**Infographics of car**

|  |  |
| --- | --- |
| **Name:** Change perspective of car | **ID:** 1 |
| **Stakeholders and Goals:** User – to view different parts of the car | |
| **Description:** A user wants to view different parts of the car | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view different parts of the car. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Infographics” button. Upon clicking on Infographics, the user will be brought to the infographics page where the user will be able to view the picture from the corner perspective (the default view). 2. To change to a different perspective (i.e to see a different part of a car), the user will then click on an arrow button which will be located above the description box, either to the left or to the right. If the user wishes to zoom in a perspective the up and down buttons can be used. The user will be able to view different components from different perspectives of the car. Different components will surround the picture of the different perspective of the car (View user manual for an example of interface). 3. Step 2 is repeated if the user wishes to see another perspective of the car. 4. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Obtain more information on part | **ID:** 2 |
| **Stakeholders and Goals:** User – to get more information on a component | |
| **Description:** A user wants to know more about a specific component of the car | |
| **Actors:** User | |
| **Trigger:** User runs the application and wants to know more about a specific part of the car | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Infographics” button. Upon clicking on Infographics, the user will be brought to the infographics page where the user will be able to view the corner perspective of the car (the default view). 2. The user will choose a perspective of the car he/she wishes to view. 3. Upon selecting a perspective that users want (except the corner perspective), to obtain more information on a particular component, the user clicks on the name of the component. The user can also click on the part on the picture. 4. The system will then display the description of the component on the description box. 5. Step 2 to 4 are repeated if the user wishes to see another component of the car. 6. Upon completion of this use case (Use Case ID: 3), the next use case **can/will** be triggered (\*\*The reason behind “can/will” is due to the ability of the user to choose if he/she wants to run the use case or not). | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** View videos | **ID:** 3 |
| **Stakeholders and Goals:** User – to view videos on a component | |
| **Description:** A user wants to gain more information on a component by viewing a video prepared by the developers. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view a video on a specific component of the car | |
| **Normal Flow:**   1. This use case is continued from the previous use case. Under the description box, the user will be able to click on a video link at the end. 2. The system will then pop up an overlay with the video but will not start playing the video 3. The user clicks “Play”. 4. The system starts playing the video. 5. Upon finishing the video or in the middle of watching the video, the user will be able to click the outer surroundings of the video overlay to go back to the Infographics screen. 6. If the user wishes to view a video for a different component, the user will have to run the previous use case again (Use Case ID: 2). 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (IF overlay cannot be done on the software itself)  2. The system will then prompt the user if he/she wishes to launch his/her default media player.  3. The user enters his/her input  4. The video will be played in an external player and upon completion the media player will be closed and the user will be brought back to the Infographics page of the system. | |

**Simulation of car**

1. **Persistent dashboard**

|  |  |
| --- | --- |
| **Name:** Start/Stop engine | **ID:** 4 |
| **Stakeholders and Goals:** User – to start/stop the engine | |
| **Description:** A user wants to start or stop an engine | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to start or stop the engine | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 3. The system will show that the engine has started, by showing different components being lit up on the dashboard. 4. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user wishes to turn off the engine instead)   1. On the persistent dashboard the user will be able to see the “Stop Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to stop the engine. 2. The system will show that the engine has stopped, by showing different components being dim down on the dashboard.   (The user wishes to turn off the engine while the car is still moving)   1. On the persistent dashboard the user will be able to see the “Stop Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to stop the engine. 2. The system will provide a warning to the user that the car is still moving and the engine cannot be stopped until the car has been turned off | |

