**Project Report**

**System Application**

*SDV601*

*By Travis Byrman*

**Contents**

[Progress Report One 3](#_Toc75185008)

[Wireframe 3](#_Toc75185009)

[Error Prompt 1: Vehicle activity removal confirmation 3](#_Toc75185010)

[Error Prompt 2: Vehicle removal confirmation 3](#_Toc75185011)

[Error Prompt 3: Invalid input 4](#_Toc75185012)

[Error Prompt 4: Empty fields 4](#_Toc75185013)

[Error Prompt 5: Taken registration number 4](#_Toc75185014)

[Application Mock-up 5](file:///C:\Users\tbyrm\Documents\NMIT\SDV601\SDV%20Project%20Documentation.docx#_Toc75185015)

[Use-Cases 6](#_Toc75185016)

[Use-case priority 6](#_Toc75185017)

[Use-case descriptions 7](#_Toc75185018)

[Progress Report Two 10](#_Toc75185019)

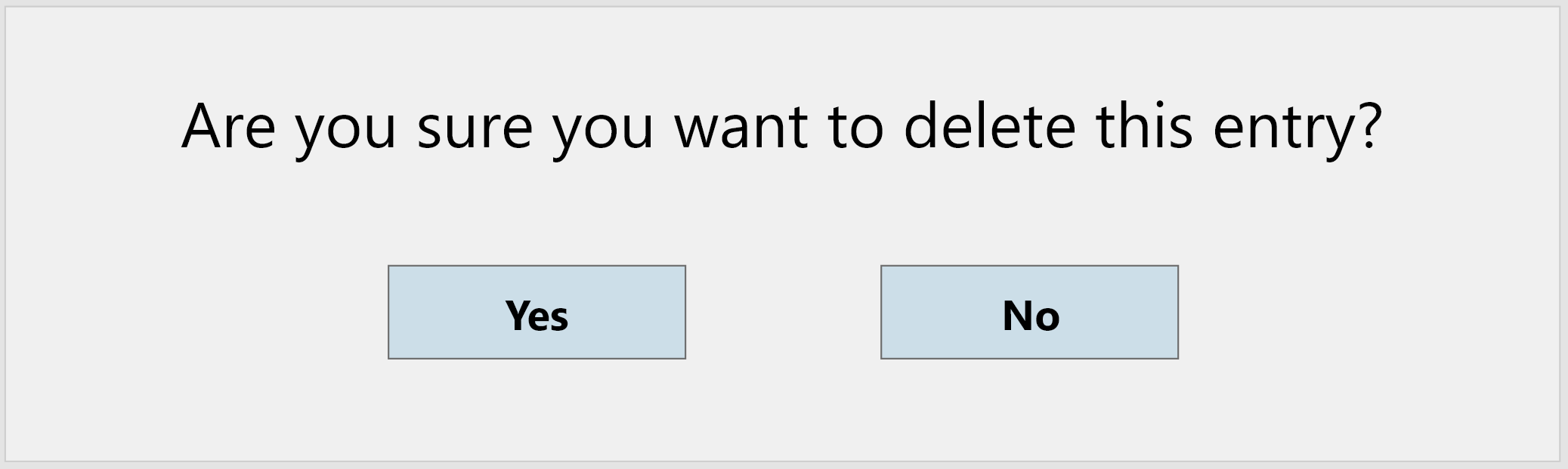
[Conceptual UML Class Diagram 10](#_Toc75185020)

[Package Diagram 11](#_Toc75185021)

