

# Car Systems

---

## Requirement Specifications

### Group LM4

Kapil Haresh Vigneswaren  
Ng Shien Wee  
Yeoh Hui Jia  
Dyalan Shanmugarajah

## Table of Contents

### Introduction

Purpose.....3

Scope.....3

System Overview.....4

System Structure.....6

Users.....7

Constraints.....7

Requirements Summary.....8

Functional Requirements.....10

Non Functional Requirements.....18

# Introduction

## Purpose

This document is designed to describe the Car Systems project, to be branded as Drvr, and its requirements, to the members of the project and the relevant stakeholders

## Scope

This document will be providing a general overview of the Car Systems project, addressed as Drvr from this point on. As this is only an overview, we won't be going into the specific details (like implementation methods) of the project.

In addition, this document will list down all the necessary requirements with its priorities needed for this project, like the features and qualities it should have. However this requirements list is not exhaustive and is subject to changes should it be deemed fit.

## System Overview

Drvr, stylized from the word, Driver, has one goal in mind, and that is to teach the user about the various systems available in a car, so that a user has a better understanding on how a car works.

The system has three large components that make up the learning process. These components, while independent, help the user to achieve the same goal, that is to learn more about their cars. The features are as below:

- Infographic section

This is the first feature of the system, where the user is expected to begin their learning experience. As the system is designed for beginners, we expect that the user has no prior knowledge on cars. Due to this, the infographic section will display an image of a car, which allows users to change the view and select on a certain part of the car to know more about it. For example, if the user would like to know what is a sunroof, by clicking on the sunroof in the image of the car, the description of the sunroof would then appear on the side, telling the user how the sunroof of a car works and what is its purpose. The main goal for this feature would be for the user to familiarize themselves with the various sections of the car before they move on to the next stage.

- Simulation

This feature is expected to hold the bulk of the appeal of this product. One of the unique aspects of this section is that it allows for the user to “control” the parts of a car, like switching on the headlights, or activating the windshield wipers. Some form of conformation would accompany this action on the virtual instrument cluster to indicate the action completed successfully.

In addition to that, the user would be able to learn more via the simulation of how information or substances move about in the car. For example, a fuel system could be simulated to show how does the fuel move from the tank to the engine. The simulation also allows the user to see the various differences between different implementations of a car, like the electric propulsion system in an electric vehicle against a fuel-powered engine in a regular car. This is expected to teach users with a bit more detail on how the car works. It is expected that the simulations in this feature would be multithreaded.

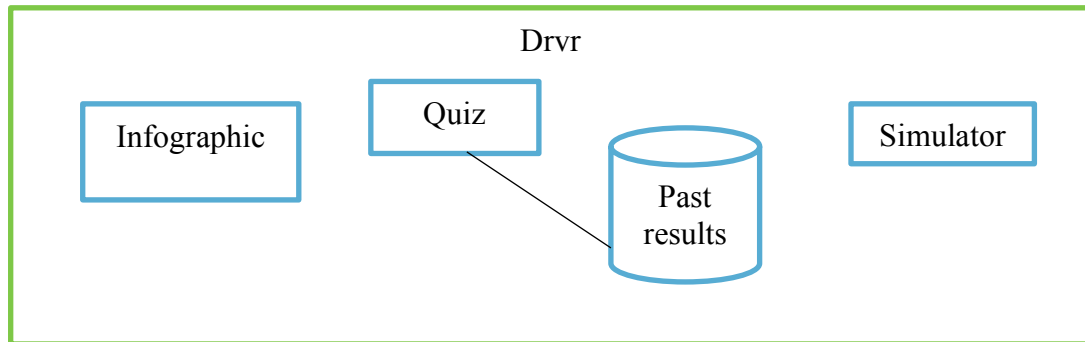
- Quiz

This feature marks the culmination of the learning process for the user. As a timed exercise, the user is presented questions related to the various parts of the car or in relation to the simulations of the systems in the car. It would be a selection of multiple-choice questions, and the user should be able to track their scores of their previous attempts. The expectation is that a user should have spent sufficient amount of time with the infographics and simulation to be able to correctly answer the quiz. Every time the quiz is started, the questions should be chosen from a bank of questions at random.

The system is designed to be a standalone application, which would run on a regular Windows desktop/laptop. The user would only need a mouse and keyboard to move around this program.

