|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Log In | |
| Actor | Admin | |
| Preconditions | * The user must have account | |
| Postconditions | * None | |
| Exceptions | * If user entered wrong information * If user miss fill required fields | |
| Flow of Events | User Steps | System Steps |
| 1. The user will fill Log In fields | 1. The system will display dashboard |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add User | |
| Actor | Admin | |
| Preconditions | * The admin must have account | |
| Postconditions | * A success message will be displayed and a new user is added | |
| Exceptions | * If admin entered wrong information * If admin miss fill required fields | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select type of the new user 2. The admin will fill new user form | 1. The system will take new data and create new user 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit User | |
| Actor | User | |
| Preconditions | * The user must be logged in | |
| Postconditions | * A success message will be displayed and data is updated | |
| Exceptions | * If user entered wrong information | |
| Flow of Events | User Steps | System Steps |
| 1. The user will click on edit   3- The user will enter the new data | 1. The system will display a pop up form for editing   4- The System will return the page with the new data and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete User | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with user is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the type of user   3- The admin will select the user or group of users to delete | 1. The system will show all users with type selected   4- The system will return the page without the deleted users with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Users | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Users will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the type of users | 1. The system will show all users with type selected |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add User Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will enter the new type of users | 1. The system will receive and save it in data base |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit User Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new user type name | 1. The system will display a pop up form for editing   4- The System will return the page with the new user type and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete User Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with user type is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the type of user to delete | 1. The system will return the page without the deleted user type with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show User Types | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * User types will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show user types | 1. The system will display all user types |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Data Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required field | 1. The system will receive the data type and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Data Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the data type is updated | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new data type name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new data type and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Data Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with data type is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the data type to delete | 1. The system will return the page without the deleted data type with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Data Types | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Data types will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show data types | 1. The system will display all data types |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Country | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required field | 1. The system will receive the country name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Counrty | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new country name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new country name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new country name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Country | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with country is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the country to delete | 1. The system will return the page without the deleted country with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Countries | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All Countries will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show countries | 1. The system will display all countries |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add State | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which country he will add a new state to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the state name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit State | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new country name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new state name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new state name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete State | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with state is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the state to delete | 1. The system will return the page without the deleted state with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show States | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All States will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show states | 1. The system will display all states |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add City | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which state he will add a new city to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the city name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit City | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new country name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new city name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new city name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete city | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with city is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the city to delete | 1. The system will return the page without the deleted city with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show cities | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All cities will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show cities | 1. The system will display all cities |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Region | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which city he will add a new region to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the region name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit region | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new region name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new region name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new region name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Region | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with region is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the region to delete | 1. The system will return the page without the deleted region with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Regions | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All regions will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show regions | 1. The system will display all regions |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Street | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed * A new location will be added | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which region he will add a new street to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the street name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit street | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new street name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new street name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new street name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Street | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with street is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the street to delete | 1. The system will return the page without the deleted street with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Streets | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All streets will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show streets | 1. The system will display all streets |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Main Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the main type name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Main Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new Main Type name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new main type name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new main type name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Main Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with main type is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the main type to delete | 1. The system will return the page without the deleted main type with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Main Types | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All main types will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show main types | 1. The system will display all main types |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Sub Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which main type he will add a new sub type to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the main type name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Sub Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new Sub Type name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new sub type name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new sub type name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Sub Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with sub type is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the sub type to delete | 1. The system will return the page without the deleted sub type with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Sub Types | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All sub types will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show sub types | 1. The system will display all sub types |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Details Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which sub type he will add a new detail name to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the detail name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Detail Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new detail name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new detail name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new detail name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Detail Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with detail name is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the detail name to delete | 1. The system will return the page without the deleted detail name with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Detail Names | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All detail names will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show detail names | 1. The system will display all detail names |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Details Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which detail name he will add a new detail value to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the detail value and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Detail Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new detail value | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new detail value | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new detail value and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Detail Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with detail value is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the detail value to delete | 1. The system will return the page without the deleted detail value with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Detail Values | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All detail values will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show detail values | 1. The system will display all detail values |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Operation | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the operation name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Operation | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the operation name is updated | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new operation name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new operation and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Operation | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with operation is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the operation to delete | 1. The system will return the page without the deleted operation with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Operations | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All operations will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show operations | 1. The system will display all operation |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Operation Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the operation type and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Operation Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new operation type | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new operation type | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new operation type and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Operation Type | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with operation type is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the operation type to delete | 1. The system will return the page without the deleted operation type with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Operation Types | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All operation types will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show operation types | 1. The system will display all operation types |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Operation Type Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which operation type he will add a new operation type name to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the operation type name and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Operation Type Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new operation type name | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new operation type name | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new operation type name and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Operation Type Name | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with operation type name is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the operation type name to delete | 1. The system will return the page without the deleted operation type name with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Operation Type Names | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All operation type names will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show operation type names | 1. The system will display all operation type names |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Operation Type Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If admin entered data already exists * If the entered data does not match the required format * If admin didn’t select which operation type name he will add a new operation type value to. | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will fill required fields | 1. The system will receive the operation type value and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Operation Type Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the new operation type value | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new operation type value | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new operation type value and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Operation Type Value | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with operation type value is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the operation type value to delete | 1. The system will return the page without the deleted operation type value with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Operation Type Values | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All operation type values will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show operation type values | 1. The system will display all operation type values |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Add Item | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will add the owner of the item 2. The admin will add the location of the item 3. The admin will fill the required fields in the steps of adding item | 1. The system will receive the information and save it 2. The system will display success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Edit Item | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * Success message will be displayed and the is updated | |
| Exceptions | * If the entered data does not match the required format | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on edit   3- The admin will enter the new details | 1. The system will display a pop up form for editing with the current data   4- The System will return the page with the new details and display a success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Delete Item | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * A success message will be displayed with item is deleted | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will select the item to delete | 1. The system will return the page without the deleted item with success message |

|  |  |  |
| --- | --- | --- |
| Use Case Id |  | |
| Use Case Name | Show Items | |
| Actor | Admin | |
| Preconditions | * The admin must be logged in | |
| Postconditions | * All items will be displayed | |
| Exceptions | * None | |
| Flow of Events | User Steps | System Steps |
| 1. The admin will click on show items | 1. The system will display all items |

|  |  |
| --- | --- |
| Scenario Id | 1 |
| Scenario name | Add Datatype |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “other”, click “add data type”, so the system will display to the index page of data type, the admin will enter new datatype and click “add”, the system will redirect to the same page with success message: “Data type created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing datatype the system will show error message that “Datatype already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 2 |
| Scenario name | Edit Datatype |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, the system will display the index page of data type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this datatype, the system will redirect the same page with the edit of the data type name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing datatype the system will show error message that “Datatype already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 3 |
| Scenario name | delete Datatype |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, so the system will display the index page of data type and show all data types, the admin has to select the data type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted data type. |

|  |  |
| --- | --- |
| Scenario Id | 4 |
| Scenario name | Show Datatype |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, so the system will display the index page of data type and show all data types. |

|  |  |
| --- | --- |
| Scenario Id | 5 |
| Scenario name | Add User |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “User”, click “add user”, so the system will display to the index page of user, the admin will follow the steps to add new user, the system will redirect to the same page with success message: “User created successfully”.  Top of Form  If the admin leaves any field empty and clicks “add” the system will show error by changing the field color.    If the admin writes an existing email the system will show error message that “user already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 6 |
| Scenario name | Edit User |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, the system will display the index page of user with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this user, new email, new Phone number, the system will redirect the same page with the edit of the user.    If the admin leaves the fields of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing user e-mail the system will show error message that “User already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 7 |
| Scenario name | delete User |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, so the system will display the index page of user and show all users, the admin has to select the user that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted user. |

|  |  |
| --- | --- |
| Scenario Id | 8 |
| Scenario name | Show User |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, so the system will display the index page of user and show all users. |

|  |  |
| --- | --- |
| Scenario Id | 9 |
| Scenario name | Add User type |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “User”, click “Add User Type”, so the system will display to the index page of user type, the admin will enter new User type and click “add”, the system will redirect to the same page with success message: “User type created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing datatype the system will show error message that “User type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 10 |
| Scenario name | Edit User Type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show User Type”, the system will display the index page of user type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this user type, the system will redirect the same page with the edit of the user type name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing user type the system will show error message that “user type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 11 |
| Scenario name | Delete User Type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show user type”, so the system will display the index page of user and show all user types, the admin has to select the user type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted user type. |

|  |  |
| --- | --- |
| Scenario Id | 12 |
| Scenario name | Show User Type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show user Type”, so the system will display the index page of user type and show all user type. |

|  |  |
| --- | --- |
| Scenario Id | 13 |
| Scenario name | Add Item |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “Item”, click “Add Item”, so the system will display to the index page of user type, the admin will insert the email or any part of it of the owner, after that the admin will click “search” button, the system will redirect the emails that contain these words will be displayed in list, the admin has to select one of these emails then he must fill all the fields in the wizard form , after the admin clicks “add” a success message appears “Item created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing datatype the system will show error message that “item already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 14 |
| Scenario name | Edit Item |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “detail of items”, the system will display the index page of choosing item, the admin will choose the item the admin wants to edit then click “see item”, the system will redirect to the page of item details.    If the admin wants to edit owner’s information, the admin must click the edit icon that is in User, the system will display the page of search to choose the owner, the admin will insert the email or any part of it of the owner, after that the admin will click “search” button, the system will redirect the emails that contain these words will be displayed in list, the admin has to select one of these emails to be the owner of the item .  If the admin wants to edit location, the admin must click edit icon that is in , the system will display location form, the admin must insert all fields in the wizard form then the admin clicks “edit location”, the system will display the page of item details. |

|  |  |
| --- | --- |
| Scenario Id | 15 |
| Scenario name | Delete Item |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to delete it, the admin must click “see item”, the system will display the index of the item detail page, the admin will click trash icon to delete this item. |

|  |  |
| --- | --- |
| Scenario Id | 16 |
| Scenario name | Show Item |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to show it, the admin must click “see item”, the system will display item details that is chosen. |

|  |  |
| --- | --- |
| Scenario Id | 17 |
| Scenario name | Show Reservation |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to show it, the admin must click “see item”, the system will display item details that is chosen, the admin will click reservation icon, the system will display the index of detail of item . |

|  |  |
| --- | --- |
| Scenario Id | 18 |
| Scenario name | Add Operation |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “”, click “”, so the system will display to the index page of operation, the admin will enter new operation and click “add”, the system will redirect to the same page with success message: “operation created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing operation the system will show error message that “operation already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 19 |
| Scenario name | Edit Operation |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, the system will display the index page of operation with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this operation, the system will redirect the same page with the edit of the operation name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing operation, the system will show error message that “operation already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 20 |
| Scenario name | Delete Operation |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation and show all operations, the admin has to select the operation that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation. |

|  |  |
| --- | --- |
| Scenario Id | 21 |
| Scenario name | Show Operation |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of Operations and show all operations. |

|  |  |
| --- | --- |
| Scenario Id | 22 |
| Scenario name | Add country |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “location”, click “country”, so the system will display to the index page of country, the admin will enter new country and click “add”, the system will redirect to the same page with success message: “country created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing country the system will show error message that “country already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 23 |
| Scenario name | Edit Country |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “country”, the system will display the index page of country with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this country, the system will redirect the same page with the edit of the country name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing country the system will show error message that “country already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 24 |
| Scenario name | delete Country |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Country”, so the system will display the index page of country and show all countries, the admin has to select the country that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted country. |

|  |  |
| --- | --- |
| Scenario Id | 25 |
| Scenario name | Show country |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “country”, so the system will display the index page of country and show all countries. |

|  |  |
| --- | --- |
| Scenario Id | 26 |
| Scenario name | Add state |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “location”, click “state”, so the system will display to the index page of state, the admin will choose the country before he inserts the state name and click “add”, the system will redirect to the same page with success message: “state created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing state the system will show error message that “state already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 27 |
| Scenario name | Edit state |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “state”, the system will display the index page of state with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this state, the system will redirect the same page with the edit of the state name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing state the system will show error message that “state already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 28 |
| Scenario name | delete state |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “State”, so the system will display the index page of State and show all states, the admin has to select the state that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted state. |

|  |  |
| --- | --- |
| Scenario Id | 29 |
| Scenario name | Show State |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “state”, so the system will display the index page of state and show all states. |

|  |  |
| --- | --- |
| Scenario Id | 30 |
| Scenario name | Add City |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “location”, click “City”, so the system will display to the index page of City, the admin will choose the country after that he will choose the state before he inserts the city name and click “add”, the system will redirect to the same page with success message: “city created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing city the system will show error message that “city already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 31 |
| Scenario name | Edit city |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “city”, the system will display the index page of city with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this city, the system will redirect the same page with the edit of the city name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing city the system will show error message that “city already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 32 |
| Scenario name | delete city |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “City”, so the system will display the index page of city and show all cities, the admin has to select the state that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted city. |

|  |  |
| --- | --- |
| Scenario Id | 33 |
| Scenario name | Show City |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “City”, so the system will display the index page of state and show all cities. |

|  |  |
| --- | --- |
| Scenario Id | 34 |
| Scenario name | Add region |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “location”, click “region”, so the system will display to the index page of region, the admin will choose the country after that he will choose the state, choose city before he inserts the region name and click “add”, the system will redirect to the same page with success message: “region created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing region the system will show error message that “region already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 31 |
| Scenario name | Edit Region |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “region”, the system will display the index page of region with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this region, the system will redirect the same page with the edit of the region name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing region the system will show error message that “region already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 32 |
| Scenario name | delete Region |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Region”, so the system will display the index page of region and show all regions, the admin has to select the region that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted region. |

|  |  |
| --- | --- |
| Scenario Id | 33 |
| Scenario name | Show Region |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “Region”, so the system will display the index page of state and show all regions. |

|  |  |
| --- | --- |
| Scenario Id | 34 |
| Scenario name | Add street |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “location”, click “street”, so the system will display to the index page of street, the admin will choose the country after that he will choose the state, choose city, choose region before he inserts the street name and click “add”, the system will redirect to the same page with success message: “street created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing street the system will show error message that “street already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 35 |
| Scenario name | Edit Street |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “Street”, the system will display the index page of street with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this street, the system will redirect the same page with the edit of the street name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing street the system will show error message that “street already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 36 |
| Scenario name | delete street |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Street”, so the system will display the index page of street and show all streets, the admin has to select the street that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted street. |

|  |  |
| --- | --- |
| Scenario Id | 37 |
| Scenario name | Show street |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “location” then will click “street”, so the system will display the index page of state and show all streets. |

|  |  |
| --- | --- |
| Scenario Id | 38 |
| Scenario name | Add Operation type name |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “”, click “”, so the system will display to the index page of operation type name, the admin will choose operation type before the admin inserts operation type name and click “add”, the system will redirect to the same page with success message: “operation type created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing operation type the system will show error message that “operation type already exists”.  If the entered data does not match the required format, error message will occur “”. |
| Scenario Id | 39 |
| Scenario name | Edit Operation type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, the system will display the index page of operation type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this operation type, the system will redirect the same page with the edit of the operation type name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing operation type, the system will show error message that “operation type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 40 |
| Scenario name | Delete Operation type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation type and show all operation types, the admin has to select the operation type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation type. |

|  |  |
| --- | --- |
| Scenario Id | 41 |
| Scenario name | Show Operation type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation types and show all operation types. |

|  |  |
| --- | --- |
| Scenario Id | 42 |
| Scenario name | Add Main type |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “main type”, so the system will display to the index page of main type, the admin will enter new main type and click “add”, the system will redirect to the same page with success message: “main type created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing main type the system will show error message that “Main type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 43 |
| Scenario name | Edit Main type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “main type”, the system will display the index page of main type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this main type, the system will redirect the same page with the edit of the main type name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing main type the system will show error message that “main type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 44 |
| Scenario name | delete main type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Main type”, so the system will display the index page of main type and show all main types, the admin has to select the main type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted main type. |

|  |  |
| --- | --- |
| Scenario Id | 45 |
| Scenario name | Show main type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Main type”, so the system will display the index page of main type and show all main types. |

|  |  |
| --- | --- |
| Scenario Id | 46 |
| Scenario name | Add Sub type |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “sub type”, so the system will display to the index page of sub type, the admin will choose new main type before the admin inserts sub type and click “add”, the system will redirect to the same page with success message: “sub type created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing sub type the system will show error message that “Sub type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 47 |
| Scenario name | Edit sub type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “sub type”, the system will display the index page of sub type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this sub type, the system will redirect the same page with the edit of the sub type name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing sub type the system will show error message that “Sub type already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 48 |
| Scenario name | delete sub type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type”, so the system will display the index page of sub type and show all sub types, the admin has to select the sub type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted sub type. |

|  |  |
| --- | --- |
| Scenario Id | 49 |
| Scenario name | Show sub type |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type”, so the system will display the index page of sub type and show all sub types. |

|  |  |
| --- | --- |
| Scenario Id | 50 |
| Scenario name | Add Subtype property |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “sub type property”, so the system will display to the index page of sub type property, the admin will choose new main type, chooses sub type before the admin inserts sub type property and click “add”, the system will redirect to the same page with success message: “sub type property created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing sub type property, the system will show error message that “Sub type property already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 51 |
| Scenario name | Edit sub type property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “sub type property”, the system will display the index page of sub type property with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this sub type property, the system will redirect the same page with the edit of the sub type property name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing sub type property, the system will show error message that “Sub type property already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 53 |
| Scenario name | delete sub type property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type property”, so the system will display the index page of sub type and show all sub type properties, the admin has to select the sub type property that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted sub type property. |

|  |  |
| --- | --- |
| Scenario Id | 54 |
| Scenario name | Show sub type property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type property”, so the system will display the index page of sub type property and show all sub type properties. |

|  |  |
| --- | --- |
| Scenario Id | 55 |
| Scenario name | Add details of property |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “Details of property”, so the system will display to the index page of details of property, the admin will choose new main type, chooses sub type, chooses sub type property, before the admin inserts detail of the property and click “add”, the system will redirect to the same page with success message: “Detail of property created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing details of property, the system will show error message that “Details of property already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 56 |
| Scenario name | Edit details of property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, the system will display the index page of details of property with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this details of property, the system will redirect the same page with the edit of the details of property name.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the admin writes in the field of the pop-up an existing details of property, the system will show error message that “details of property already exists”.  If the entered data does not match the required format, error message will occur “”. |

|  |  |
| --- | --- |
| Scenario Id | 57 |
| Scenario name | delete details of property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, so the system will display the index page of details of property and show all details of properties, the admin has to select the details of property that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted details of property. |

|  |  |
| --- | --- |
| Scenario Id | 58 |
| Scenario name | Show details of property |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, so the system will display the index page of details of property and show all details of properties. |

|  |  |
| --- | --- |
| Scenario Id | 59 |
| Scenario name | Add details of item |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “Details of item”, so the system will display to the index page of details of item, the admin will choose new main type, chooses sub type, chooses sub type property, chooses detail of property before the admin inserts detail of the item and click “add”, the system will redirect to the same page with success message: “Detail of item created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.    If the admin writes an existing details of item, the system will show error message that “Details of item already exists”.  If the entered data does not match the required format, error message will occur “”. |