# Progress Report One

## Wireframe

Error Prompt 1: Vehicle activity removal confirmation



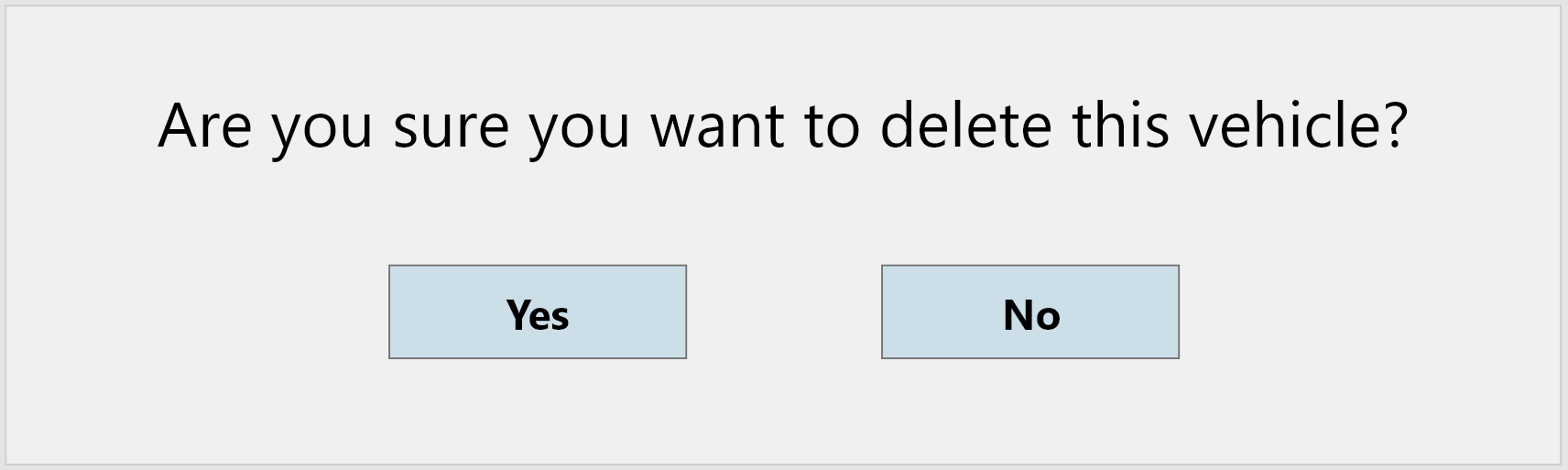
1.0 **Yes confirmation button** – This button will delete the selected vehicle activity and return the user to the main form.

1.1 **No confirmation button** – This button will not delete the selected vehicle activity and return user to the view vehicle form.

1.0

1.1

Error Prompt 2: Vehicle removal confirmation



2.0 **Yes confirmation button** – This button will delete the selected vehicle and return the user to the main form.

2.1 **No confirmation button** – This button will not delete the selected vehicle and returns the user to the view vehicle form.

2.1

2.0

Error Prompt 3: Invalid input



3.0 **Confirmation button** – This button closes the prompt and returns the user to the form.