## System Structure

It would be reasonable to say that the system is made up of three main components working together, to deliver the experience. This breakdown is as per below:



In essence, Drvr is made up of three main components, namely the Infographic component, the quiz component and the simulator component. While there is no dependency between the three components for the system to function (that is the infographic doesn't depend on the simulator for the system to function) the user would depend on all three components to be able to get a good experience with using the system.

We have opted against using a game engine to build this program, as our group consists of students who don't have adequate experience in game engine, hence it would be too long for us to be able to learn up the game engine before we can actually build the system. This would lead to a significantly lower number of features being delivered. However, although it is being built without a game engine, we would not need to build any form of underlying framework before we can start working on the project. At most, we would just need to build the UI first then implement the functions of the application.

The following is a breakdown of the purpose of each component in the system:

- Infographic section

In this section, we will be able to provide the user a visual breakdown of the car. This section will allow the user to be able to find out more about how each part of the car helps to ensure the car is usable. As we are not targeting mechanics who would need highly detailed schematics, this section would be detailed enough without making it too intimidation (or causing an information overload) for the target user.

- Simulation

The user will be able to virtually control the car in this section. In this section, the controllable items like lights, indicators, wipers and climate control system would run in their own threads, to allow them to run independently, while at the same time, being able to send messages to other parts of the car, like turning on the lights would send a message to the instrument cluster thread to light up the headlight on indicator. Besides being able to control the car, the user will be able to view the simulation of how a car works, for example, they would be able to see how an electric drive train works. Again, this demonstration will be multithreaded, such that each component in the drive train (for example) can communicate with the other components in it, like the electric motor can

send a message regarding it's power use to the thread that runs the battery so that the battery level can be updated.

- Quiz

For the quiz component, a randomizer should be used to randomly pick out a specified number of questions from a question bank and presented to the user. The user is expected to have used both the simulation and infographic component to be able to answer the questions correctly. There should be some form of tracking of the user progress. This would depend on access to the past results file.

## Users

For this system, there is only one type of user that is expected to use it, that is a general user. This user would be someone who would like to improve their knowledge on cars, and is expected to use this software to complement what they have learnt in the past about cars.

## Constraints

At this point in time, there is only one constraint for the system:

- **Operating System:** It is expected that the user is running at least Windows 8 and above, on a reasonably powerful machine.

## Requirements Summary

There are two levels of priorities for the requirements:

- High - Requirements necessary to fulfill the core functionality of the system. These must be delivered.
- Medium - Requirements that provide additional functionality for the service. We will try to deliver all of these features.
- Low – Stretch goals, which would be nice to have, but may be unfeasible to build due to the lack of time and resources.

The following is a breakdown of the functions in the system.

### A) Infographic of a car

Function	Priority
User is able to choose info graphics of multiple cutaways of a car	High
User is able to rotate the infographic of a car	High
Obtain more information about the part of a car	High
Link to online video/animations showing the part of the car in action – Medium	Medium

### B) Simulating multiple situations and functions of a car

Function	Priority
Adding components to the car	High
Simulate difference between front, all and rear wheel drive	High
Simulate the fuel system in different situations	High
Simulate cooling system in the car	Medium
Simulate electrical system in the car	High
Simulate steering system	Medium
Simulate braking system	High
Simulate adding attachment to the car	Medium
Simulate alternative fuel powered drivetrains	High
Simulate difference in driving conditions to fuel consumption	High
Allow multiple faces for the instrument cluster	Medium
Instrument cluster must be displayed always	High
Active tracking of car temperature, fuel and speed	High
Environment visualization should be possible during the simulation	Low

### C) Quiz on parts of the car

Function	Priority
Quiz should be timed	High



Questions should be preselected from the question bank	High
Questions should test user understanding on the car	High
Situation based questions should be included	High
User should be able to track their past scores	Medium
There should be at least 50 questions preloaded in the system	High
The system should be able to submit results to an online database and allow the user to see their global score	Low

#### D) Additional features

Function	Priority
The software should have some multiregional support	Medium
There should be a thread monitor for the system	High
Each feature (F1, F2, F3) should run in their own threads	High
Each component in a simulation should run in their own thread	High
There should be inter-thread communication being implemented	High
Threads that are no longer needed should be properly ended	High
The software can be controlled using a USB steering wheel and pedals	Low
The software can support the use of a VR device and Leap Motion	Low
There should be a companion version that will run on tablets	Low