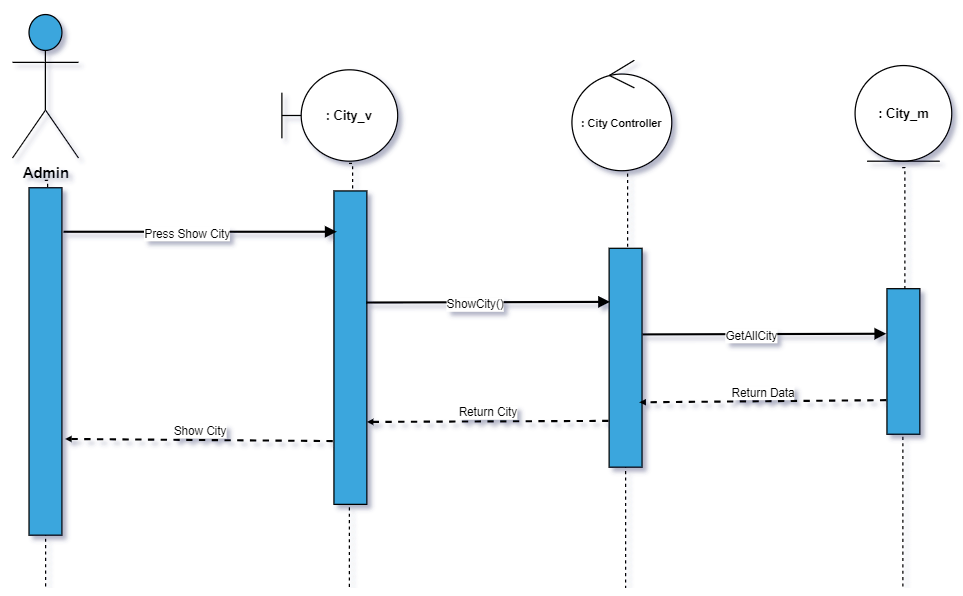
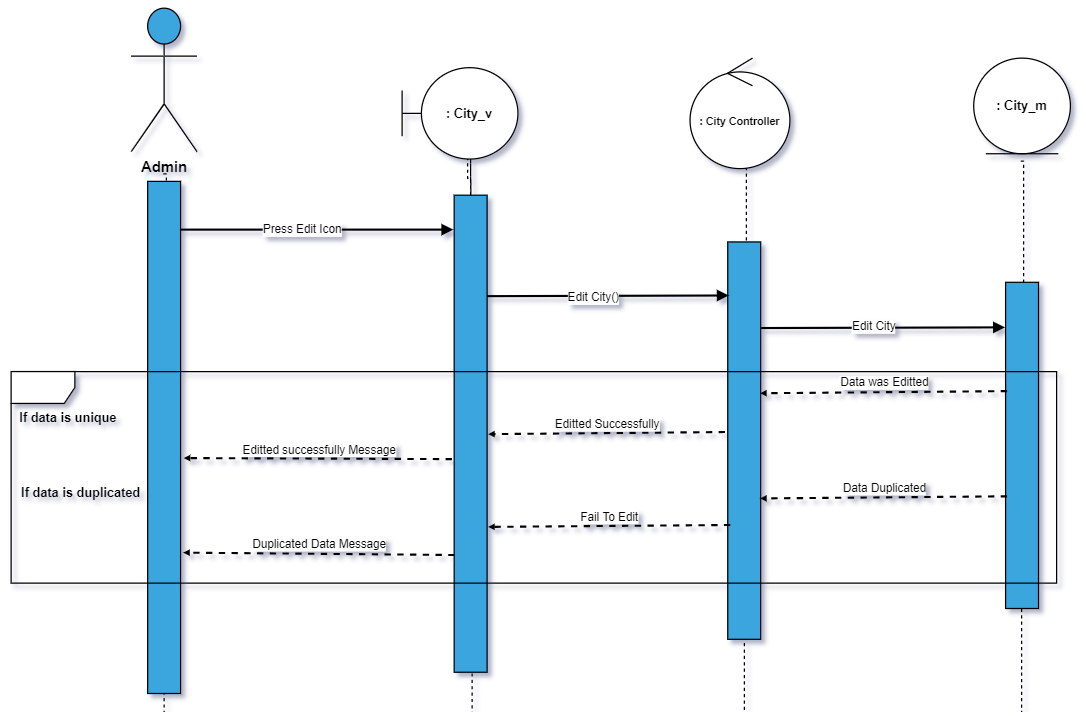
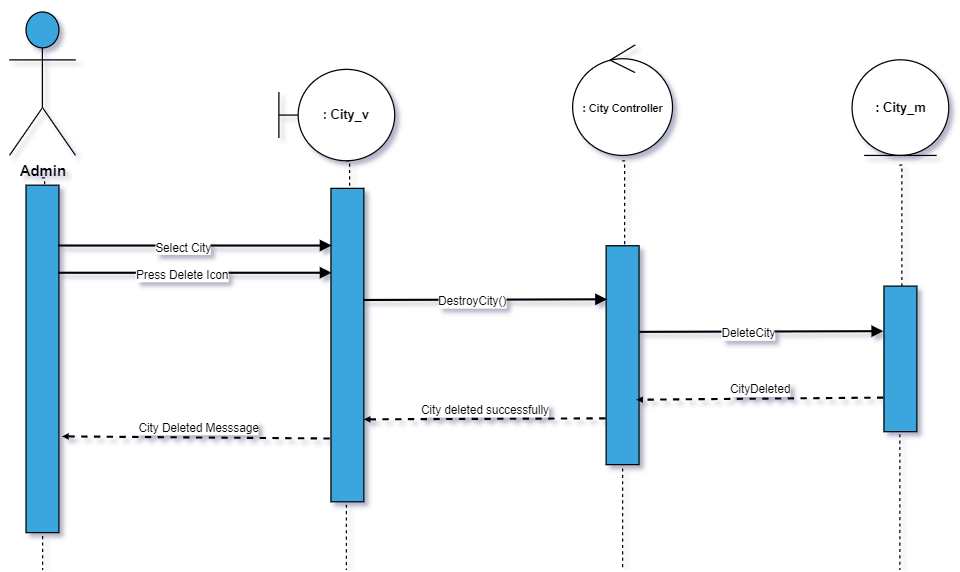
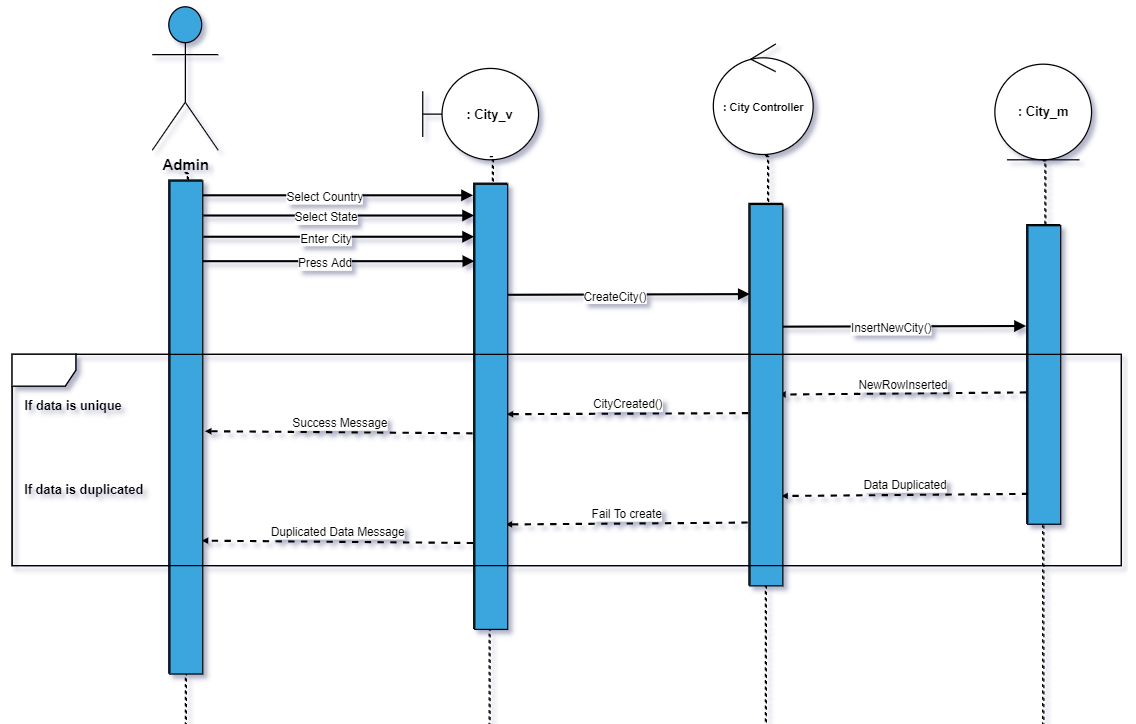
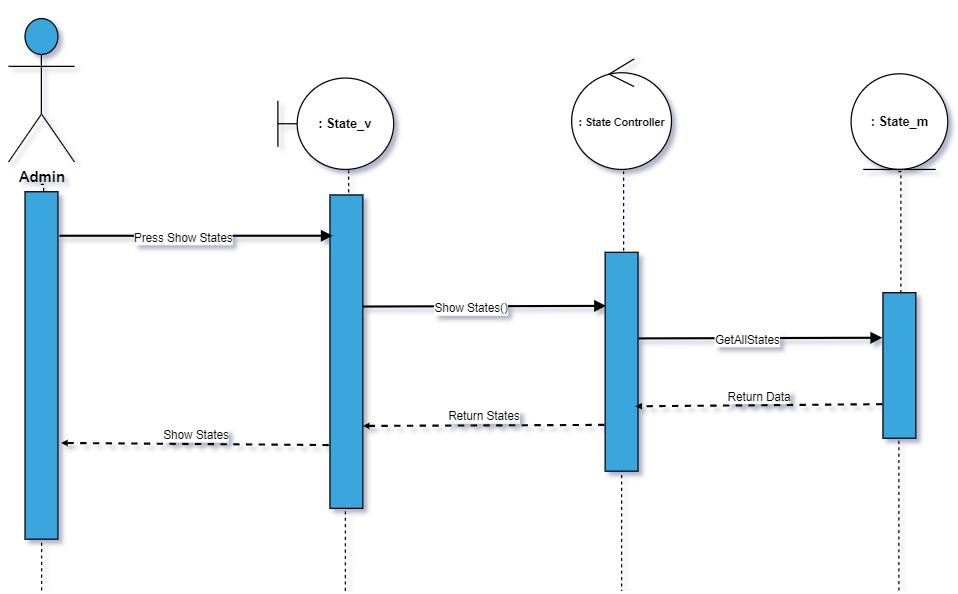
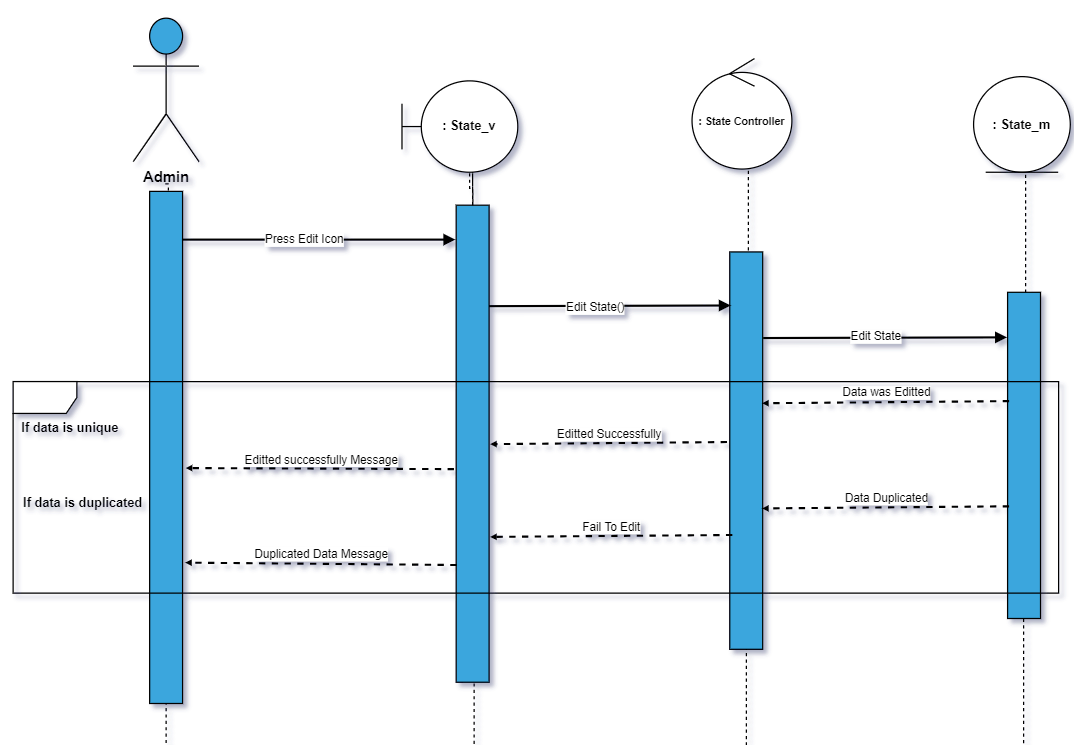
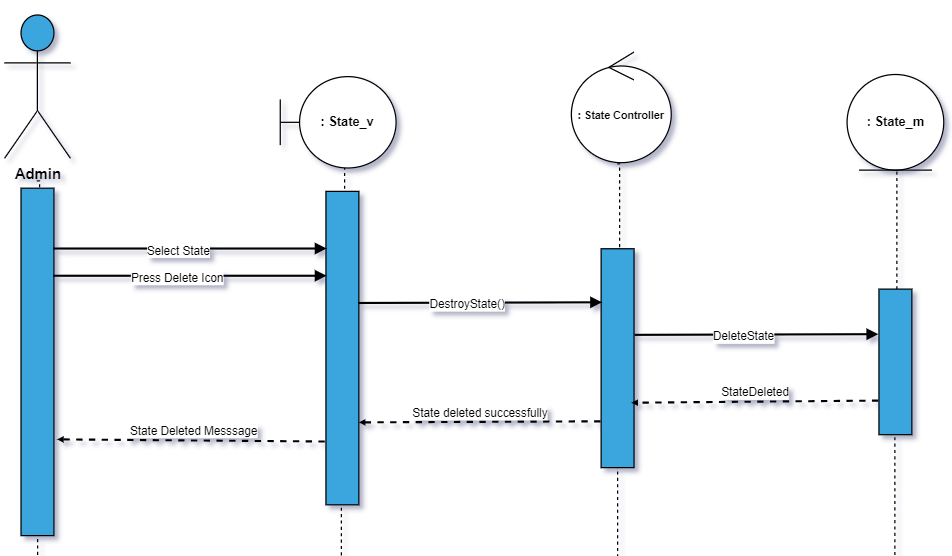
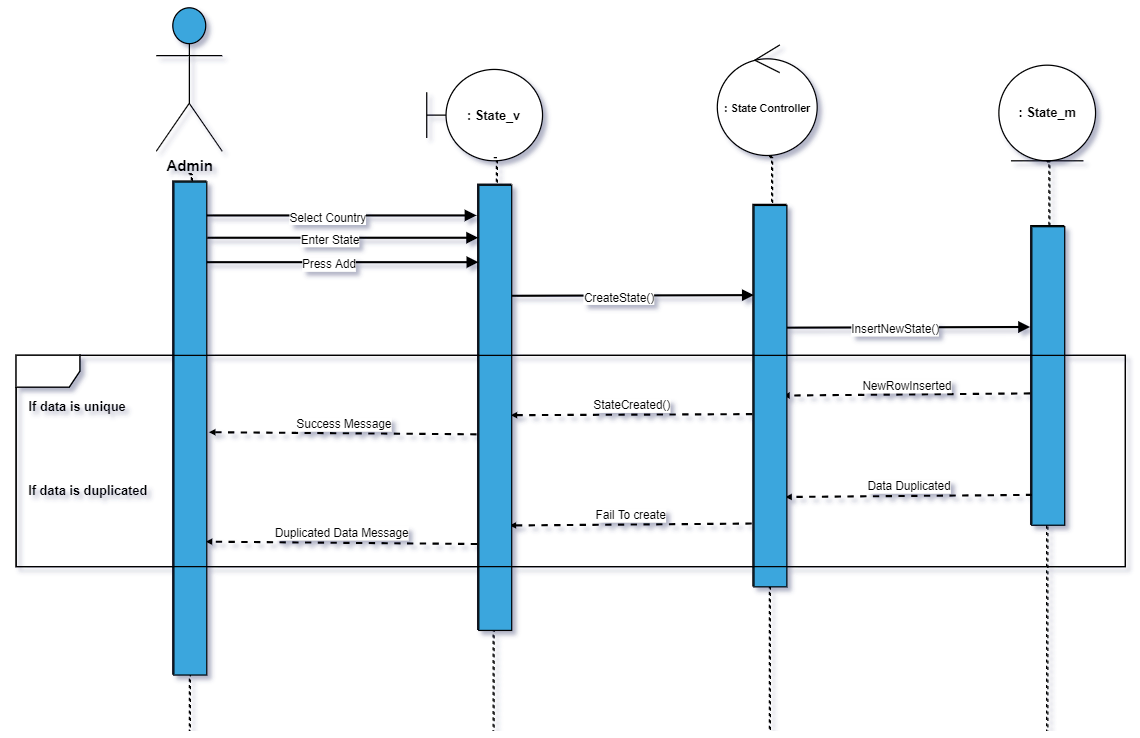
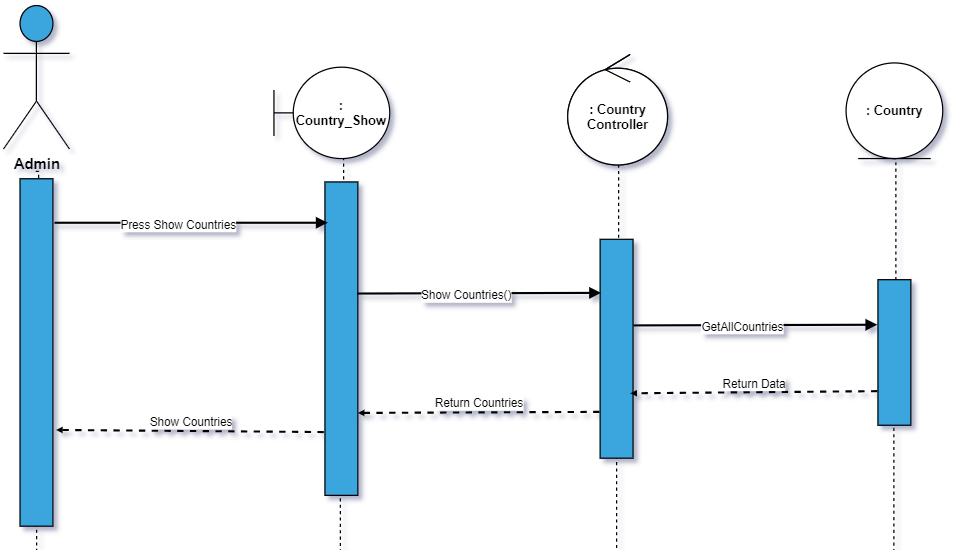
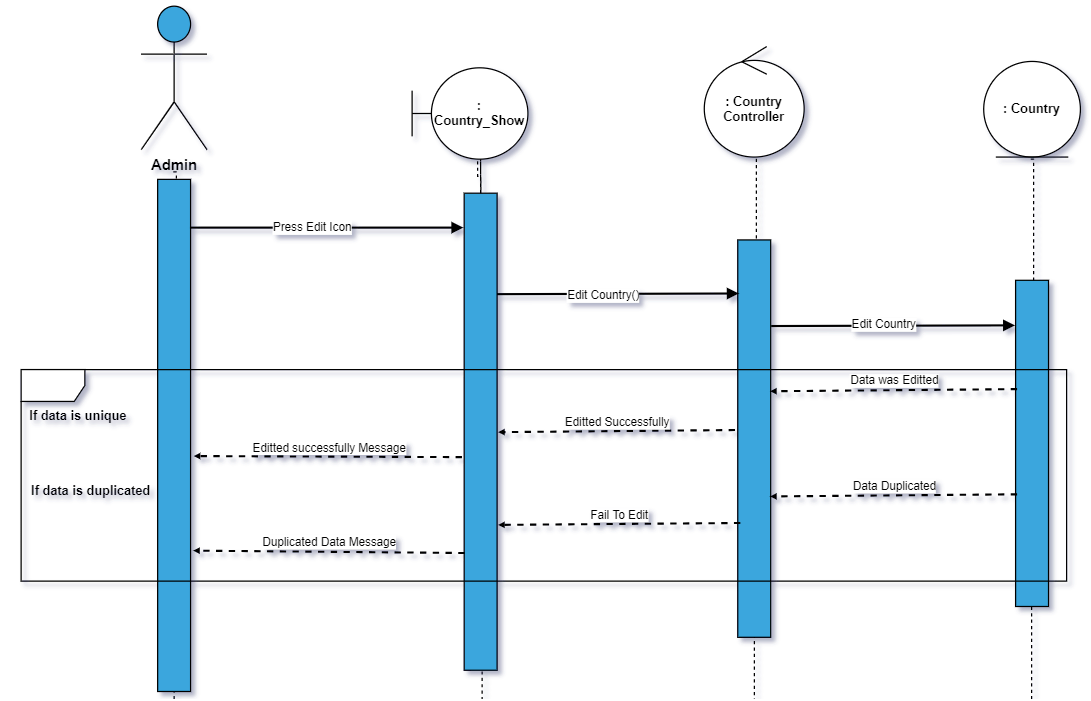
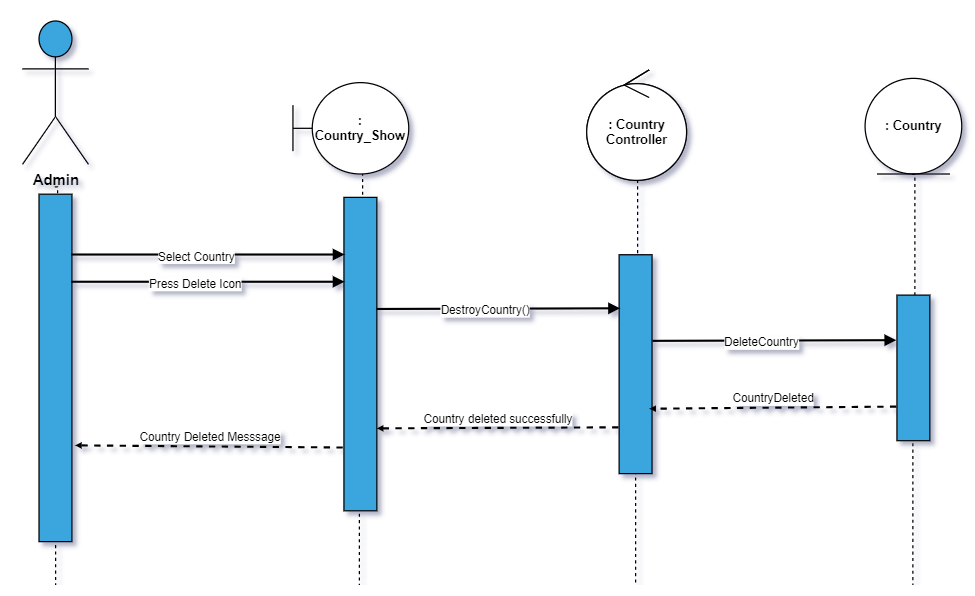
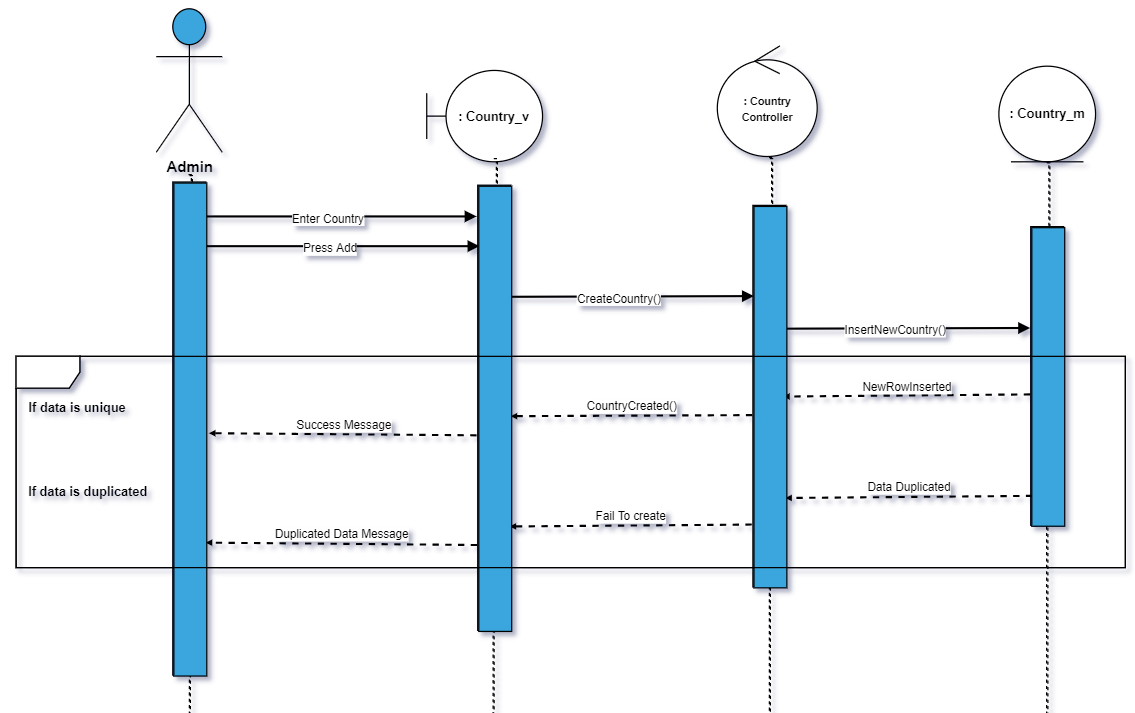
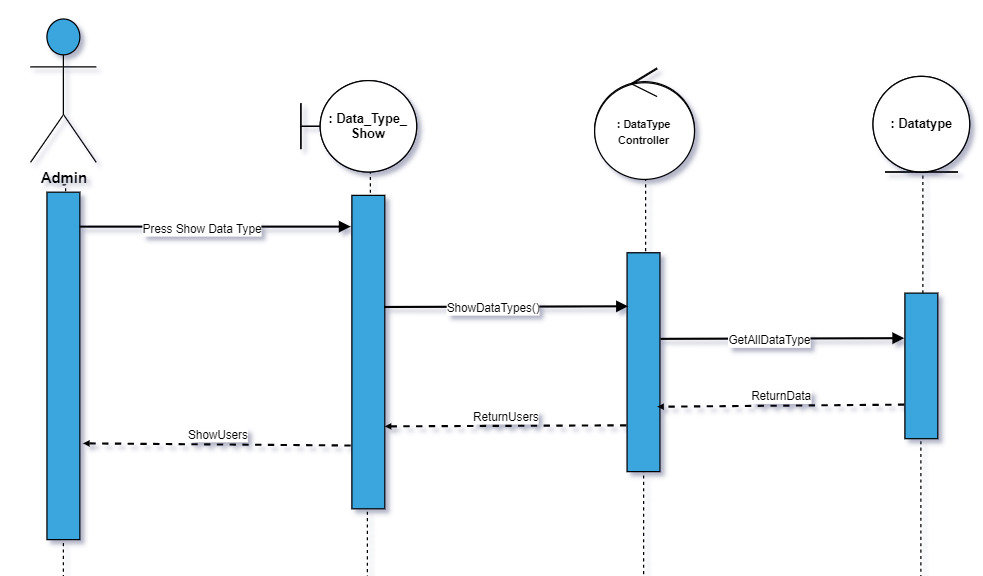
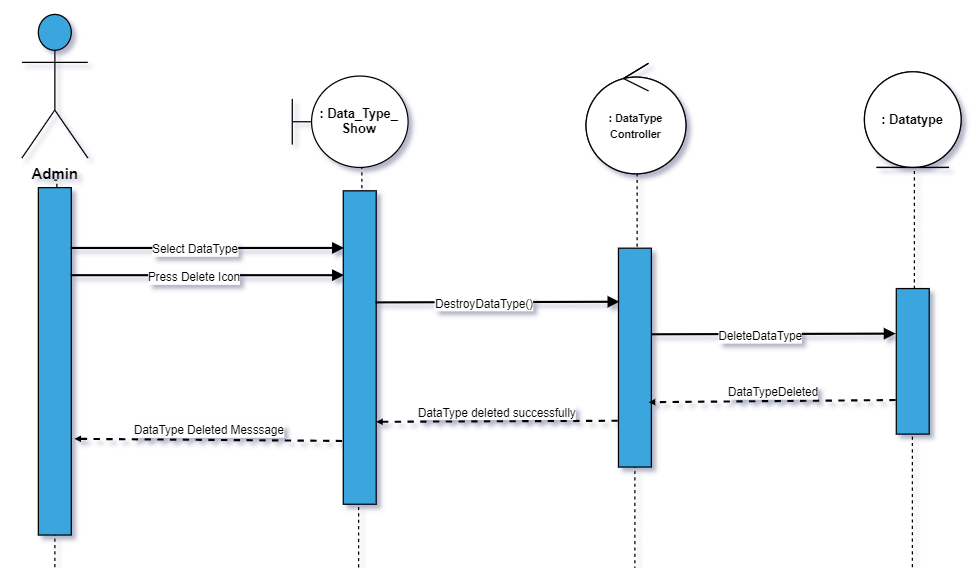
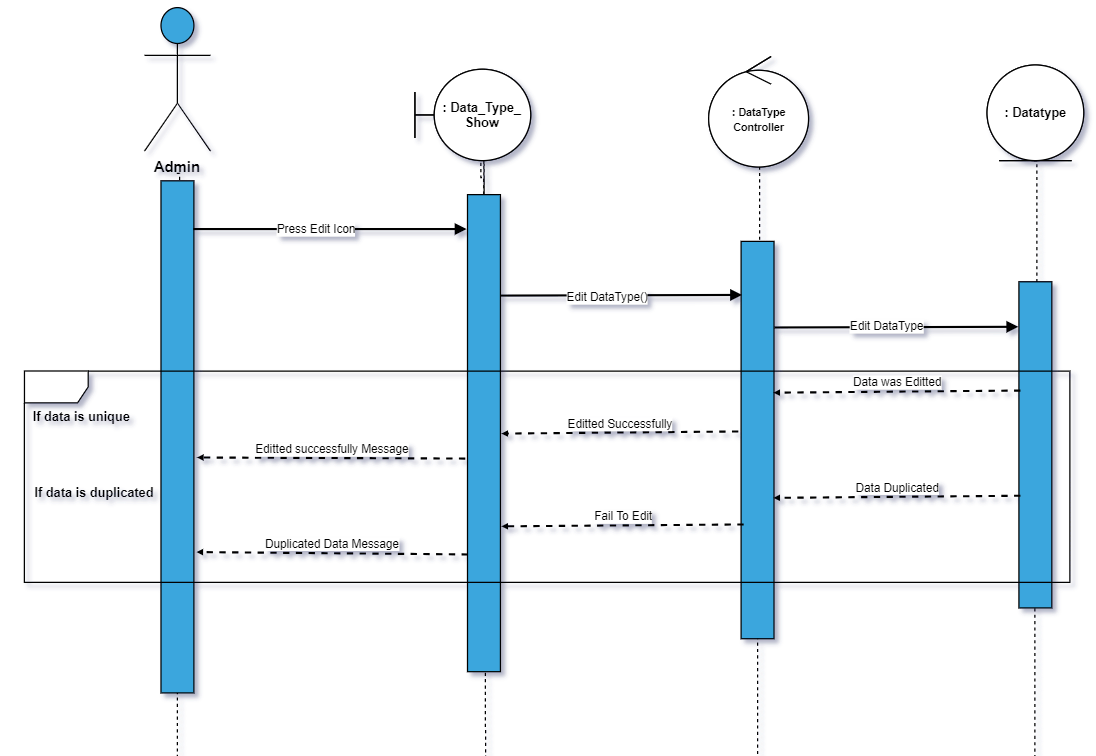
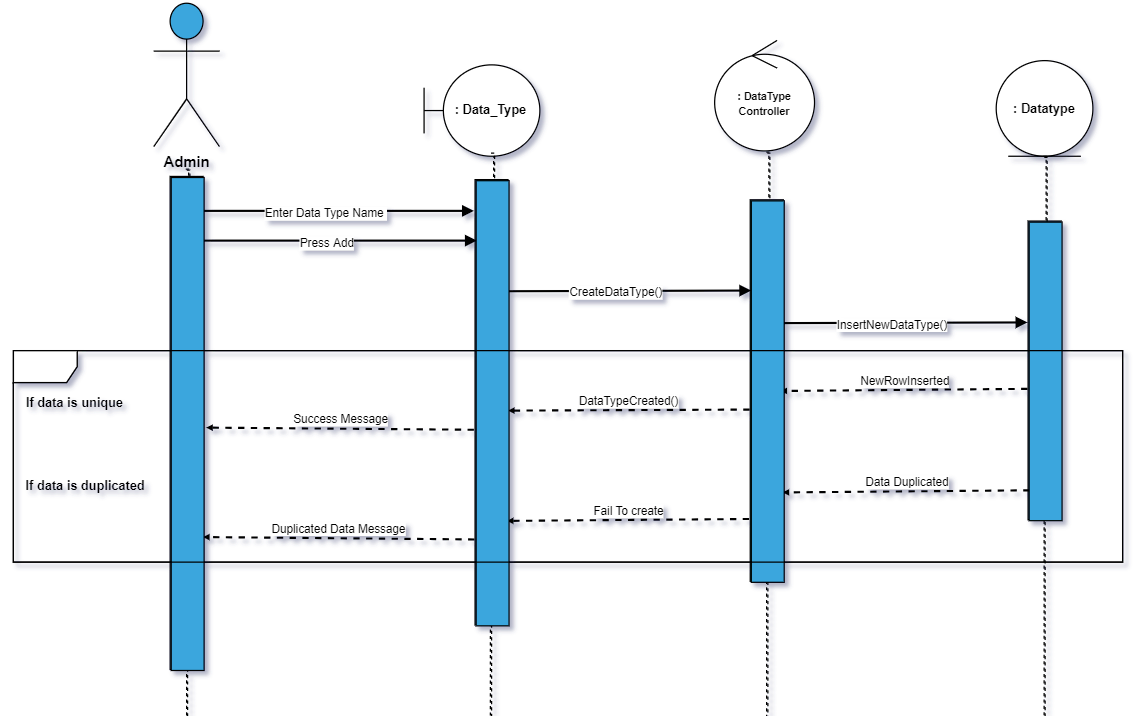
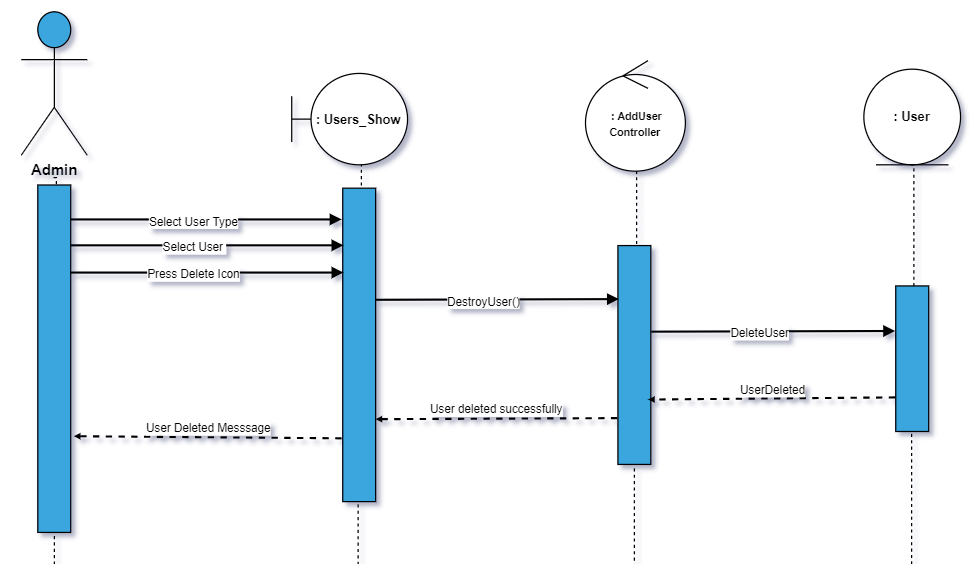
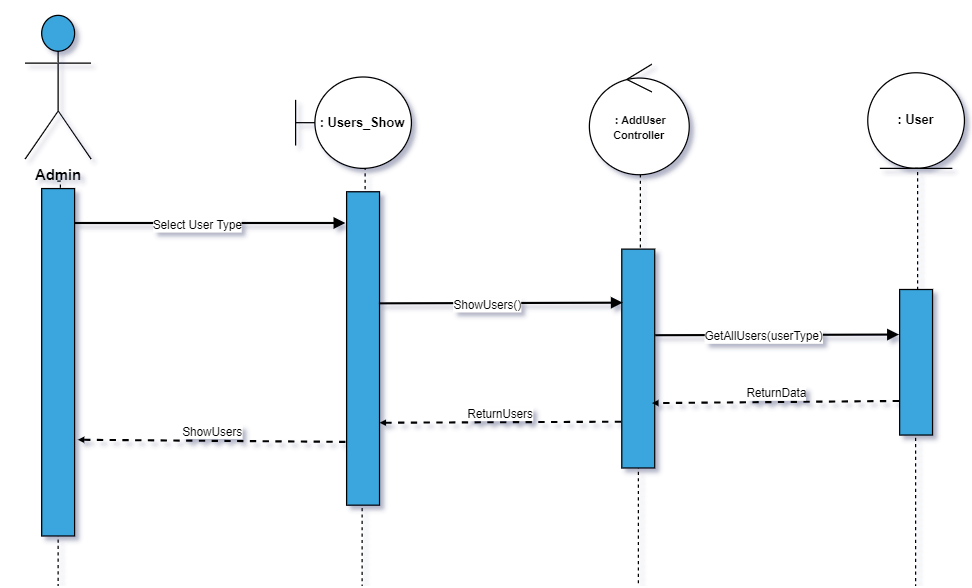
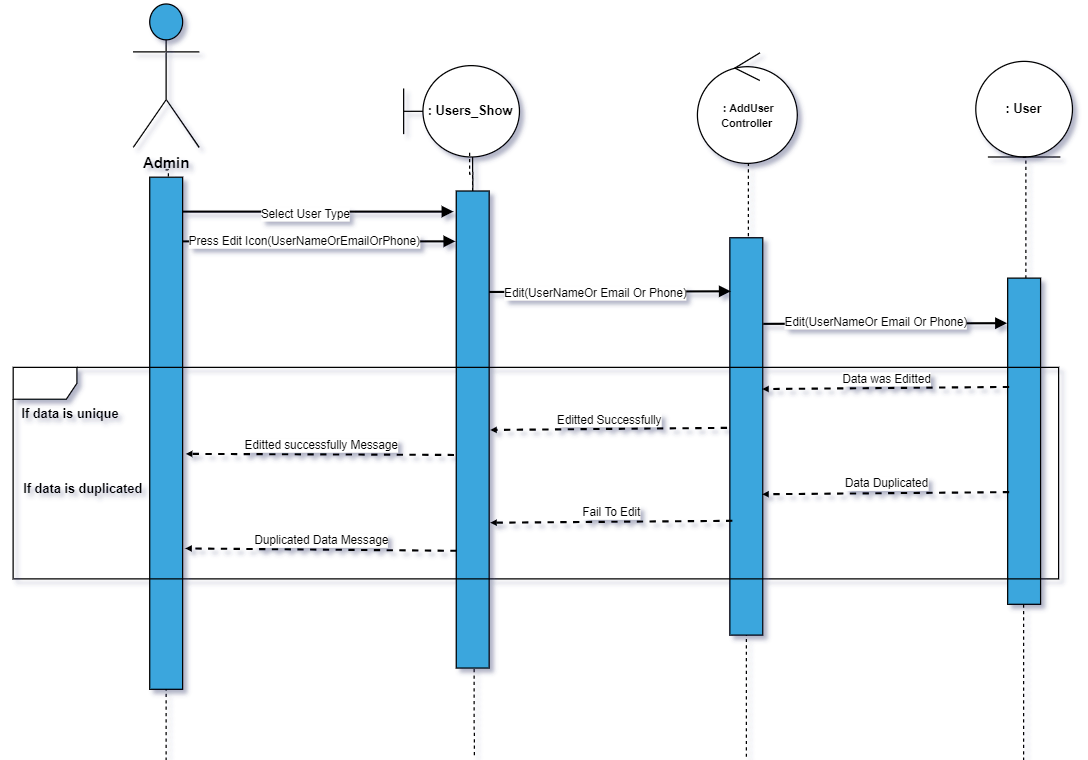
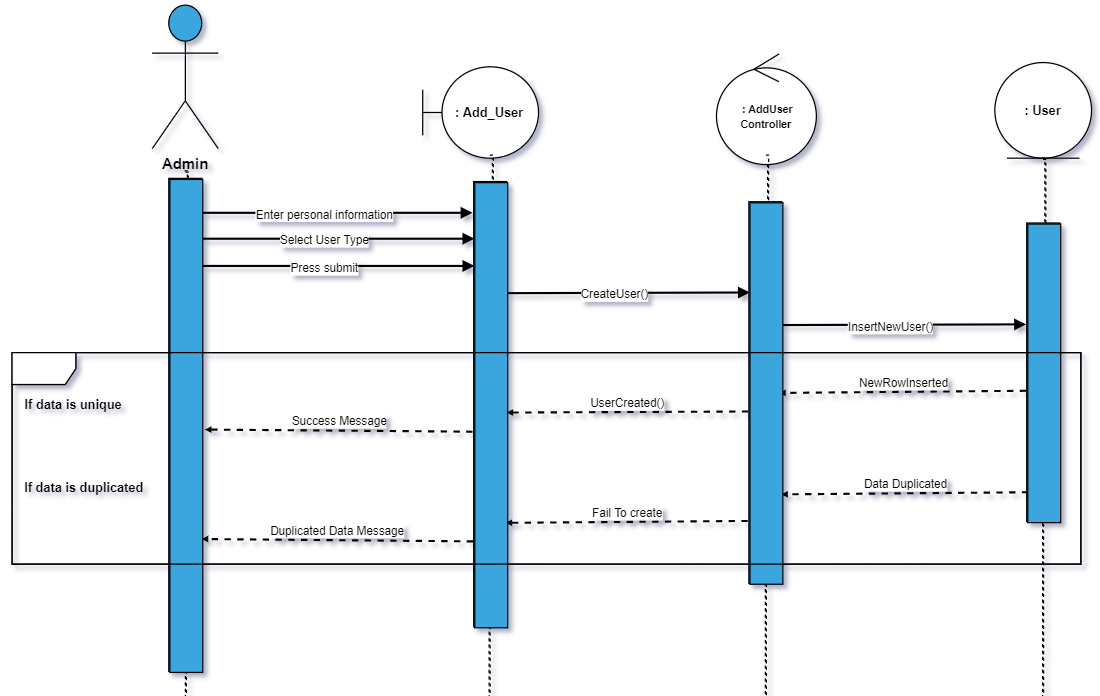
|  |  |
| --- | --- |
| Scenario Id | 60 |
| Scenario name | Add Location |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “Item”, click “Add Item”, so the system will display to the index page of add item, the admin has two ways to enter the location the first complete the wizard, secondly the admin has to click on google maps to search about the street and click in right marker, the system saves the longitude and latitude, successful message occures “location saved successfully”;  Top of Form |