|  |  |
| --- | --- |
| **Name:** Increase/Decrease incline | **ID:** 5 |
| **Stakeholders and Goals:** User – to increase/decrease the incline | |
| **Description:** A user wants to increase or decrease the degree of incline of the car. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to increase/decrease the incline | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 3. The system will show that the engine has started, by showing different components being lit up on the dashboard. 4. To increase the degree of incline, the user simply clicks on the “+” symbol under the “Incline” section. 5. The system will then display an increase in the degree of incline. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user is trying to decrease the degree of incline)   1. To decrease the degree of incline, the user simply clicks on the “-” symbol under the “Incline” section. 2. The system will then display a decrease in the degree of incline. | |
| **Name:** Increase/Decrease speed | **ID:** 6 |
| **Stakeholders and Goals:** User – to increase/decrease the speed | |
| **Description:** A user wants to increase or decrease the speed of the car. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to increase/decrease the speed of the car. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 3. The system will show that the engine has started, by showing different components being lit up on the dashboard. 4. To increase the speed of the car, the user simply clicks the “+” symbol under the “Speed” section. 5. The system will then display an increase in speed of the car on the dashboard. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user is trying to decrease the degree of incline)   1. To decrease the speed of the car, the user simply clicks on the “-” symbol under the “Speed” section. 2. The system will then display a decrease in the speed of the car on the dashboard. | |

|  |  |
| --- | --- |
| **Name:** Increase/Decrease altitude | **ID:** 7 |
| **Stakeholders and Goals:** User – to increase/decrease the altitude | |
| **Description:** A user wants to increase or decrease the altitude that the car is currently at. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to increase/decrease the altitude that the car is currently at. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 3. The system will show that the engine has started, by showing different components being lit up on the dashboard. 4. To increase the altitude that the car is currently at, the user simply clicks the “+” symbol under the “Altitude” section. 5. The system will then display an increase in altitude that the car is currently at. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user is trying to decrease the degree of incline)   1. To decrease the altitude that the car is currently at, the user simply clicks on the “-” symbol under the “Altitude” section. 2. The system will then display a decrease in altitude that the car is currently at. | |

|  |  |
| --- | --- |
| **Name:** View car’s temperature, fuel and speed | **ID:** 8 |
| **Stakeholders and Goals:** User – to view the car temperature, fuel and speed | |
| **Description:** A user wants to view the temperature, fuel level and speed of the car. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view the current car temperature, fuel level and speed. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 3. The system will show that the engine has started, by showing different components being lit up on the dashboard. 4. The user will also be able to see on the car dashboard, from the left and going in order, the first values represent the fuel level of the car, the second value represent the speed of the car and the final values represent the temperature of the car. 5. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

1. **Fuel system simulation**

|  |  |
| --- | --- |
| **Name:** Simulate normal consumption | **ID:** 9 |
| **Stakeholders and Goals:** User – to simulate normal consumption of fuel of the car | |
| **Description:** A user wants to view the consumption of fuel of the car under normal circumstances. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view the normal consumption of fuel of the car. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the list of simulations available to the user, the user will then click on “Fuel system”. 3. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 4. The system will show that the engine has started, by showing different components being lit up on the dashboard. 5. The user will then be able to see the fuel flowing from the fuel tank to the fuel pump and to the engine. 6. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Simulate low fuel | **ID:** 10 |
| **Stakeholders and Goals:** User – to simulate low fuel | |
| **Description:** A user wants to view how the car responds when the fuel level is low. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view the response of the car when the level of fuel is low. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the list of simulations available to the user, the user will then click on “Fuel system”. 3. The user will then click on “Simulate low level” button in the simulation window. 4. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 5. The system will show that the engine has started, by showing different components being lit up on the dashboard. 6. The system will then light up the fuel symbol on the persistent dashboard. The fuel symbol lighting up on the dashboard represents the fuel level is low. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Simulate wrong fuel | **ID:** 11 |
| **Stakeholders and Goals:** User – to simulate wrong fuel | |
| **Description:** A user wants to view the effects of adding the wrong fuel into the car. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to simulate wrong fuel being added to the car. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the list of simulations available to the user, the user will then click on “Fuel system”. 3. The user then clicks on “Simulate wrong fuel” on the simulation window. 4. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 5. The system will show that the engine has started, by showing different components being lit up on the dashboard. 6. After 3 seconds, the system will then shut off the engine, by turning off the lights on the persistent dashboard. This is to show that the engine has broken down from adding the wrong fuel into the car. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** Simulate fuel consumption in different altitude | **ID:** 12 |
| **Stakeholders and Goals:** User – to simulate fuel consumption in different altitude | |
| **Description:** A user wants to view the difference in the rate of fuel consumption in different altitudes. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to simulate fuel consumption in different altitude. | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the list of simulations available to the user, the user will then click on “Fuel system”. 3. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 4. The system will show that the engine has started, by showing different components being lit up on the dashboard. 5. The user will then increase the current altitude by clicking on the “+” symbol under the “Altitude” section (This is closely related to the increase/decrease altitude use case, Use case ID: 7). 6. The system will then show the user the increased rate of consumption of fuel by showing more arrows flowing from the fuel tank, to the fuel pump and to the engine. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user decides to decrease the current altitude instead)   1. The user will then decrease the current altitude by clicking on the “-” symbol under the “Altitude” section (This is closely related to the increase/decrease altitude use case, Use case ID: 7). 2. The system will then show the user the decreased rate of consumption of fuel by showing less arrows flowing from the fuel tank, to the fuel pump and to the engine. | |