## Functional Requirements

### F1: Infographic of a car

This section will cover the requirements of the first main feature of the application, which is the infographic of the car

Requirement #: F1.1	Requirement Type: Functional	Use Case #: -
<b>Requirement: A user is able to choose infographics of multiple cutaways of a car</b>		
<b>Description:</b> The system should allow the user to be able to switch views of the car, to be able to see various views such as powertrain view, exterior body view and interior view. This allows us to easily view various parts of the car, i.e. the engine cannot be seen unless we are able to remove the exterior shell of the car		
<b>Requirement Level:</b> High		

Requirement #: F1.2	Requirement Type: Functional	Use Case #: -
<b>Requirement: A user is able to rotate the infographic of the car</b>		
<b>Description:</b> A user should be able change the perspective of the car, that is they should be able to view the rear, side and front of a car. This allows for a user to be able to clearly see the various parts of the car		
<b>Requirement Level:</b> High		

Requirement #: F1.3	Requirement Type: Functional	Use Case #: -
<b>Requirement: A user is able to obtain more information of the part of the car</b>		
<b>Description:</b> By clicking on the name of the part that maps to the car in the infographic, a user should be able to get a description of the part		
The description should include the following: <ul style="list-style-type: none"><li>• Part name</li><li>• Purpose of the part</li><li>• Other names of the part (i.e. a trunk is called a boot in different variations of English)</li></ul>		
<b>Requirement Level:</b> High		

Requirement #: F1.4	Requirement Type: Functional	Use Case #: -
<b>Requirement: Link to online video or animations showing the part of the car in action</b>		
<b>Description:</b> On clicking the name of the part of the car and getting the description of the part, the system should try to get a link to the internet to obtain a video of that part in action.		
This video link could be potentially hard coded into the system to ensure the video will always be there, and is consistent with the part of the car		
<b>Requirement Level:</b> Medium		

## F2: Simulating multiple situations and functions of a car

This part of the program is probably the largest part, with the most amount of functionality. It aims to allow the user to get a demonstration on each of the parts of the car, to allow a user to get a feel on how each part works.

Requirement #: F2.1	Requirement Type: Functional	Use Case #: -
<b>Requirement: Adding components to the car</b>		
<b>Description:</b> A user should be allowed to add some selected components to the car, easily. Components that can be added may include things like fog lights, different rims, etc. There should be the limitation on the type of changes that can be done, like a user cannot add 2 sets of fog lights for example, or add more than 4 wheels.		
<b>Requirement Level:</b> High		

Requirement #: F2.2	Requirement Type: Functional	Use Case #: -
<b>Requirement: Simulate the difference between front wheel drive, rear wheel drive and all wheel drive</b>		
<b>Description:</b> It should be possible to show the difference between all wheel drive, front wheel drive and rear wheel drive. The user should be able to see the difference in the way power is being sent from the engine to the driven wheels, and switch between the 3 modes.		
<b>Requirement Level:</b> High		

Requirement #: F2.3	Requirement Type: Functional	Use Case #: -
<b>Requirement: Simulate the fuel system in a car in different situations</b>		
<b>Description:</b> The user should be able to see how fuel flows to a car in various situation, and see “live” how fuel from the fuel tank is transferred to the engine under the following situations: <ul style="list-style-type: none"><li>• Engine start-up</li><li>• Acceleration</li><li>• Cruising/Idling</li></ul> The actions of acceleration and cruising would lead to fuel being used; hence the amount of fuel in the fuel tank should be actively updated in this mode.		
<b>Requirement Level:</b> High		

Requirement #: F2.4	Requirement Type: Functional	Use Case #: -
<b>Requirement: Simulate the cooling system in the car</b>		
<b>Description:</b> The user should be able to see how the engine and drivetrain’s temperature is managed. Flow of coolant fluid should be shown. The user would be able to see how the cooling system works under the following conditions: <ul style="list-style-type: none"><li>• High temperature</li><li>• Normal/Low temperature</li></ul>		
<b>Requirement Level:</b> Medium		

Requirement #: F2.5	Requirement Type: Functional	Use Case #: -
---------------------	---------------------------------	---------------