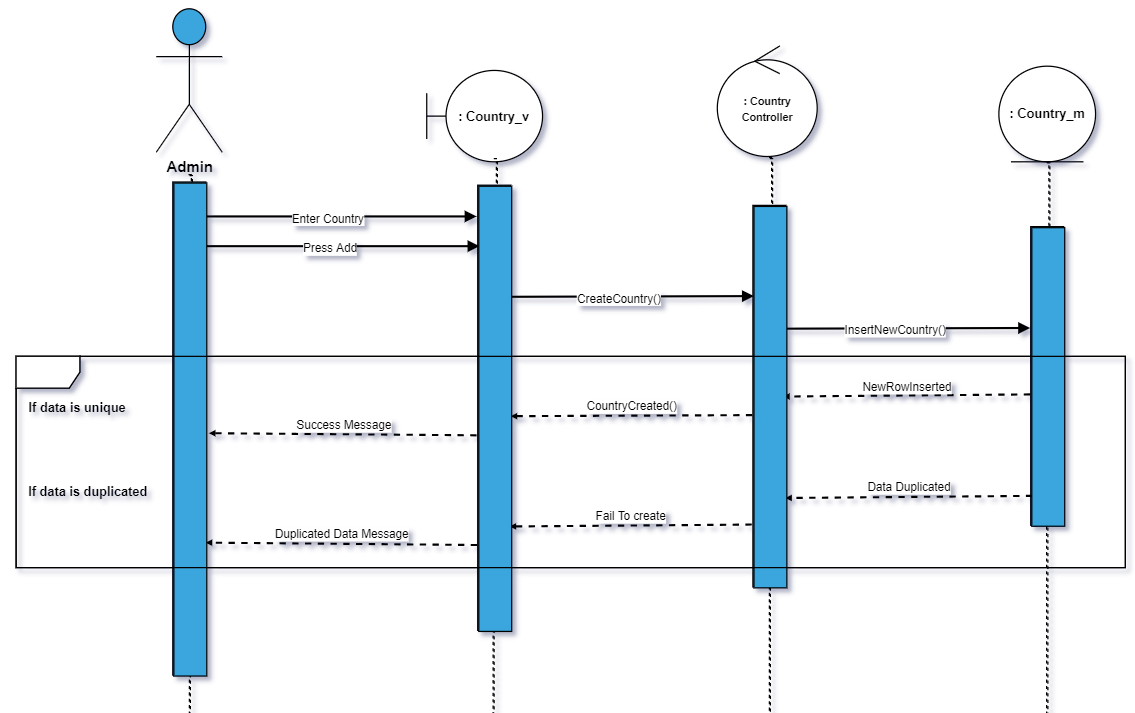
|  |  |
| --- | --- |
| Scenario Id | 61 |
| Scenario name | Add Operation type value |
| Actor | Admin |
| Normal | The Admin will log in to the system, then from the slide bar, admin will click “”, click “”, so the system will display to the index page of operation type value, the admin will choose operation type, chooses operation type name before the admin inserts operation type value and click “add”, the system will redirect to the same page with success message: “operation type value created successfully”.  Top of Form  If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.  If the entered data does not match the required format, error message will occur “”. |