|  |  |
| --- | --- |
| **Name:** Simulate fuel consumption in different incline | **ID:** 13 |
| **Stakeholders and Goals:** User – to simulate fuel consumption in different degree of incline | |
| **Description:** A user wants to view the difference in the rate of fuel consumption in different degrees of incline. | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to simulate fuel consumption in different degrees of incline | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. On the list of simulations available to the user, the user will then click on “Fuel system”. 3. On the persistent dashboard the user will be able to see the “Start Engine” button (What the user may see may differ depending if the engine of the car is on/off). The user will then click on it to start the engine. 4. The system will show that the engine has started, by showing different components being lit up on the dashboard. 5. The user will then increase the degree of incline by clicking on the “+” symbol under the “Incline” section (This is closely related to the increase/decrease incline use case, Use case ID: 5). 6. The system will then show the user the increased rate of consumption of fuel by showing more arrows flowing from the fuel tank, to the fuel pump and to the engine. 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  (The user decides to decrease the current degree of incline instead)   1. The user will then decrease the degree of incline by clicking on the “-” symbol under the “Incline” section (This is closely related to the increase/decrease altitude use case, Use case ID: 5). 2. The system will then show the user the decreased rate of consumption of fuel by showing less arrows flowing from the fuel tank, to the fuel pump and to the engine. | |

|  |  |
| --- | --- |
| **Name:** View fuel system in different situations | **ID:** ??? |
| **Stakeholders and Goals:** User – to view the fuel system under different situations | |
| **Description:** A user wants to understand the fuel system of a car in different situations, such as when the fuel is low, filled with incorrect fuel, when its running normally or on an inclined road | |
| **Actors:** User | |
| **Trigger:** User wants to understand the fuel system in different situations | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations. 2. The user clicks on “View fuel system in different situations”. The window on the right side of the persistent dashboard and list of simulations will be updated. The user will then able to see different components on the screen which consists of the Engine, Fuel Tank, and the Fuel Pump. There will also be 4 buttons that the user can click on which is “Low Fuel”, “Incorrect Fuel”, and “Normal Mode” (Default is “Normal Mode”). An extra description box will also explain to the user that the change in the degree of incline can also affect the fuel consumption. 3. The user will then choose to click on one of the buttons “Low Fuel”, “Incorrect Fuel” or “Normal Mode”. 4. Upon 5. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

|  |  |
| --- | --- |
| **Name:** View difference among different wheel drive system | **ID:** ???? |
| **Stakeholders and Goals:** User – to view the difference among different wheel drive system | |
| **Description:** A user wants to understand the difference between four wheel drive, rear wheel drive and front wheel drive | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to view difference among different wheel drive system | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Simulation” button. The user will be brought to the simulation page where the user will be able to view the persistent dashboard and also a list of simulations which allows the user to understand how different components of a car come together and work together to execute a particular process. 2. The user clicks on “View difference among different wheel drive system”. The window on the right side of the persistent dashboard and list of simulations will be updated. The user will then able to see different components on the screen which consists of the Engine, Axle and Wheels. There will also be 3 buttons available for the user to click which is “4 wheel drive”, “Front wheel drive (2 wheel drive)” and “Rear wheel drive (2 wheel drive). 3. The user clicks one of the buttons, “4 Wheel drive”, “Front wheel drive (2 wheel drive)” and “Rear wheel drive (2 wheel drive). 4. The user will need to start the engine before being able to see the differences (See Start/Stop Engine use case, Use Case ID:4 ) 5. If the user clicks on “4 wheel drive”, the user will be able to see flows/arrows coming out from the engine and moving towards both the front axle and the back axle. If the user clicks on “Front wheel drive (2 wheel drive)” the user will be able to see flows coming out from the engine only towards the front axle. If the user clicks on “Rear wheel drive (2 wheel drive)”, the user will be able to see flows coming out from the engine only towards the back axle. 6. Steps 4 to 6 are repeated if the user wishes to view a different wheel drive system. Do note that before the user can change to a different wheel drive system, the user has to turn off the engine (See Start/Stop Engine use case, Use Case ID: 4) 7. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