<b>Requirement: Simulate the electrical system in the car</b>		
<b>Description:</b> The user should be able to experience how the electrical systems work in the car. Examples of electrical systems in the car that should be simulated include: <ul style="list-style-type: none"> <li>• Lights</li> <li>• Windscreen wipers</li> <li>• Audio system</li> <li>• Cruise control</li> <li>• Rear defrost</li> </ul> <p>As the electrical features in the car are “turned on”, their status should be updated in the control panel of the car, for example, if lights are turned on, the lights on indicator light should be lighted in the control panel of the car in the program.</p>		
<b>Requirement Level:</b> High		

<b>Requirement #:</b> F2.6	<b>Requirement Type:</b> Functional	<b>Use Case #:</b> -
<b>Requirement: Simulate steering system in the car</b>		
<b>Description:</b> The system should be capable on showing how the steering system in the car works, allowing the user to see how the wheels get moved when the steering is turned from left to right (like the movement in the rack and pinion steering system) <p>It should mimic a real steering system, so it should have a point in which it cannot turn anymore (reached the lock of the steering system)</p>		
<b>Requirement Level:</b> Medium		

<b>Requirement #:</b> F2.7	<b>Requirement Type:</b> Functional	<b>Use Case #:</b> -
<b>Requirement: Simulate braking system in the car</b>		
<b>Description:</b> The system should be capable of showing how the brakes work in a car. It should be possible to show the user both how the regular pedal operated brakes work and how does the hand brake work.		
<b>Requirement Level:</b> High		

<b>Requirement #:</b> F2.8	<b>Requirement Type:</b> Functional	<b>Use Case #:</b> -
<b>Requirement: Simulate adding attachments to a car</b>		
<b>Description:</b> The user should be allowed to add certain attachments to a car, like a trailer or a motor home. It should be possible for the user to see how the attachment hooks on to the trailer hitch at the rear of the car		
<b>Requirement Level:</b> Medium		

<b>Requirement #:</b> F2.9	<b>Requirement Type:</b> Functional	<b>Use Case #:</b> -
<b>Requirement: Simulate alternative fuel powered drivetrain systems</b>		
<b>Description:</b> In addition to a regular internal combustion drivetrain systems, the system should be able to also demonstrate how alternative fuel powered drivetrains or hybrid drivetrains work <p>Example of drivetrains that should be demonstrated include:</p> <ul style="list-style-type: none"> <li>• Petrol-electric hybrid drivetrains</li> <li>• Hydrogen drivetrains</li> <li>• Electric drivetrains</li> </ul>		

<b>Requirement Level: High</b>		
<b>Requirement #: F2.10</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: Simulate the difference in driving conditions on fuel consumption</b>		
<b>Description:</b> The user should be able to see how various driving conditions affect fuel consumption. During this process, an instantaneous fuel consumption and fuel level should be actively updated and displayed to the user. Among the driving conditions that should be demonstrated are: <ul style="list-style-type: none"> <li>• Uphill driving</li> <li>• Downhill driving</li> <li>• Flatland driving</li> <li>• Driving at high altitude</li> </ul>		
<b>Requirement Level: High</b>		
<b>Requirement #: F2.11</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: Allow multiple faces for the instrument cluster</b>		
<b>Description:</b> The system should be able to allow the user to change the layout of the instrument cluster and speedometer to various predefined layouts. This may include different coloured clusters, or a completely different layout altogether		
<b>Requirement Level: Medium</b>		
<b>Requirement #: F2.12</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: Instrument cluster must always be displayed</b>		
<b>Description:</b> There should be some form of an instrument cluster and speedometer, with a message center that is persistently displayed in this mode of the software. The cluster should use standardized symbols set out by automotive manufacturers, and this cluster should be running on a thread of its own.		
<b>Requirement Level: High</b>		
<b>Requirement #: F2.13</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: Active tracking of car's temperature, fuel and speed</b>		
<b>Description:</b> As long the user is in this mode of the system, as part of the instrument cluster and message box, there should be a speedometer, a reading of the car's temperature and fuel level. This may be displayed in an integer value or a digital bar (for the fuel level and temperature). This active tracking should be running in a thread of its own as long the user is in this mode.		
<b>Requirement Level: High</b>		
<b>Requirement #: F2.14</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: Environment visualization should be possible in the simulation</b>		
<b>Description:</b> While the user is in the simulation mode, they should be allowed to switch to a view, which renders in real time a visualization of the environment. This would allow the user to feel as though they are "driving" the car in real life. For example, they should be able to see how the windscreen wipers are actively wiping the windscreen in front of them when it is turned on.		
<b>Requirement Level: Low</b>		

### F3: Quiz on parts of the car

The third part of the system is designed to be able to allow the user to test their understanding about cars, so that they would be able to get an idea on the parts of a car that they are not too sure of. It should be in the form of a multiple-choice selection, so the user chooses the correct option to gain a mark.