3.0

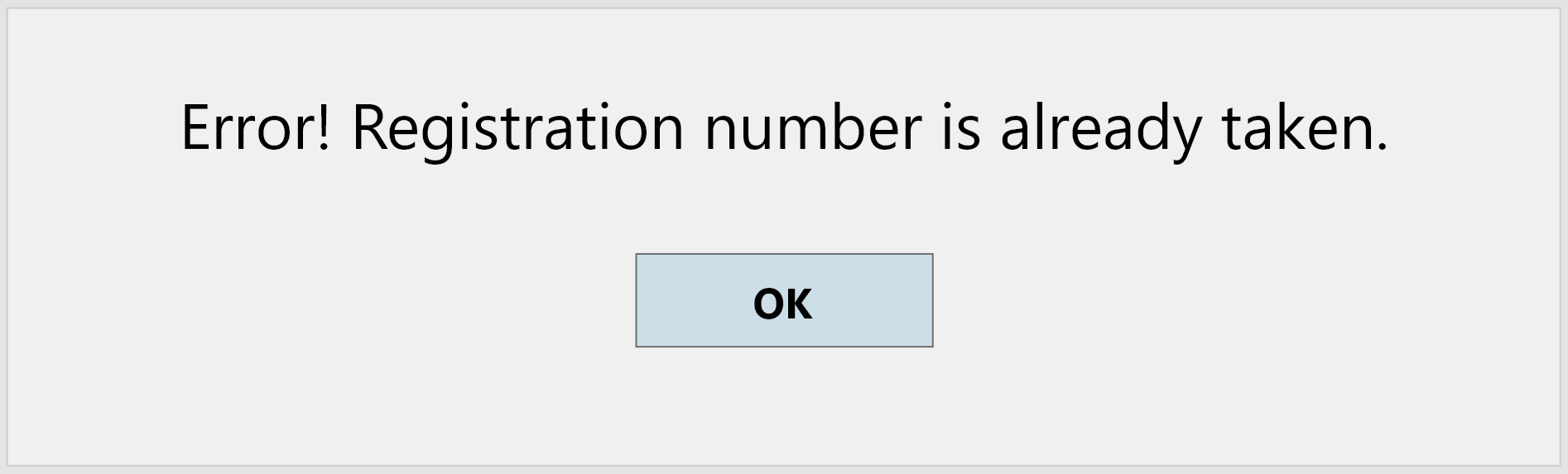
Error Prompt 4: Empty fields



4.0 **Confirmation button** – This button closes the prompt and returns the user to the form.

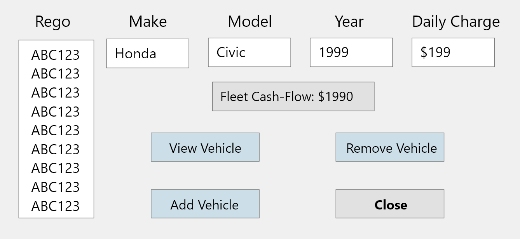
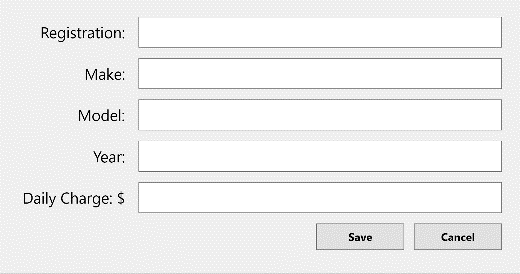
