

3 sections

- A) Infographic of a car
 - A) User can see various cutaways of a car
 - B) Able to change views of a car (front, side, rear)
 - C) Allow user to obtain more information on the part of the car by clicking on the name of the part mapped to the car's actual part
 - D) Add links to online videos/animation showing the part of the car in action
- B) Simulating multiple situations and functions of a car
 - A) Add components to a car
 - B) Show the difference between front and all wheel drive
 - C) Simulation of fuel system (ie show how fuel passed to engine when accelerating)
 - D) Simulation of cooling system
 - E) Simulation of electrical system
 - F) Simulation of steering
 - G) Simulation of braking system
 - H) Adding attachments like trailer
 - I) Modelling different fuel systems (hybrid, hydrogen)
 - J) Simulating fuel consumption (flat land, going up and down a hill)
 - K) Able to change display facia of the car (change layout of instrument cluster)
 - L) Should show vehicle symbols (standardised symbols on instrument cluster light up as features are turned on)
 - M) Able to turn on and turn off parts of the car (lights, indicators, etc)
 - N) Active tracking of fuel, temperature and speed
- C) Quiz on parts of the car
 - A) Timed (~10 minutes for 10 question)
 - B) Give situations to a user, prompt for correct action (ie turning on lights when dark)
 - C) Should choose questions at random
 - D) Ask users to identify the correct part of the car
 - E) Leaderboard
 - F) Have at least 50 questions stored in the system
- D) Additional features
 - A) Multiple language support (different names for parts in different versions of english)
 - B) Have a thread monitor (show the passing of info among threads)
 - C) Should be intuitive and have an easy to use interface
 - D) Optimised system

3 sections

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 #: UC1_1
Requirement: A user is able to choose infographics of multiple cutaways of a car		
Description: 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		
Requirement Level: High		

Requirement #: F1.2	Requirement Type: Functional	Use Case #: UC1_1
Requirement: A user is able to rotate the infographic of the car		
Description: 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		
Requirement Level: High		

Requirement #: F1.3	Requirement Type: Functional	Use Case #: UC1_1
Requirement: A user is able to obtain more information of the part of the car		
Description: 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">• Part name• Purpose of the part• Other names of the part (i.e. a trunk is called a boot in different variations of English)		
Requirement Level: High		

Requirement #: F1.4	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Link to online video or animations showing the part of the car in action		
Description: 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		
Requirement Level: 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 #: UC1_1
Requirement: Adding components to the car		
Description: 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.		
Requirement Level: High		

Requirement #: F2.2	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate the difference between front wheel drive, rear wheel drive and all wheel drive		
Description: 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.		
Requirement Level: High		

Requirement #: F2.3	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate the fuel system in a car in different situations		
Description: 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"> • Engine startup • Acceleration • Cruising/Idling <p>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.</p>		
Requirement Level: High		

Requirement #: F2.4	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate the cooling system in the car		
Description: 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"> • High temperature • Normal/Low temperature 		
Requirement Level: Medium		

E) Simulation of electrical system (lights, wipers, electronics in the car)

Requirement #: F2.5	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate the electrical system in the car		
Description: 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"> • Lights • Windscreen wipers • Audio system 		

- Cruise control
- Rear defrost

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.

Requirement Level: High

Requirement #: F2.6	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate steering system in the car		
Description: 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)		
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)		
Requirement Level: Medium		

Requirement #: F2.7	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate braking system in the car		
Description: 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.		
Requirement Level: High		

Requirement #: F2.8	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate adding attachments to a car		
Description: 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		
Requirement Level: Medium		

Requirement #: F2.9	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate alternative fuel powered drivetrain systems		
Description: 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		
Example of drivetrains that should be demonstrated include: <ul style="list-style-type: none"> • Petrol-electric hybrid drivetrains • Hydrogen drivetrains • Electric drivetrains 		
Requirement Level: High		

Requirement #: F2.10	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Simulate the difference in driving conditions on fuel consumption		
Description: 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:

- Uphill driving
- Downhill driving
- Flatland driving
- Driving at high altitude

Requirement Level: High

Requirement #: F2.11	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Allow multiple faces for the instrument cluster		
Description: 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		
Requirement Level: Medium		

Requirement #: F2.12	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Instrument cluster must always be displayed		
Description: 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.		
Requirement Level: High		

Requirement #: F2.13	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Active tracking of car's temperature, fuel and speed		
Description: 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.		
Requirement Level: High		

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 #: UC1_1
Requirement: Quiz should be timed		
Description: 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.		
Requirement Level: High		

Requirement #: F3.2	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The quiz should preselect the questions that will be asked randomly from a question bank		
Description: 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.		
Requirement Level: High		

Requirement #: F3.3	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Questions should include testing the user's understanding on the parts of the car		
Description: Since the system mainly demonstrates the various parts of a car, the user should be		
Requirement Level: High		

Requirement #: F3.4	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The quiz should have situation based questions		
Description: 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.		
Requirement Level: High		

Requirement #: F3.5	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The quiz section should provide the user a method to track their past scores		
Description: 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.		
Requirement Level: Medium		

Requirement #: F3.6	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The quiz should have at least 50 question preloaded		
Description: 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		
Requirement Level: High		

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 #: UC1_1
Requirement: The software should have some form of multiregional support		

Description: Due to the variations in terms used based on the location (ie 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
Requirement Level: Medium

Requirement #: F4.2	Requirement Type: Functional	Use Case #: UC1_1
Requirement: There should be a thread monitor implemented in the system		
Description: 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.		
Requirement Level: High		

Requirement #: F4.3	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Each main feature (F1, F2, F3) should be running in their own threads		
Description: 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.		
Requirement Level: High		

Requirement #: F4.4	Requirement Type: Functional	Use Case #: UC1_1
Requirement: Each component in a simulation should be running in its own thread		
Description: 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.		
Requirement Level: High		

Requirement #: F4.5	Requirement Type: Functional	Use Case #: UC1_1
Requirement: There should be an implementation of inter-thread communication		
Description: For some features, there is a necessity for there to be some form 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.		
Requirement Level: High		

**this interthread comm can be done by having say 4 threads, one for the fuel level, one for the fuel consumption, and one that does the calculation. The calculation locks the global fuel level and fuel consumption value, updates it with a new one, while the fuel level and FC thread repeatedly check the value in the global variable, and update the displayed value accordingly.

Requirement #: F4.6	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The system should ensure that once the threads are no longer needed they should be properly ended		
Description: 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.		
Requirement Level: High		

Requirement #: F4.7	Requirement Type: Functional	Use Case #: UC1_1
Requirement: The UI designed for the system must be functional and enables users to easily use the system		
Description: 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.		
Requirement Level: High		