Requirement #: F3.1	Requirement Type: Functional	Use Case #: -
<b>Requirement: Quiz should be timed</b>		
<b>Description:</b> The quiz must be held with a time constraint, as this would allow the user to learn to make a quick decision, as on the roads, drivers have to be able to react quickly and make quick decisions. It would be expected that the quiz should allow the user to answer questions at a rate of 1 question per minute.		
<b>Requirement Level:</b> High		

Requirement #: F3.2	Requirement Type: Functional	Use Case #: -
<b>Requirement: The quiz should preselect the questions that will be asked randomly from a question bank</b>		
<b>Description:</b> The system will choose a random set of questions, with varying difficulty, every time the quiz mode is started. The random function should run on a thread of it's own, as this would allow for some form of processing/loading animation to be displayed while waiting to show the system is still responsive. The random function should choose exactly 10 questions to ask the user.		
<b>Requirement Level:</b> High		

Requirement #: F3.3	Requirement Type: Functional	Use Case #: -
<b>Requirement: Questions should include testing the user's understanding on the parts of the car</b>		
<b>Description:</b> Since the system mainly demonstrates the various parts of a car, the user should be		
<b>Requirement Level:</b> High		

Requirement #: F3.4	Requirement Type: Functional	Use Case #: -
<b>Requirement: The quiz should have situation based questions</b>		
<b>Description:</b> Instead of only testing on the user's understanding on each part of the car and what their purpose is, some questions should present a form of a situation that needs a user to react accordingly. Examples include prompting the user on the appropriate actions that should be taken when driving in the rain.		
<b>Requirement Level:</b> High		

Requirement #: F3.5	Requirement Type: Functional	Use Case #: -
<b>Requirement: The quiz section should provide the user a method to track their past scores</b>		
<b>Description:</b> In addition to allowing the user to test their understanding, there should be some form of tracking the user's past performance to monitor their progress. This would allow the user to track their learning and check to see how far they have improved.		
<b>Requirement Level:</b> Medium		

<b>Requirement #: F3.6</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: The quiz should have at least 50 question preloaded</b>		
<b>Description:</b> There should be at least 50 questions preloaded into the system, to allow there to be a random choice of questions. The questions should be of varying difficulty and there should be some consideration to ensure that when the questions are chosen, there is a balance of easy and difficult questions		
<b>Requirement Level:</b> High		

<b>Requirement #: F3.7</b>	<b>Requirement Type:</b> Functional	<b>Use Case #: -</b>
<b>Requirement: The system should be able to submit results to an online database and allow the user to see their global score</b>		
<b>Description:</b> With an internet connection, at the end of each quiz, the system should connect to a remote server and submit the user's score. It then should query the server and get the user's current position on the global leaderboard.		
<b>Requirement Level:</b> Low		

#### F4: Other functionalities

In addition to the abovementioned features, there are a number of technological features that should be met by the system

Requirement #: F4.1	Requirement Type: Functional	Use Case #: -
<b>Requirement: The software should have some form of multiregional support</b>		
<b>Description:</b> Due to the variations in terms used based on the location (i.e. gas instead of petrol in US English, compared to the latter in UK/Australian English); there is a need to adapt the software based on the language used. By default, the system should run in Australian English, but there should be a manual override to change the type of English used. If possible, an automatic detection of region could be done to automate this process		
<b>Requirement Level:</b> Medium		

Requirement #: F4.2	Requirement Type: Functional	Use Case #: -
<b>Requirement: There should be a thread monitor implemented in the system</b>		
<b>Description:</b> The system should have a thread monitor implemented, such that the user can invoke the thread monitor in order to see what are the currently running threads in the system.		
<b>Requirement Level:</b> High		