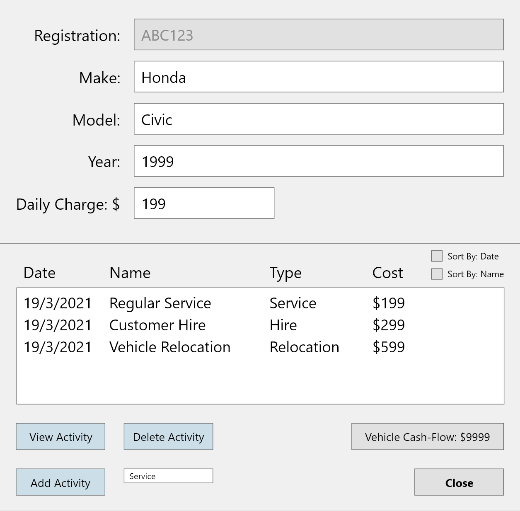
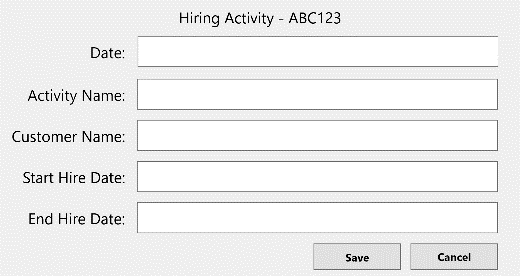
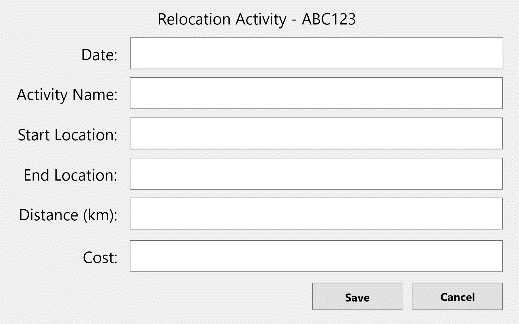
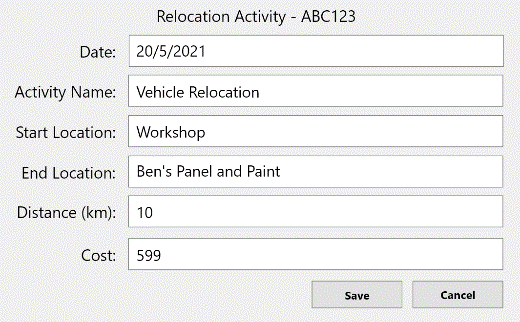
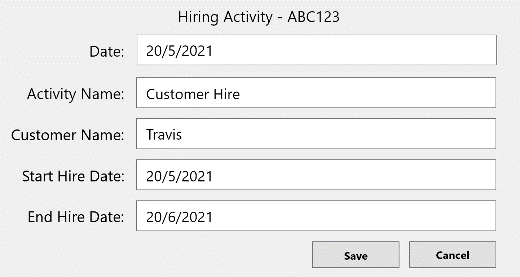
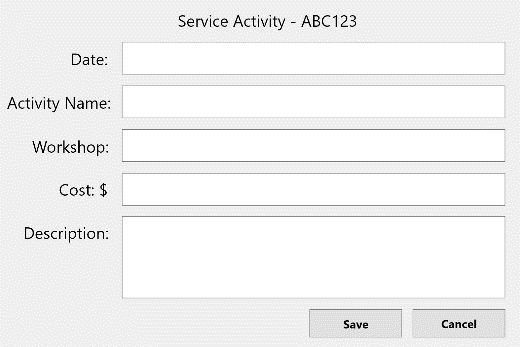
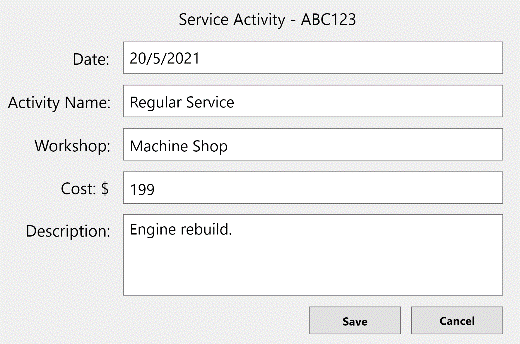
4.0