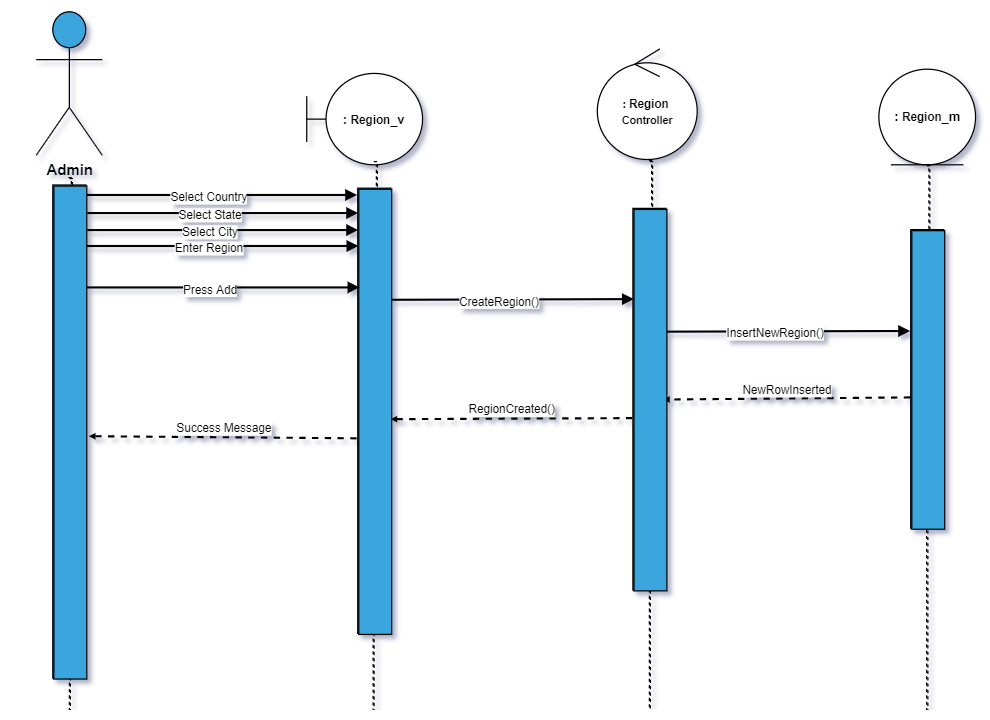
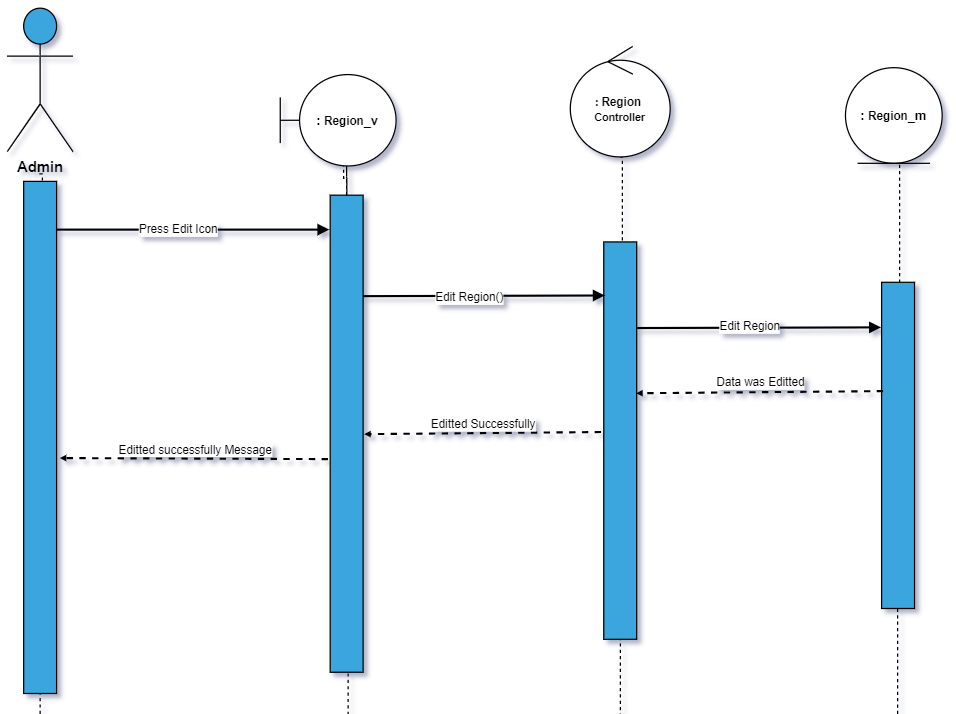
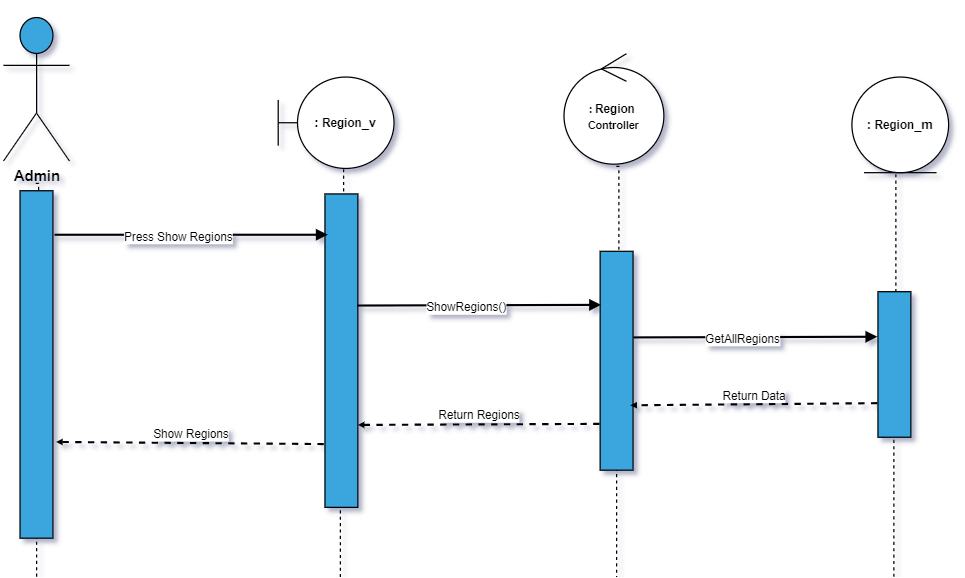
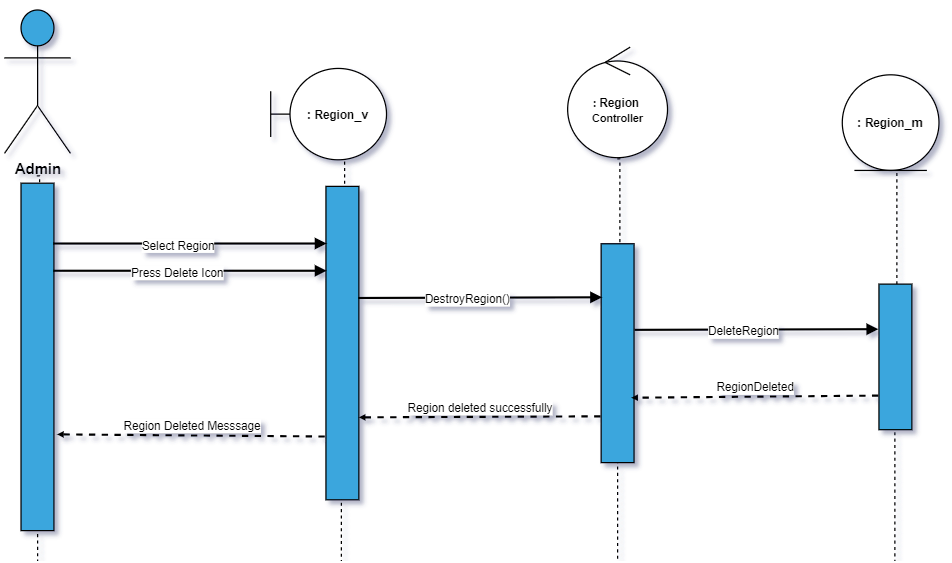
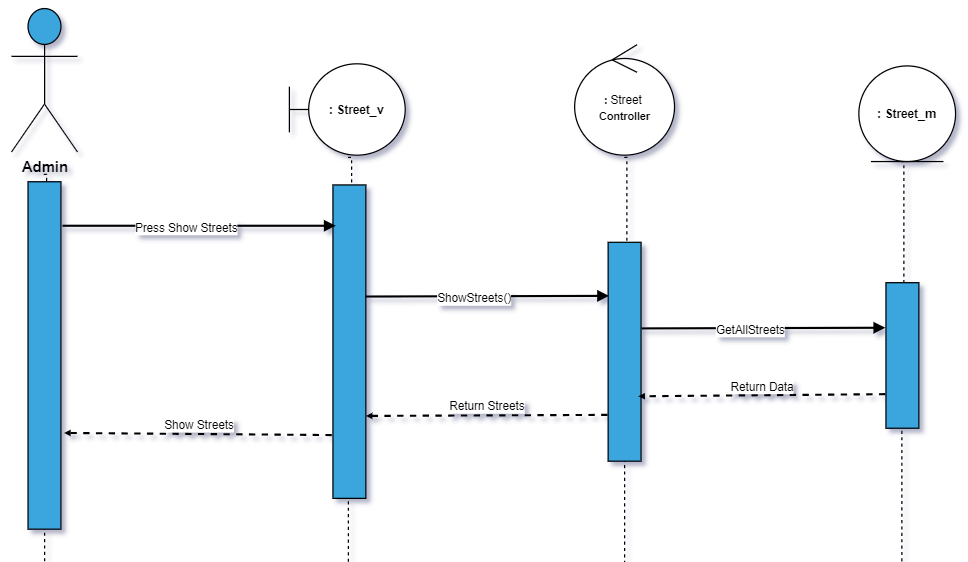
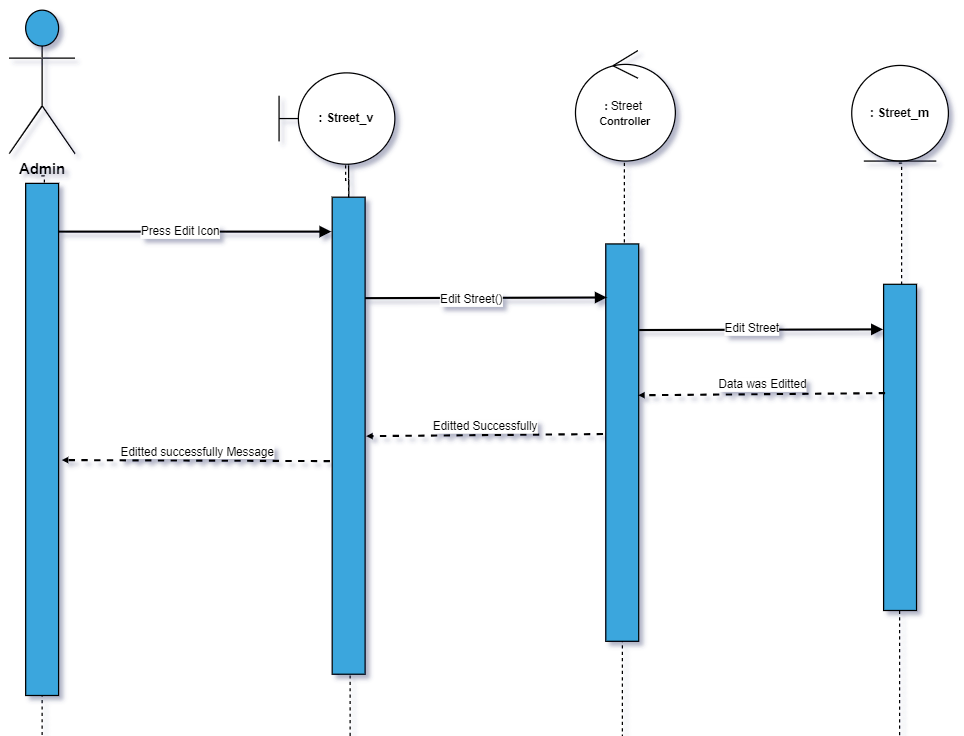
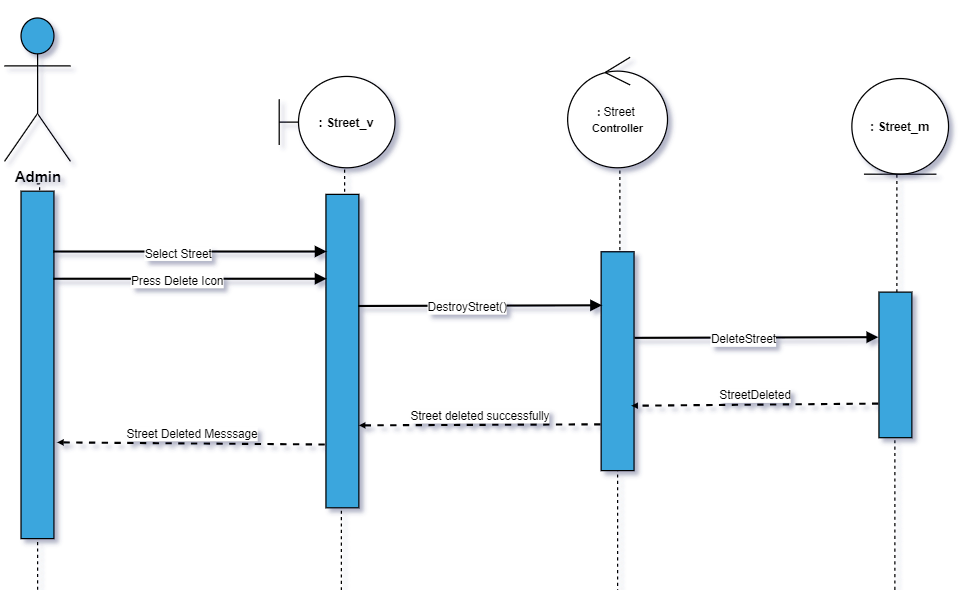
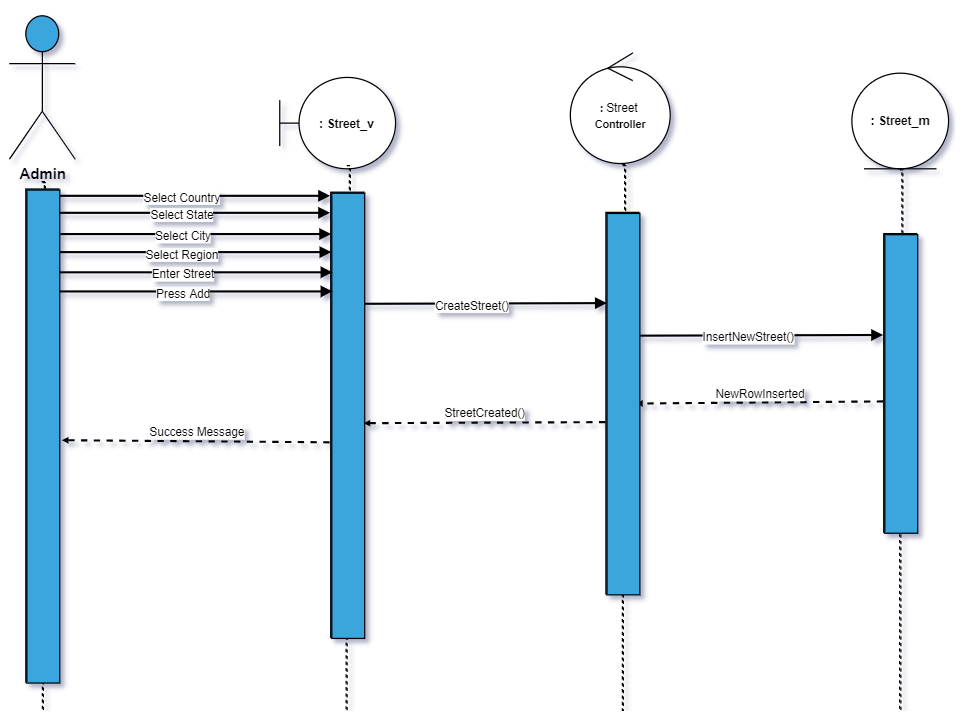
|  |  |
| --- | --- |
| Scenario Id | 62 |
| Scenario name | Edit Operation type value |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, the system will display the index page of operation type value with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new input of this operation type value, the system will redirect the same page with the edit of the operation type value.    If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.  If the entered data does not match the required format, error message will occur “”. |