Requirement #: F4.3	Requirement Type: Functional	Use Case #: -
<b>Requirement: Each main feature (F1, F2, F3) should be running in their own threads</b>		
<b>Description:</b> The system must be designed such that from the main menu, upon choosing a mode to run the software, the system should run that mode in a separate thread of its own.		
<b>Requirement Level:</b> High		

Requirement #: F4.4	Requirement Type: Functional	Use Case #: -
<b>Requirement: Each component in a simulation should be running in its own thread</b>		
<b>Description:</b> In the case of a simulation, each component should be running in a thread of its own. For example, in the case of a fuel system, the engine should be in a thread of its own, the fuel line, fuel pump and fuel tank too should be in their own separate threads.		
We would expect to have at most 4-5 threads running at any one time.		
<b>Requirement Level:</b> High		

Requirement #: F4.5	Requirement Type: Functional	Use Case #: -
<b>Requirement: There should be an implementation of inter-thread communication</b>		
<b>Description:</b> For some features, there is a necessity for there to be some forms of inter thread communication. For example when simulating the fuel consumption test, the algorithm that calculates the fuel consumption needs to find a way to get the fuel level to be updated, and the fuel consumption reading to be updated. Due to this, we need a way to allow the threads to communicate among each other.		
<b>Requirement Level:</b> High		



Requirement #: F4.6	Requirement Type: Functional	Use Case #: -
<b>Requirement: The system should ensure that once the threads are no longer needed they should be properly ended</b>		
<b>Description:</b> In order to ensure the performance is maintained, and to prevent accidental conflicts, at the end of the use of a thread, once it is no longer needed, it should be immediately terminated, instead of waiting for the termination of the program to end the threads. This makes sure the chances of us having a runaway thread are greatly reduced.		
<b>Requirement Level:</b> High		

Requirement #: F4.7	Requirement Type: Functional	Use Case #: -
<b>Requirement: The software should be able to be controlled using a USB steering wheel</b>		
<b>Description:</b> The software should have the ability for the user to connect a USB steering wheel kit (usually comes with a set of pedals) and use this digital steering wheel to be able to control the software. This should be at least for the simulation section, while for the quiz and info graphics, the buttons on the steering wheel can be used to control the software.		
<b>Requirement Level:</b> Low		

Requirement #: F4.8	Requirement Type: Functional	Use Case #: -
<b>Requirement: The software can support the use of a VR device and Leap Motion.</b>		
<b>Description:</b> With the connection of a VR device to the computer, the software should be able to render a virtual reality mode, and send the video output to the VR device. This would give the user the ability to get close up to the car, and feel as though they are next to it.		
Leap motion can then be used to track the user's hand motion and use the gestures it captures to control the system.		
<b>Requirement Level:</b> Low		

Requirement #: F4.9	Requirement Type: Functional	Use Case #: -
<b>Requirement: There should be a companion version that runs on tablets</b>		
<b>Description:</b> In addition to a desktop application, there should be a version of the software that can take advantage of touch screens on tablets. The same functionalities should be available on the tablet edition however the UI has to be touch friendly.		
<b>Requirement Level:</b> Low		

## Non Functional Requirements

The following are the non-functional requirements for this system:

ID: NF1	Requirement: Accessibility	Priority: High
Description: The system should be able to run on all Windows 8 and newer PCs		
ID: NF2	Requirement: Ease of use	Priority: High
Description: A new user should be able to operate the system as long as they know how to use a computer, and should not need any training to learn how to use the system		
ID: NF3	Requirement: UI Design consistency	Priority: High
Description: The system UI should be consistent throughout the whole system, to ensure a user will not find it confusing when switching between sections of the system		
ID: NF4	Requirement: Code maintainability	Priority: Medium
Description: The code used to create this system should be highly maintainable, that is it should be easy to be modified and upgraded in the future		
ID: NF5	Requirement: Help and Support	Priority: High
Description: Adequate and clear instructions should be provided in the system, without the need of a help file, to prompt the user what they should do while using the program		
ID: NF6	Requirement: User Interface	Priority: High
Description: The UI designed for the system must be functional and enables users to easily use the system. Ease of use is essential in ensuring the system will appeal to a wide range of users. Complicated user interfaces only frustrate users, greatly reducing their satisfaction of the software, regardless of how advanced the software is.		