the analysis phases and the design phase. Which design pattern to use and so on.
- So implementing the coding part, all functionality must be taken into consideration, so that the application will have them on deploy or when ready for deployment.
- Specific features can be figured out since scrum is the implementing model. These features can be assigned to specific sprints and release plans.
- A sprint is two weeks or fourteen days long.
- Sprint 1: Trouble Code generator, live status monitoring and troubleshooting issues. After this sprint, a review meeting should be conducted in which the team will present the job to the Project or product owner. At this time the owner will decide if this meets the acceptance criteria's.
- Sprint 2: User interface and mobile development. Again, after this sprint, a meeting should be schedule with project owner for acceptance criteria's.
- Sprint 3: Database and system updates. After sprint 3, a testing sprint should be planned and review meeting also. Test the application using techniques like black box and white-box testing.

38 Migration to the New Product

38a Requirements for Migration to the New Product

There is no currently existing product. Therefore, we will not be migrating from an existing product to a new product. ADS application is a new product that will be built from scratch.

38b Data That Has to Be Modified or Translated for the New System

ADS application is a new product, hence there will be no data to be modified or translated to the new system.

39 Risks

The following risks are the most relevant to our projects:

- Excessive schedule pressure
- Management malpractice
- Inaccurate cost estimating

40 Costs

The cost estimate for ADS application will be little higher when compared to other normal applications as it includes parallel development in both Android and iPhone. It also includes the cost of design and development of ODB2 connector. The two major costs are listed below,

- Monetary Costs
 - Project setup cost
 - Copyright charges
 - Payroll expenses
 - Testing costs
 - Advertisement costs
 - Travel and Requirement gathering expenses
 - Material and research cost for the hardware designing
- Non-Monetary Costs
 - Design issues during the development phase
 - Target platform upgrade
 - Rise in man power costs

41 Waiting Room

Other requirements that we would like to develop in the future releases of the application include:

- Make the application work not only on mobile OS, but also work on Windows and Mac OS.
- Develop in-app support, the user can chat with a professional about their vehicle's issue or car related problems.

42 Ideas for Solution

There were lots of requirements that came up for this application which we had to let go because of implementation and time constraints. Some of the requirements that were not considered as part of the final solution are,

- Integrating the application the vehicles manufacturer to keep track of the vehicles condition and notify the users as and when the vehicle has to be brought to the manufacturer's facility for servicing.
- For major issues, the application will inform the nearest available mechanic and have them come attend to the vehicles problem.
- Saving the report information in cloud.
- Advertisements in the application

43 Project Retrospective

The ADS application will prove worthy of the effort that was spent in designing and developing the application. Below are the notable aspects of the application,

- The application will come in handy for solving many of the basic problems for which we have been visiting the mechanic all this while.
- Helps the users in maintaining their vehicle in a good condition with frequent scans and fixes.
- Will assist the user in locating auto workshops that specializes in fixing the current problem of the vehicle, especially during emergency situations.
- The applications versatile design is the backbone of its support in different mobile platforms which has to be used in implementing the application in computers platforms.

VI Glossary

ADS – Auto Diag Scan is abbreviated as ADS throughout the application

ODB2 – This is a special form of connector that helps in connecting the vehicle's computer system with the mobile application

Android – This is one of the most prevalent mobile platform developed by Google

SQLite – This is the inbuilt database that comes with the mobile phones

VII References/Bibliography

- Bernd Bruegge and Allen H. Dutoit, "Object-Oriented Software Engineering - Using UML, Patterns, and Java", third edition

- Design Patterns Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.
- UML Distilled by Martin Fowler, third edition.
- Visual Paradigm 12.0 for creating Class diagrams, sequence diagrams and package diagrams.

VIII Index

Class Diagrams,	40	Scope,	10
Class Interfaces,	61	SkyGraph,	23, 24, 26
Dynamic Model,	42	Test Cases,	65
Index,	7	User,	22, 23, 24, 25, 26
Object Design,	60	User Interface,	53
Requirements,	21, 67		