**Quiz**

|  |  |
| --- | --- |
| **Name:** Take a quiz | **ID:** ????? |
| **Stakeholders and Goals:** User – to take a quiz | |
| **Description:** A user wants to take a quiz to test his/her understanding on car processes and components | |
| **Actors:** User | |
| **Trigger:** User runs the application and wishes to take a quiz | |
| **Normal Flow:**   1. User reaches the main page of the system and clicks on the “Quiz” button. Upon clicking the quiz button, the user will be brought into the quiz page. 2. The user will be prompted to choose the difficulty of the quiz he wishes to work on (At the moment there will be no difficulty, stretch goal). 3. Upon selecting the difficulty, a progress bar will appear on the screen to show that the system is preparing the questions to be asked for the quiz. The quiz will then start after the loading is complete. A timer will start. 4. The user will then answer the questions by clicking on the answer the user thinks that it is correct. The user will be able to use the keyboard as a form of input for the answers, numbers will represent each answer of the question (i.e 1 for A, 2 for B). 5. The user then clicks the next button 6. The system will show the next question. 7. The user will be able to have quick access to different questions by clicking on the quick access bar which will be placed above the question. 8. Steps 4 to 7 are repeated until the users have answered all the questions 9. Upon completion of all the questions available the user will be able to click on “Complete Quiz” button which will appear on the top right of the question box. 10. The user will then be brought to the “Review quiz” page, which will be the next use case. | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:**  **(After the quiz has been started from step 4 onwards)**   1. If the user wishes to quit halfway during the quiz, the user will be able to click on a “X” button which will be present on the top right of the screen. No statistics will be saved. 2. The systems prompts for reconfirmation to quit the quiz. 3. The user enters confirmation. 4. The system brings the user back to the main menu. | |

|  |  |
| --- | --- |
| **Name:** Review quiz | **ID:** ???? |
| **Stakeholders and Goals:** User – to review the quiz he/she has completed | |
| **Description:** A user wants to gain feedback (correct and wrongly answered questions) on the quiz that he/she has completed. | |
| **Actors:** User | |
| **Trigger:** User finishes the quiz and wishes to review his/her progress. | |
| **Normal Flow:**   1. Continuing from the previous use case, the user will now be on the “Review page” and user will be able to see his/her final score for the quiz. 2. The interface for reviewing the questions will be similar to the interface of the quiz. The user will then be able to click on the quick access bar to access different previously answered questions. Different colour on the quick access bar will represent if the user has answered the questions correctly or wrongly (Green for correctly answered questions, red for wrongly answered questions). 3. Upon entering a wrongly answered question, the answer that the user has chosen (which is the wrong one, if it is not clear enough) will be highlighted in red and the correct answer will be highlighted in green. If the user enters a question that he/she has already answered correctly, the user will see the correct answer (which is also the answer that he/she has chosen) in green. 4. The user will then be able to click the “X” button which will be on the top right corner to quit the quiz review and go back to the main menu. 5. End | |
| **Sub-Flows:** None | |
| **Alternative/Exceptional Flows:** None | |

**Change log**

|  |  |
| --- | --- |
| **Date** | **Changes made** |
| 16/5/2015 | * Use case 1, updated to zoom in, use the up and down * Use case 2 updated * Use case 3 update to be linked from previous use case. Since video links can only been seen under the description of the video. Updated alternative * Use case 4 added * Use case 5 updated, updated from number 4 to 5, updated on what the user needs to do in order to view different wheel drive system * Use case “Take a quiz”, updated how one will take a quiz * Use case “Review quiz”, added today * Split use case into 3 different part and added change log |