Error Prompt 5: Taken registration number



5.0 **Confirmation button** – This button closes the prompt and returns the user to the form.

5.0



Date fields using a Date/Time picker.

ComboBox that selects specific activity type

One edit vehicle activity form will be shown and vary depending on which activity type is selected.

One add vehicle activity form will be shown and vary depending on which activity type is selected.

Radio buttons, only one can be activated, when pressed it sorts the activities by either name or date.

ViewServiceActivity

Confirmation prompt will be displayed (explained above).

Confirmation prompt will be displayed (explained above).

AddServiceActivity

ViewHiringActivity

ViewRelocationActivity

AddRelocationActivity

AddHiringActivity

ViewVehicle

AddVehicle

MainForm

Application Mock-up

## Use-Cases

Use-case priority

Below are the use-cases for this system ordered by its priority.

|  |  |  |
| --- | --- | --- |
| **Priority** | **Use-Case** | **Reason** |
| 1 | Edit Vehicle | This use case is responsible for editing/maintaining a vehicle record. This is an important use-case as it allows users to edit vehicle details that have been created. This minimises miscommunication and input errors, allowing users to correct mistakes. |
| 2 | Edit Activity | This use case is responsible for editing activities created for vehicles. This is an important use-case as it allows users to edit activities that have been created. This minimises miscommunication and input errors, allowing users to correct mistakes. |
| 3 | Add Vehicle | This use-case is responsible for creating a vehicle and adding it to the vehicle list. This is an important use-case as other use-cases rely on this use-case to function and work as intended. The whole application would not work if a vehicle cannot be created. |
| 4 | New Activity | This use case is responsible for creating vehicle activities and adding them to an activity log. This is an important use-case as it allows users to create activities for vehicles that have been created. Without this use-case, users would not be able to create activity logs for each vehicle. |
| 5 | Quick View | This use case is responsible for allowing users to quickly view vehicle details efficiently. This is an important use case as it maximises the productivity a user can have with the system. |
| 6 | Delete Vehicle | This use case is responsible for allowing users to delete vehicles. This is an important use case as it allows users to remove vehicles that are no longer in their fleet. This declutters the system/application as it removes any redundant data that users no longer need to keep. |
| 7 | Delete Activity | This use case is responsible for allowing users to delete vehicle activities. This is an important use case as it allows users to remove activities that have been added to the wrong vehicle. This declutters the system/application as it removes any redundant data that users no longer need to keep. |
| 8 | Re-Sort Log | This use case is responsible for sorting the vehicle activities log. This makes activities easy to find as they can be resorted based on their date or name. |
| 9 | Save | This use case is responsible for saving the work/data created by users in the system. This allows vehicles and activities to be stored, so that when the system is opened again, the data that was previously created/edited can be restored. |
| 10 | Retrieve | This use case is responsible for retrieving work/data created by users in the system. This allows vehicles and activities to be retrieved from the system and utilised from when the last edit was made. |