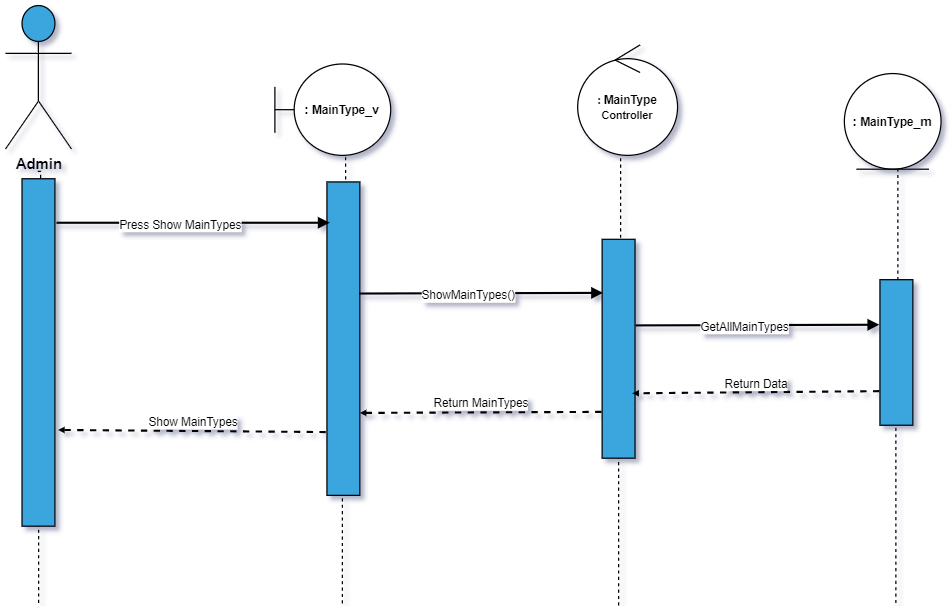
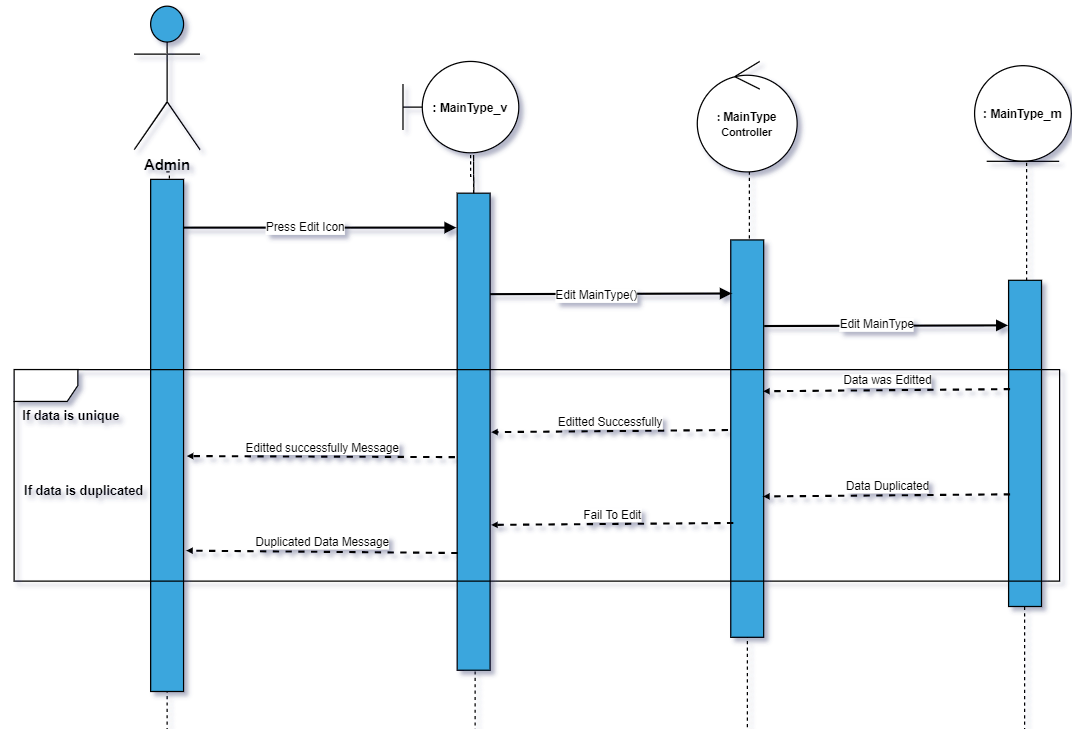
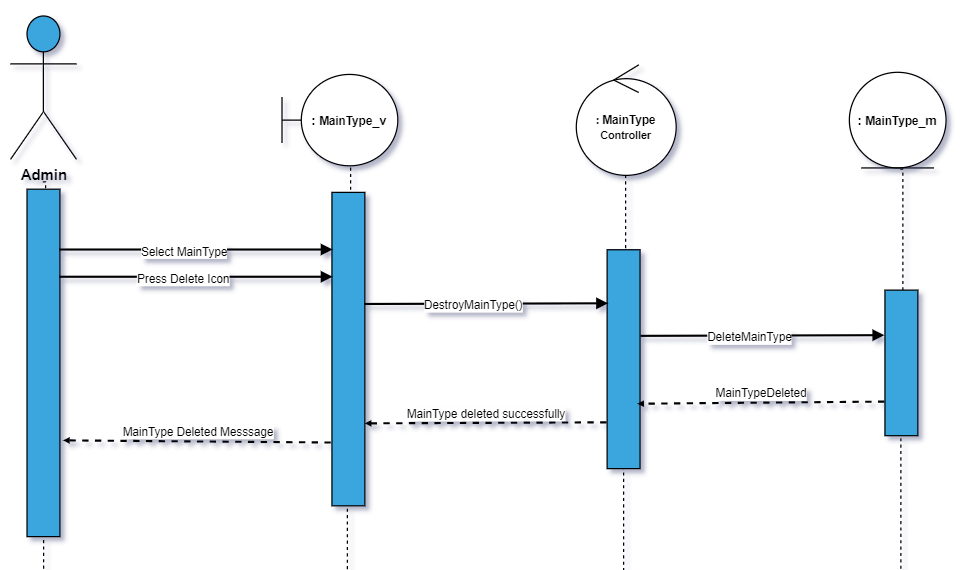
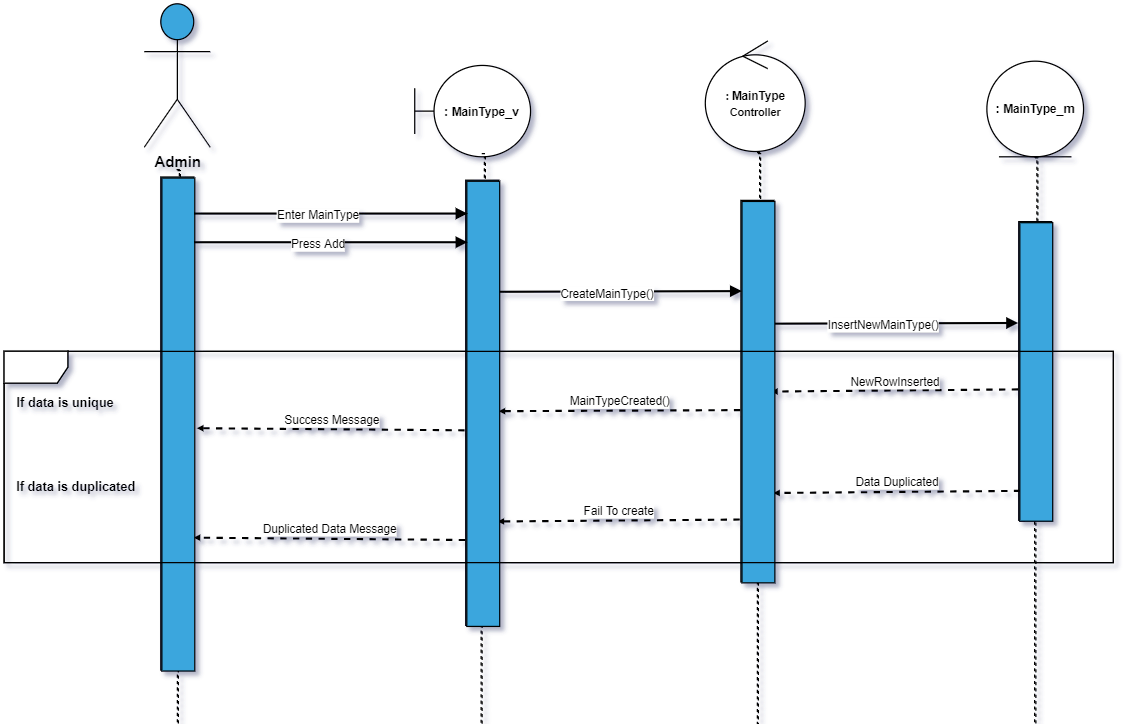
|  |  |
| --- | --- |
| Scenario Id | 63 |
| Scenario name | Delete Operation type value |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation type value and show all operation type values, the admin has to select the operation type value that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation type value. |

