|  |  |
| --- | --- |
| Client | A multinational automobile company |
| User | * The employees of the company that work in a car dealership. |
| Functional requirements | * R1: Register vehicles for sale. * R2: Calculate the sale price of a vehicle. * R3: Generate a report with the information of a vehicle. * R4: Generate a report with the data regarding the documentation of a vehicle. * R5: Generate a map of the parking lot that stores old vehicles. * R6: Generate a report of a car that is found within the old vehicle parking lot * R7: Find the percentage to which the old vehicle parking lot is filled. |
| Context | A multinational specialize in the dealership of vehicles has contracted you to create an application to manage the cars found in a vehicle dealership they are planning to open in Cali, Colombia. |
| Non-functional requirements | * Rn1: It must be fast. * Rn2: It must be efficient. * Rn3: It must be safe and secure. * Rn4: It must not lose the date that is stored in it. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | * R1: Register vehicles for sale. | | |
| Summary | The application must be able to register a vehicle, with all the data related to it, including but not limited to: Vehicle type, brand, model, Fuel type, documentation and whether is used or not. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Brand | String | None |
| Model | int | None |
| Used status | String | Condition outside of answer domain (different from yes or no) |
| Vehicle type | int | To be inputted before registration, invalid if outside of supported vehicles |
| Cylinder capacity | Double | None |
| Base price | Double | None |
| License plate | String | Only to be inputted if the vehicle is used |
| Property card info | String | Only to be inputted if the vehicle is used |
| Soat info | String | To be inputted if it exists |
| Technical mechanical revision info | String | To be inputted if it exists |
| Fuel Type | String | If the value inputted is outside of the possible fuel types |
| Mileage | Double | Only to be inputted if the vehicle is used |
| Number of doors | Int | Only to be inputted if the vehicle is of the type “Car” |
| Window type | Int | Only to be inputted if the vehicle is of the type “Car”, to be repeated if the input is outside of the possible window types (tinted, non-tinted) |
| Charger type | Int | Only to be inputted if the vehicles id of the types “Car/Electric” or “Car/Hybrid” |
| Brand | String | None |
| General activities needed for completion | 1. Ask the data to the user. 2. Reads the data. 3. The data is saved. | | |
| Results | A car is registered in the car database | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
|  |  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | * R2: Calculate the sale price of a vehicle. | | |
| Summary | The application must be able to search for a vehicle and calculate the sale price for said vehicle depending on its type, used status, and documentation status. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Car id | String | Nonexistent car |
| General activities needed for completion | * 1. Read the vehicle id.   2. Search for the vehicle id in the vehicle database.   3. Calculate the sale price of the found vehicle.   4. Print the sale price to the user. | | |
| Results | The sale price is displayed on-screen | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
| Sale price | Double | If the car does not exist |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | * R3: Generate a report with the information of a vehicle. | | | |
| Summary | The application must be able to find a group of vehicles meeting a certain criteria (Vehicle type, Fuel type, used status), and generate a report with all the information regarding every vehicle. | | | |
| Inputs | **Input name** | **Data type** | | **Retry condition** |
| Criteria | String | | Criteria outside of domain. |
| General activities needed for completion | 1. Ask the criteria to the user 2. Find the vehicles that meet said criteria. 3. Print a report on all of the vehicles found | | | |
| Results | A summary of several vehicle’s information is displayed on screen | | | |
| Outputs | **Output name** | **Data type** | | **Retry condition** |
| report | String | | No vehicles met the condition |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | * R4: Generate a report with the data regarding the documentation of a vehicle. | | |
| Summary | The application must be able to search for a vehicle and generate a report on the state of the documentation of said vehicle. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Vehicle id | String | Nonexistent vehicle |
| General activities needed for completion | * 1. Asks for the vehicle id.   2. Finds the vehicle.   3. Create a report with the documentation.   4. Print the reporu | | |
| Results | A car is registered in the car database | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
| Documentation report | String | Nonexistent vehicle |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | R5: Generate a map of the parking lot that stores old vehicles. | | |
| Summary | The application must be able to create a visual representation of the parking lot where the old vehicles are store, informing the spaces that are occupied and the ones that aren’t | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| General activities needed for completion | * + 1. Find the empty and filled spaces of the parking lot.     2. Make a visual representation of the lot.     3. Print the visual representation. | | |
| Results | An image of the parking lot is displayed to the user | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
| Image of the parking lot | String | none |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | R6: Generate a report of a car that is found within the old vehicle parking lot. | | |
| Summary | The application must be able to generate a report of the vehicles that meet a certain criteria (a certain range of years, being the oldest, being the newest) and are part of the parking lot. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Criteria | String | Criteria outside the domain. |
| General activities needed for completion | * + - 1. Look for the vehicles in the parking lot that meet the condition.       2. Get a report of those vehicles.       3. Display the report | | |
| Results | The report of the desired cars is displayed | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
| Report | String | None |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | R7: Find the percentage to which the old vehicle parking lot is filled. | | |
| Summary | The application must be able to find the percentage to which the parking lot has been filled | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| General activities needed for completion | * + - * 1. Find out how many vehicles are in the parking lot.         2. Calculate the percentage comparing it to the capacity.         3. Print the result | | |
| Results | A car is registered in the car database | | |
| Outputs | **Output name** | **Data type** | **Retry condition** |
| Percentage occupied | double | None |