Use-case descriptions

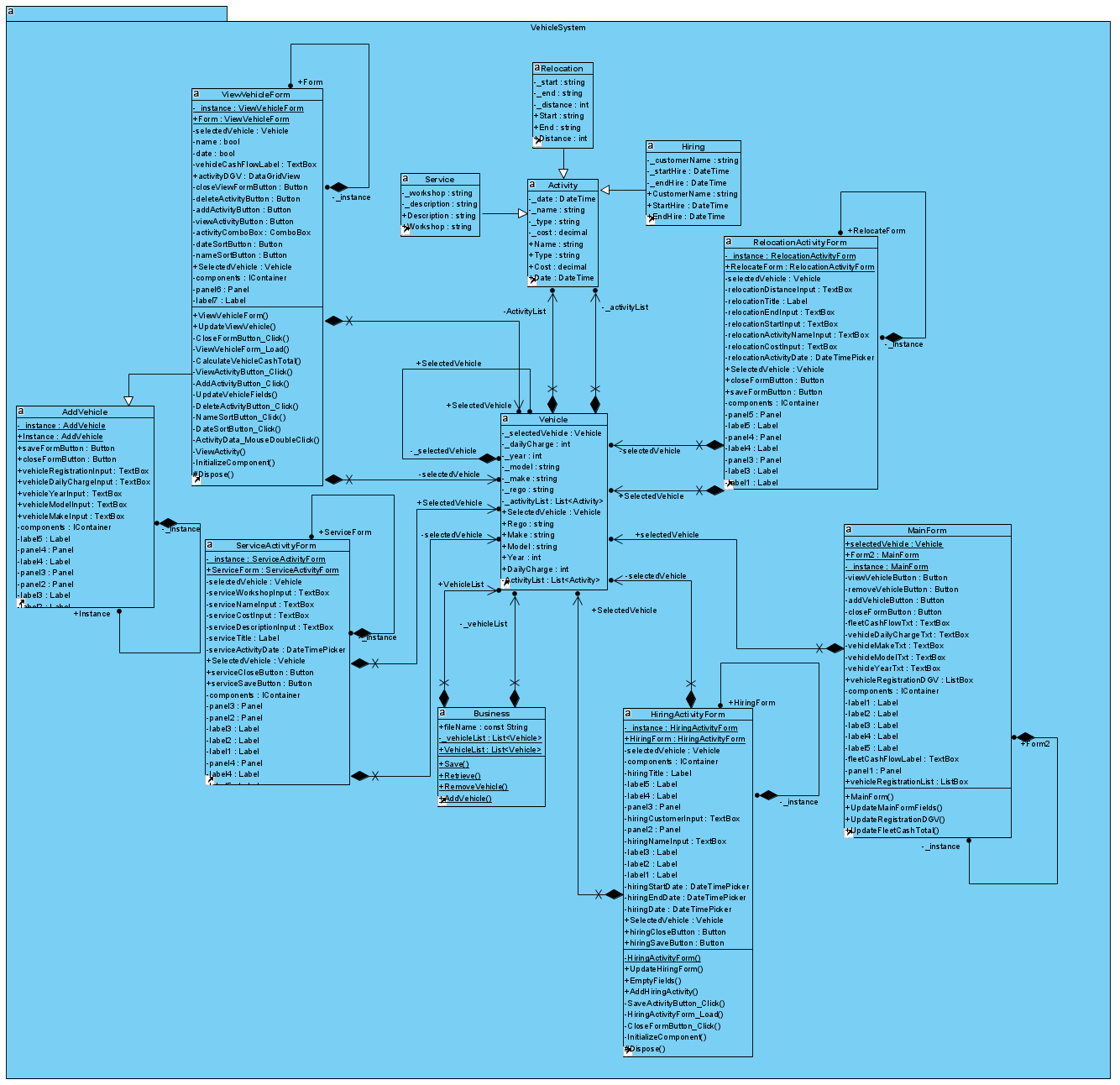
|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case** | | Edit Vehicle | |
| **Version / Date** | | 1.0 - 11 May 2021 | |
| **Author** | | Travis Byrman | |
| **Summary** | | A user edits the details of an existing vehicle | |
| **Priority** | | Medium. Other use cases depend on this. | |
| **Dependencies** | **Includes** |  | |
| **Included by** | Add Vehicle | |
| **Actor(s)** | | Admin staff member/user | |
| **Trigger** | | Selecting "Edit Vehicle" from the vehicle info menu. | |
| **Pre-Conditions** | | The vehicle has been created. | |
| **Description** | | 1. User enters new vehicle details. 2. System verifies new vehicle details. 3. System identifies valid input. 4. System notifies user of successful changes. | |
| **Extensions/ Exceptions** | | **Cause** | **Action** |
| User has entered invalid vehicle details. | An error message prompts the user of invalid vehicle details and asks the user to try again. The vehicle change is unsuccessful. |
| The registration number that the user has entered has already been taken. | An error messages prompts the user that an existing vehicle is created that uses the same registration number. The vehicle change is unsuccessful. |
| **Illustrations** | | Screenshots on page 3. | |
| **Post-Conditions** | | The vehicles details have successfully been changed. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case** | | Delete Vehicle | |
| **Version / Date** | | 1.0 - 1 May 2021 | |
| **Author** | | Travis Byrman | |
| **Summary** | | A user deletes a vehicle record. | |
| **Priority** | | Low. Other use cases do not depend on this. | |
| **Dependencies** | **Includes** |  | |
| **Included by** |  | |
| **Actor(s)** | | Admin staff member/user | |
| **Trigger** | | Selecting "Remove Vehicle" from the vehicle info menu, after selecting a vehicle from the list in the vehicle info menu. | |
| **Pre-Conditions** | | The vehicle has been created. | |
| **Description** | | 1. System asks user to confirm action. 2. User confirms action with system. 3. System notifies user of successful record delete. | |
| **Extensions/ Exceptions** | | **Cause** | **Action** |
| User clicks ‘No when system asks for confirmation. | The system does not delete the vehicle and returns user to the vehicle info menu. |
| **Illustrations** | | Screenshots on page 3. | |
| **Post-Conditions** | | The vehicle is successfully deleted. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case** | | New Activity | |
| **Version / Date** | | 1.0 - 11 May 2021 | |
| **Author** | | Travis Byrman | |
| **Summary** | | A user creates a new activity. | |
| **Priority** | | High. Other use-case depends on this one. | |
| **Dependencies** | **Includes** | Edit Activity | |
| **Included by** | Edit Vehicle | |
| **Actor(s)** | | Admin staff member/user | |
| **Trigger** | | Selecting "Add Entry" from the view vehicle menu | |
| **Pre-Conditions** | | The vehicle already exists. | |
| **Description** | | 1. User enters new activity details. 2. System verifies new details. 3. System identifies valid input. 4. System notifies user of successful changes. | |
| **Extensions/ Exceptions** | | **Cause** | **Action** |
| User has entered invalid activity details. | An error message prompts the user of invalid activity details and asks the user to try again. The operation is unsuccessful. |
|  | | User has left an input field blank. | An error message prompts user of missing details and asks the user to try again. The operation is unsuccessful. |
| **Illustrations** | | Screenshots on page 4. | |
| **Post-Conditions** | | A new activity is successfully created. | |

# Progress Report Two

## Conceptual UML Class Diagram



## Package Diagram