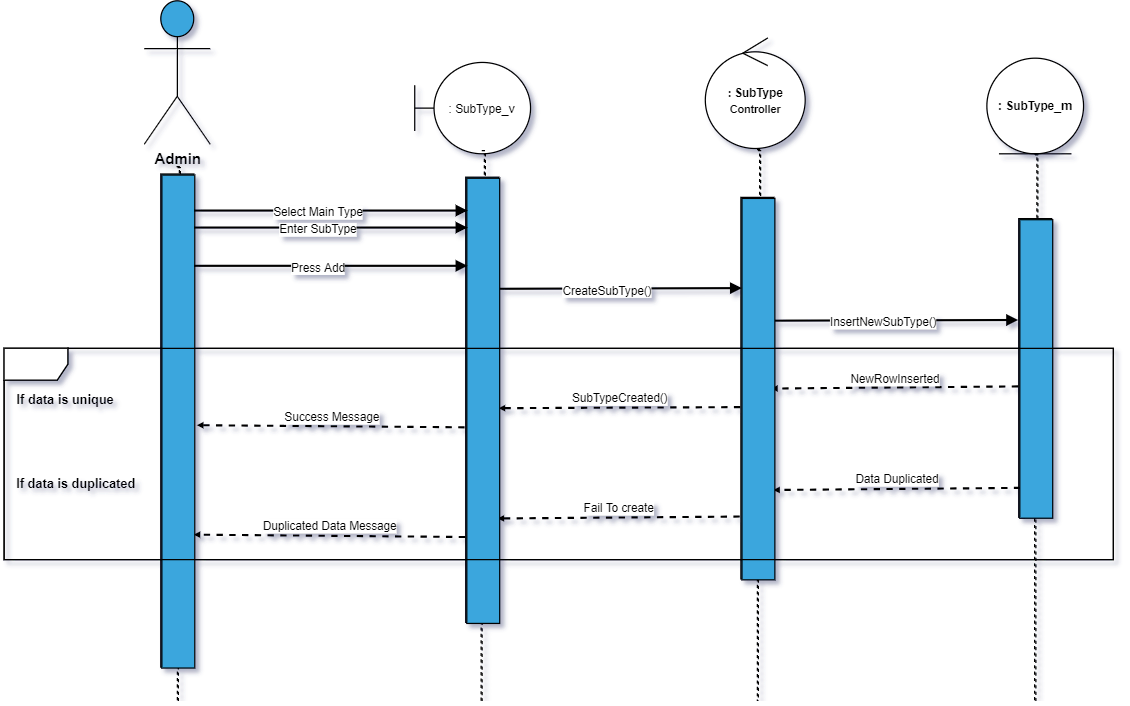
|  |  |
| --- | --- |
| Scenario Id | 64 |
| Scenario name | Show Operation type value |
| Actor | Admin |
| Normal | The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation types and show all operation type values. |

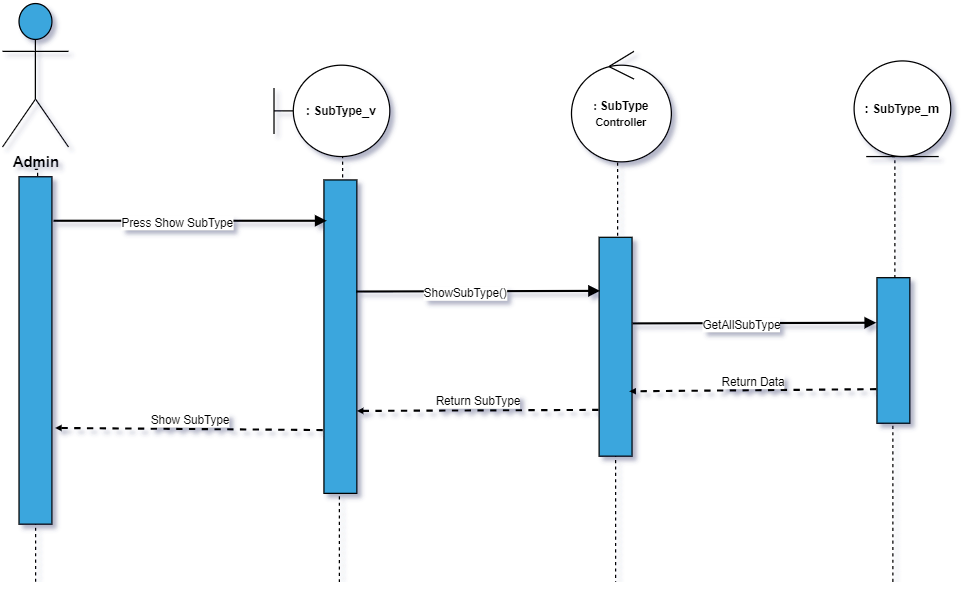
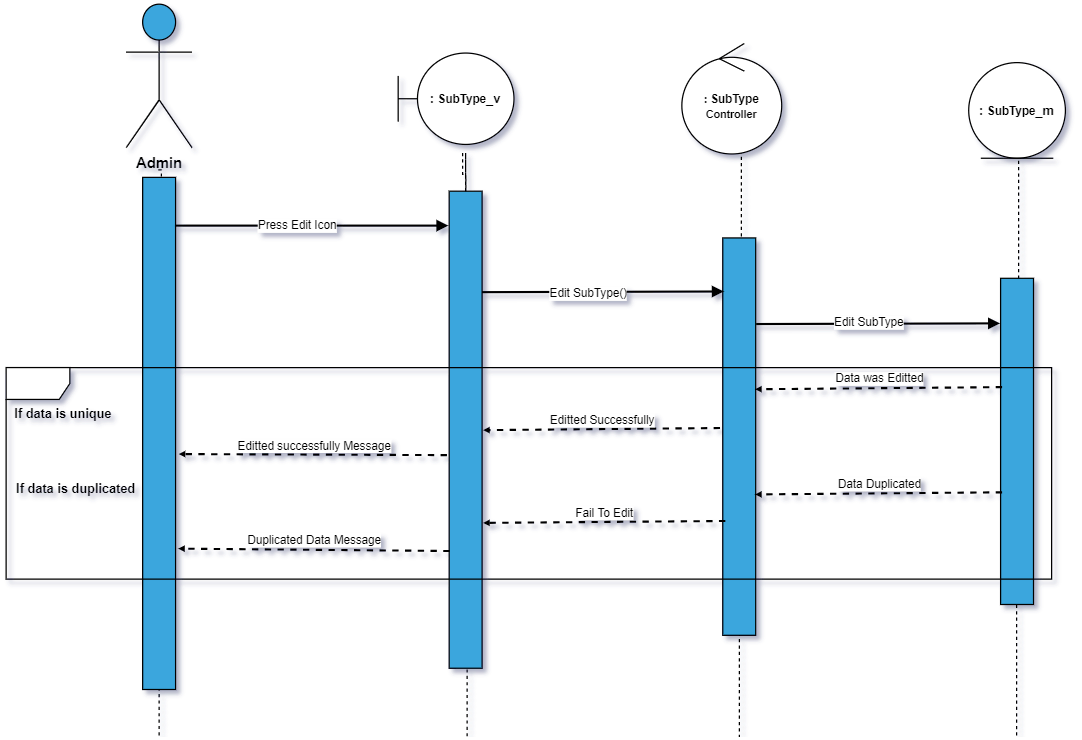
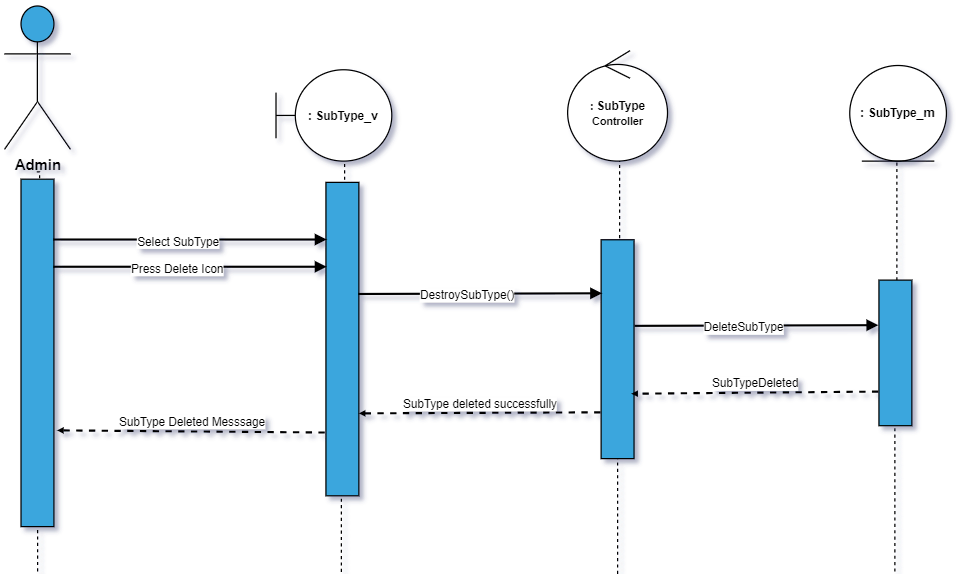


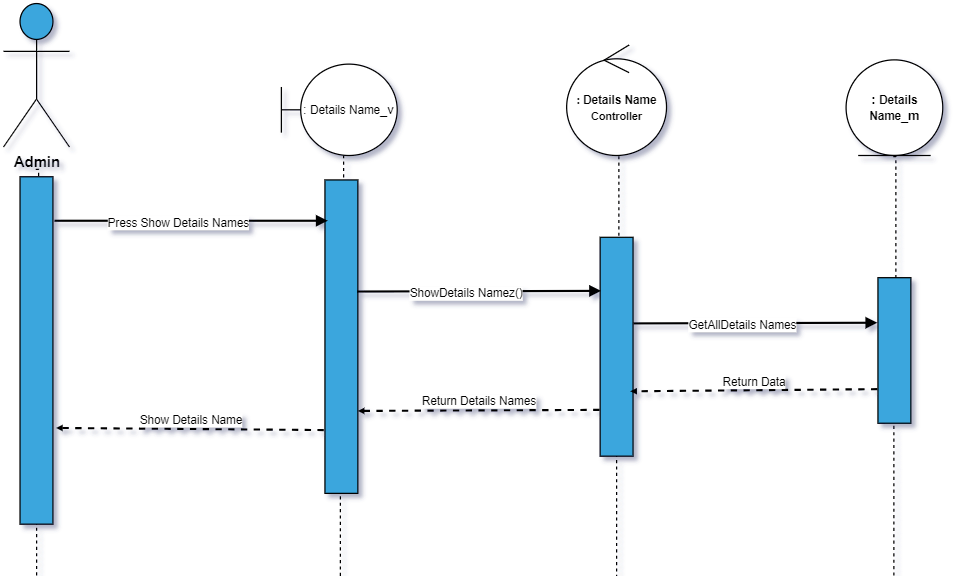
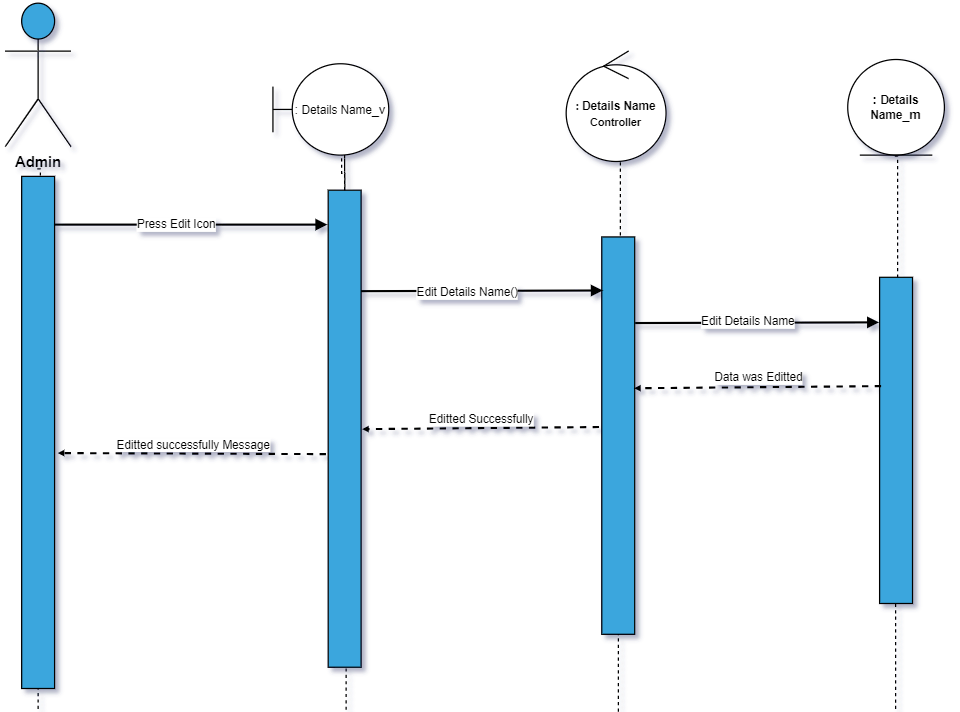
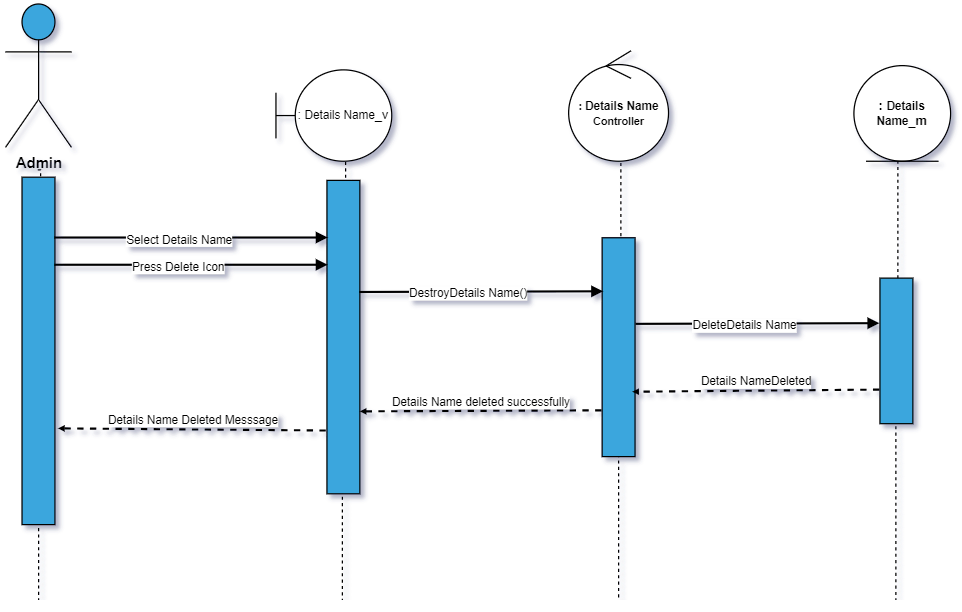
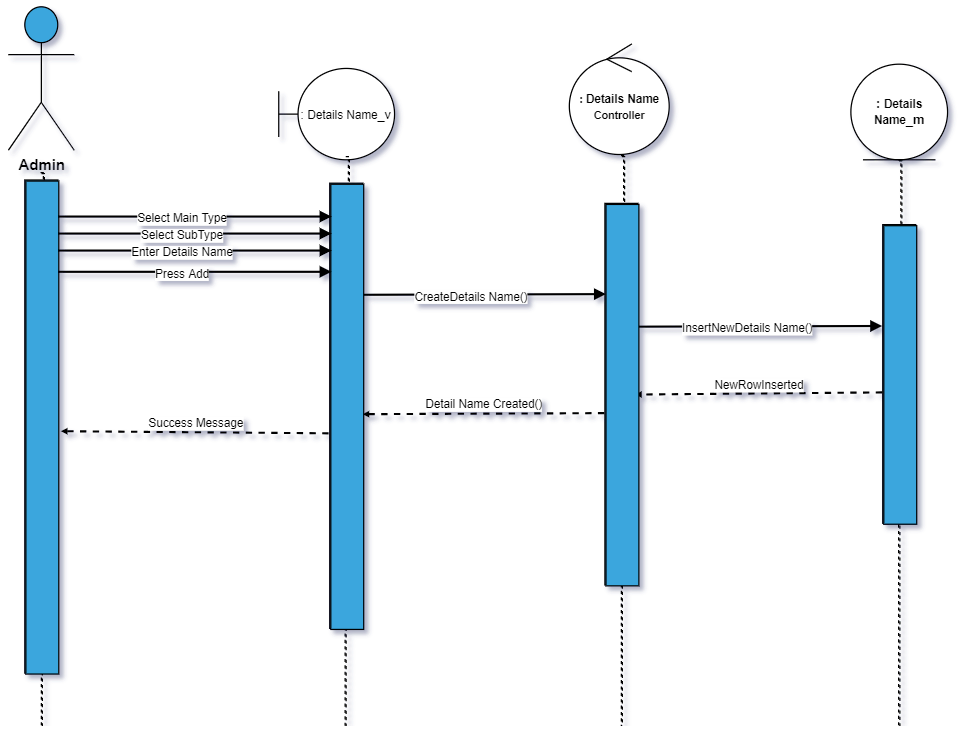


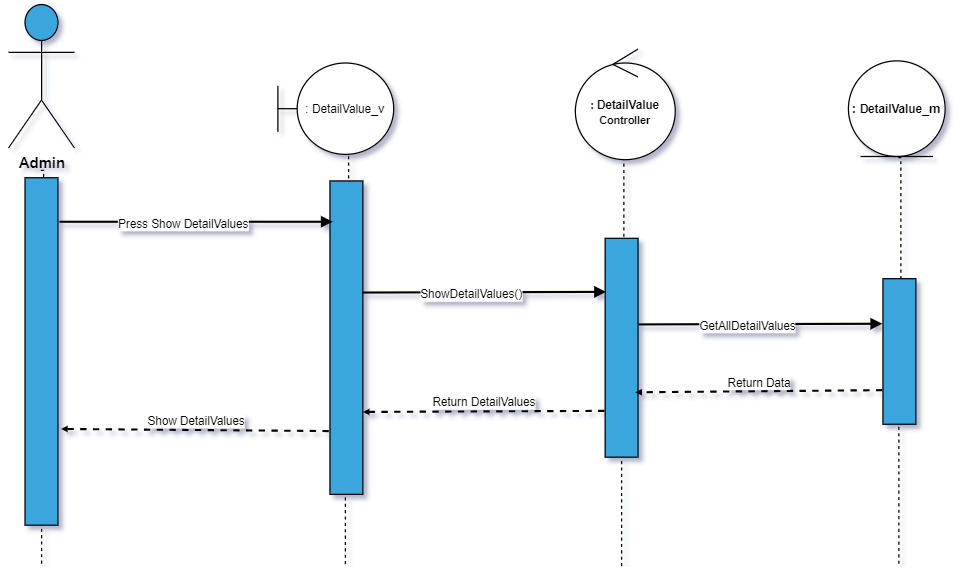
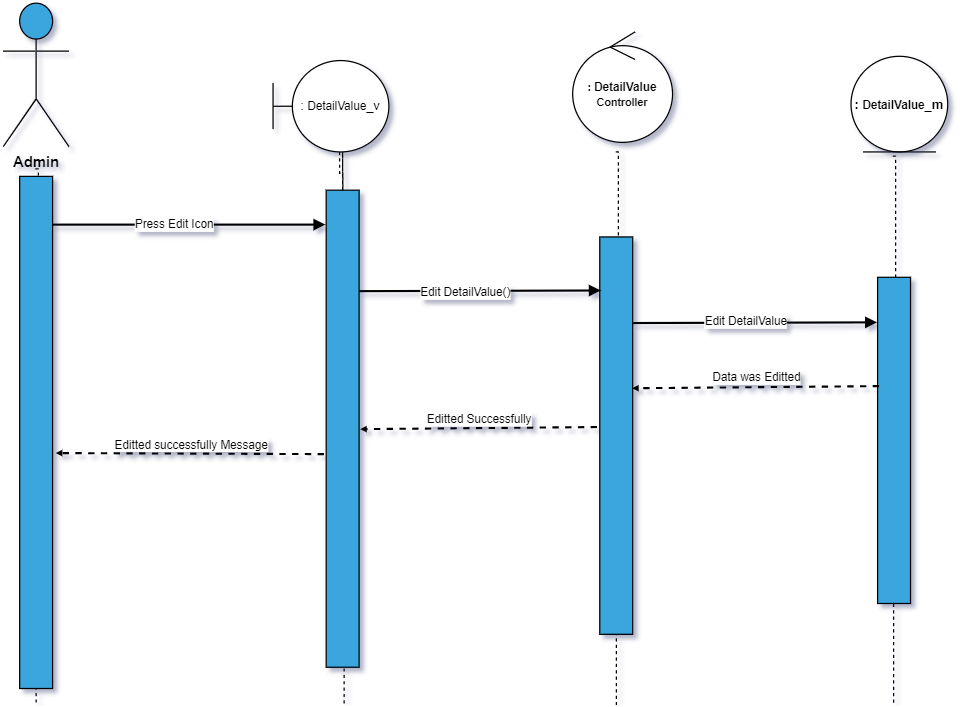
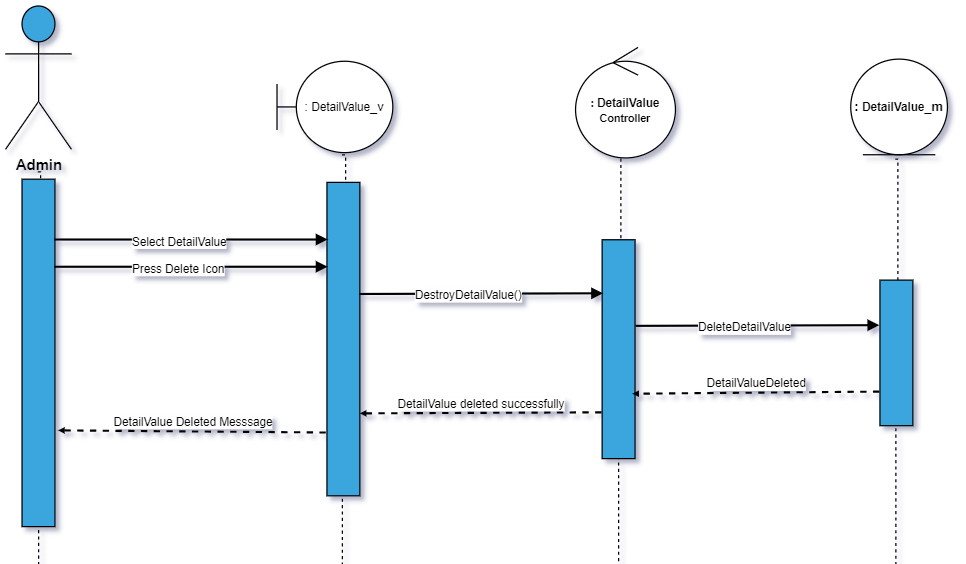
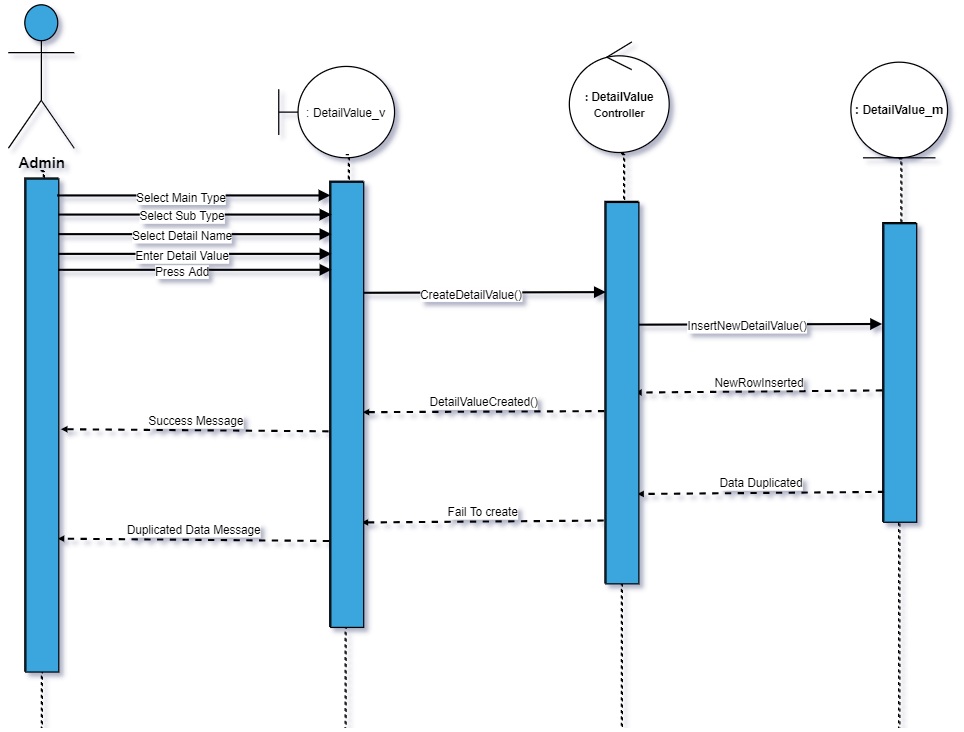
 

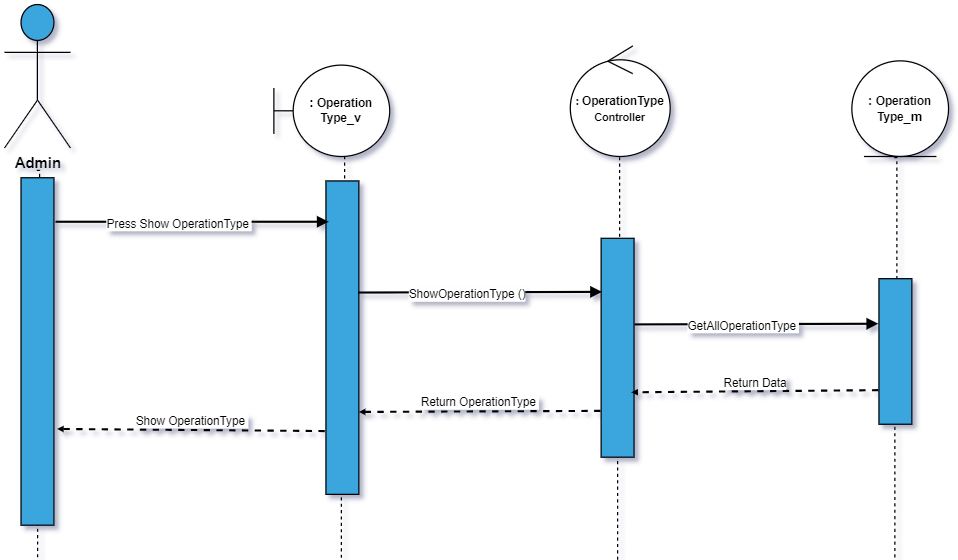
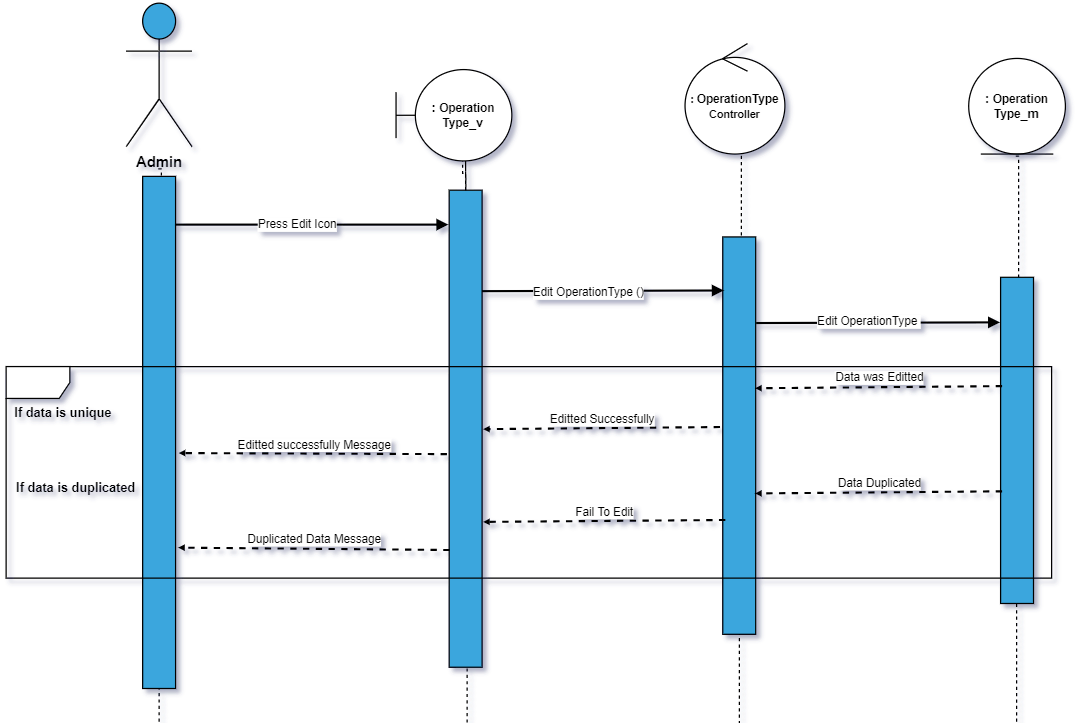
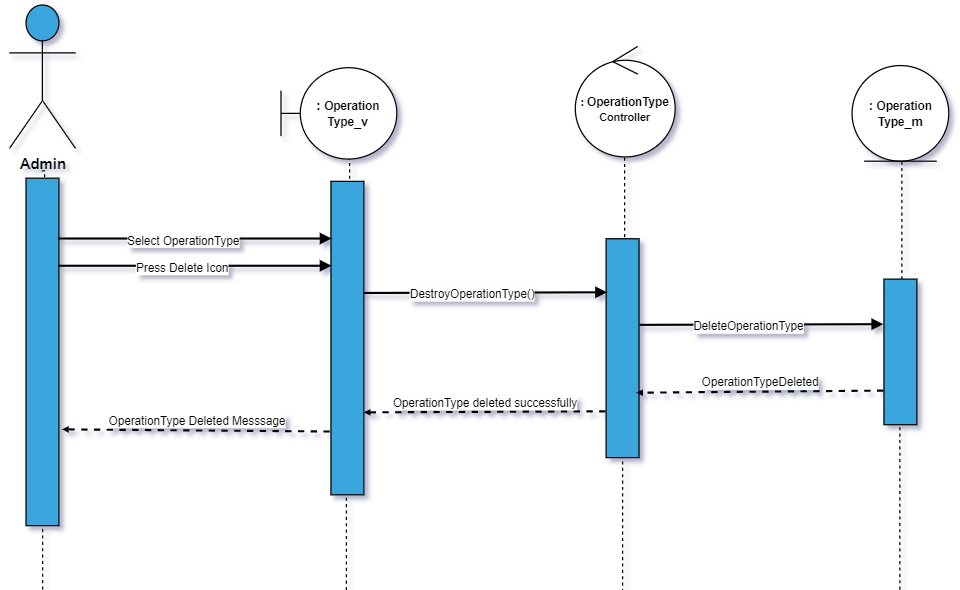


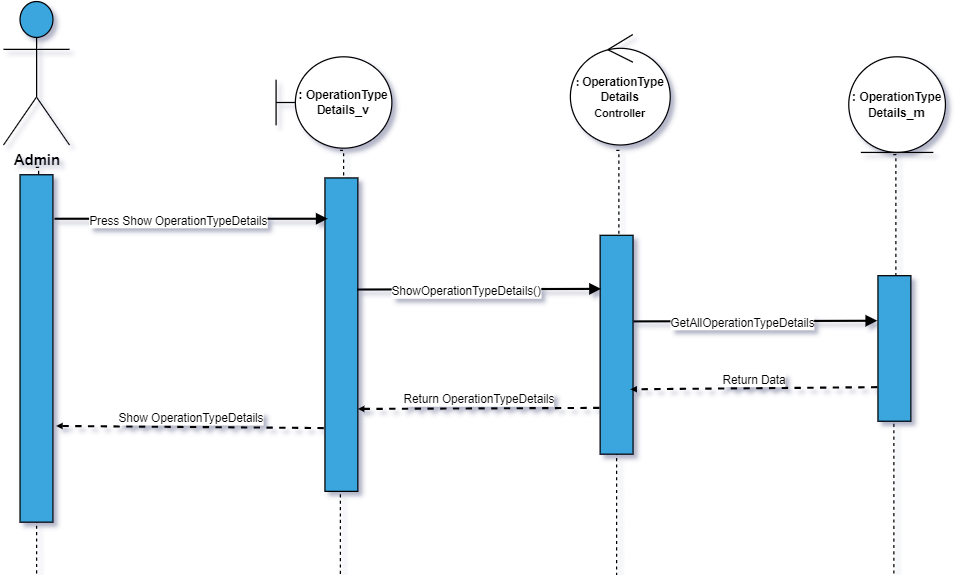
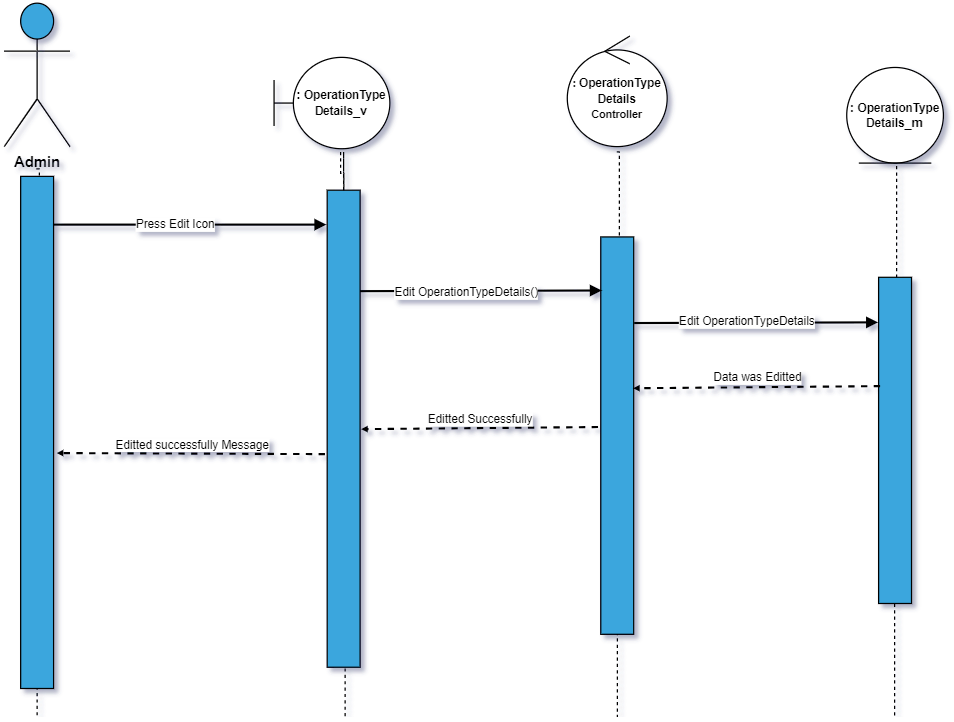
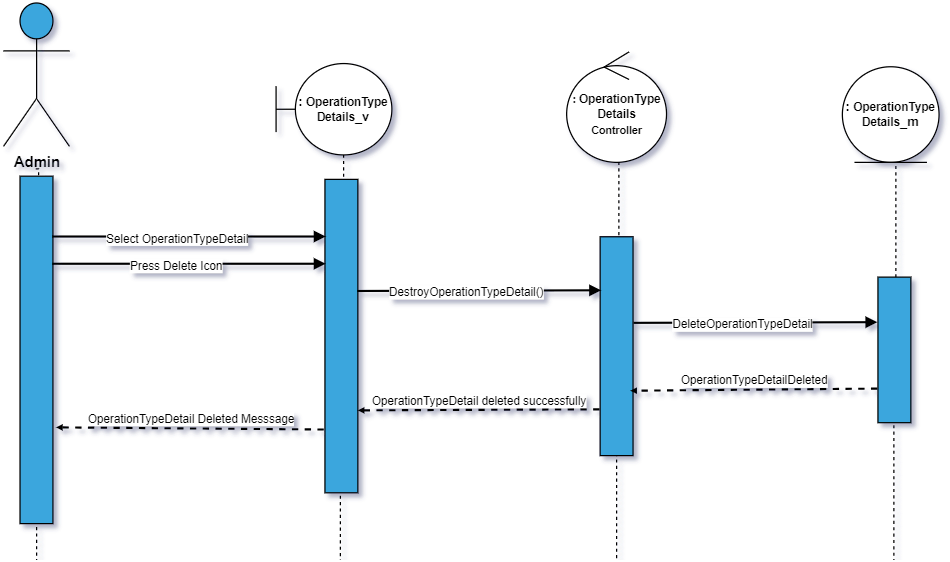
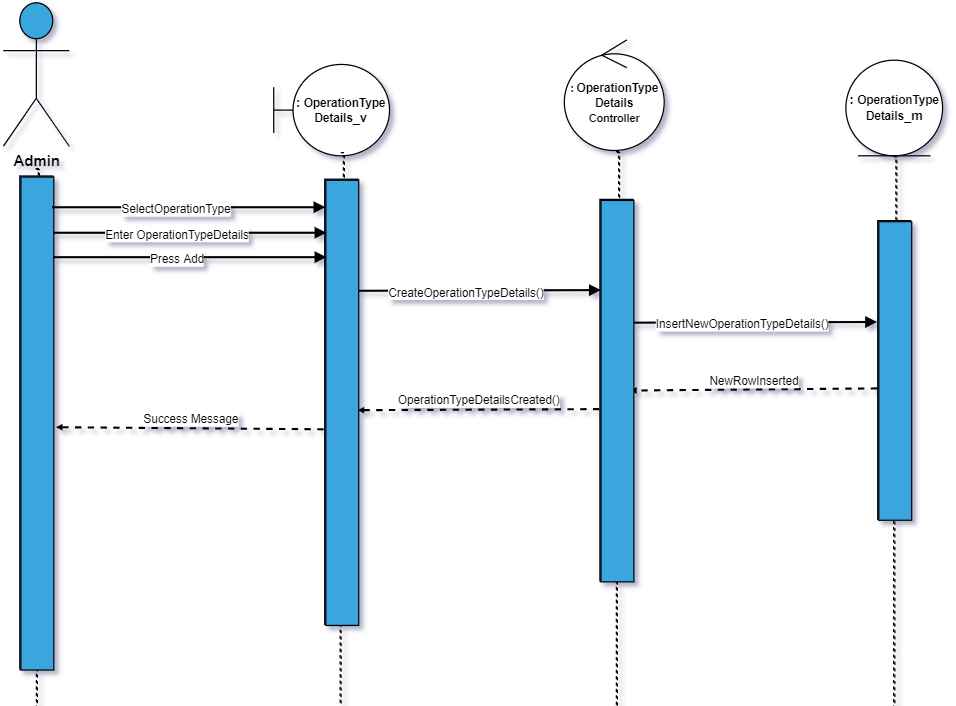


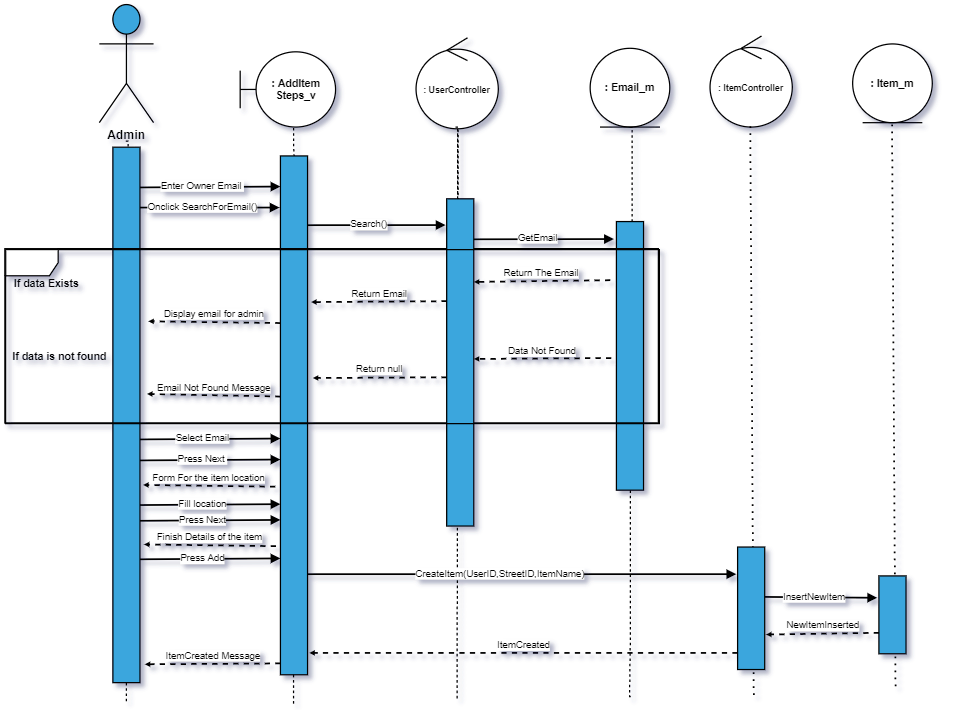


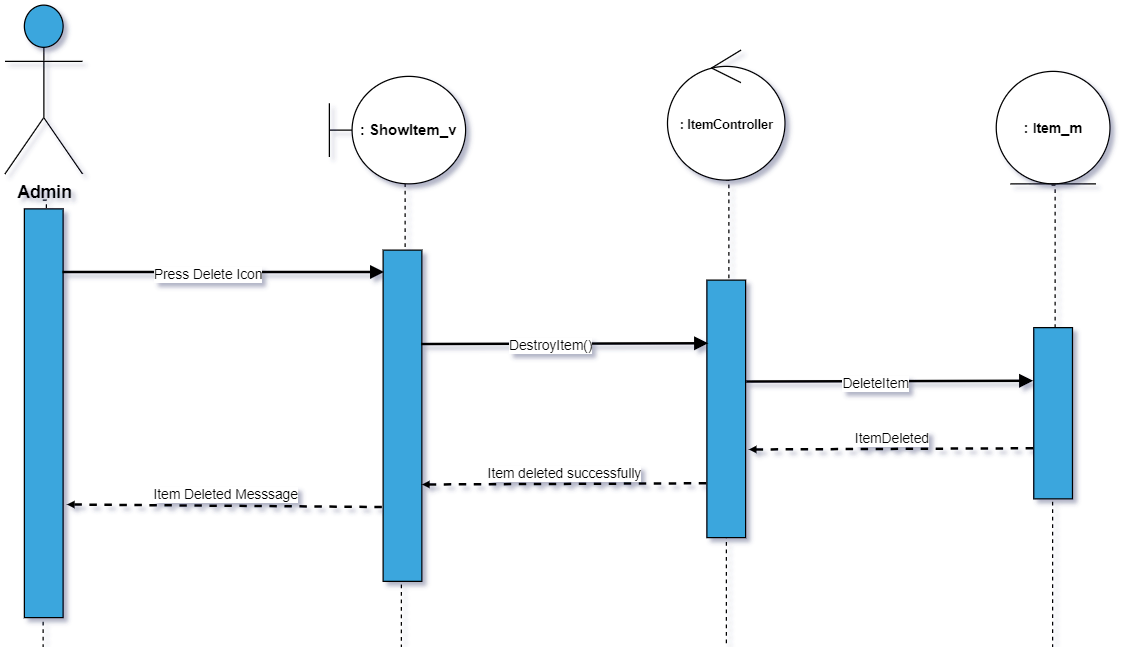
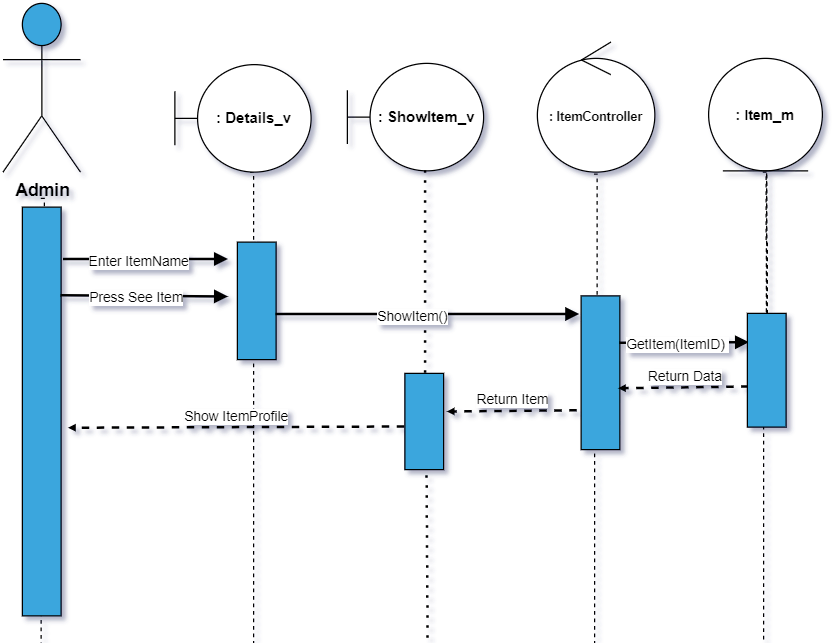


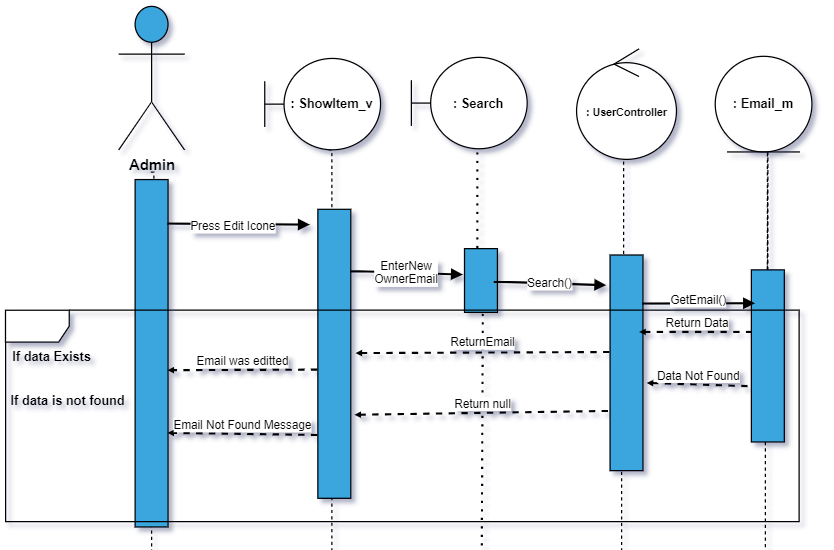


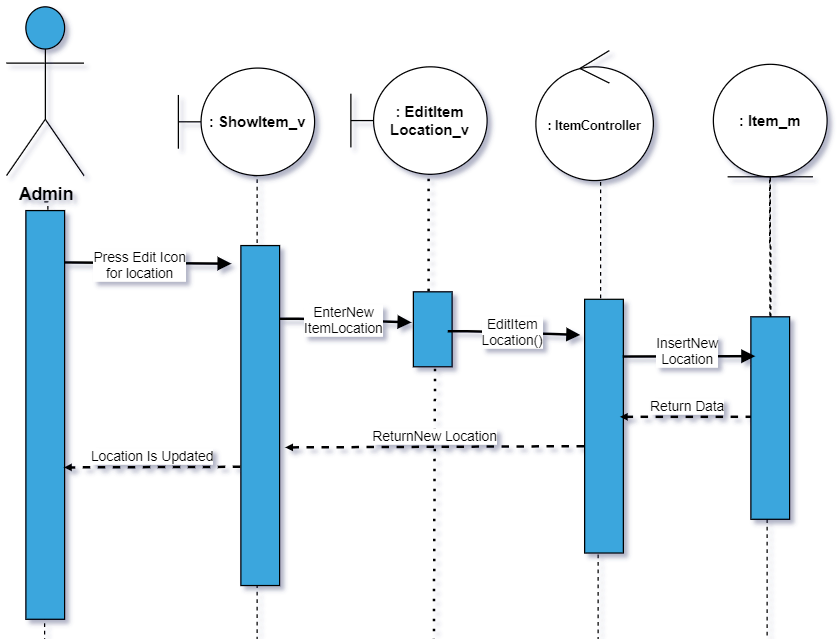


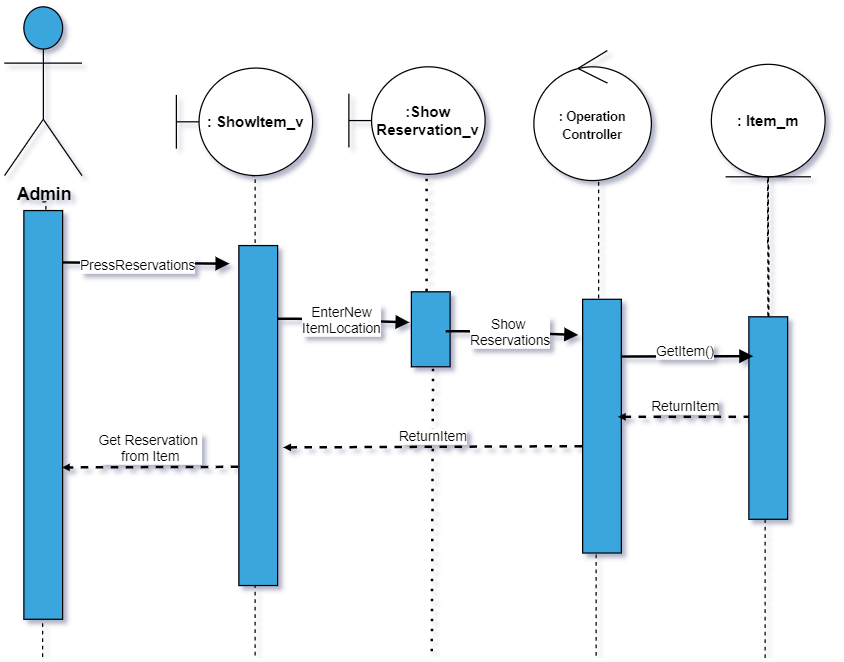


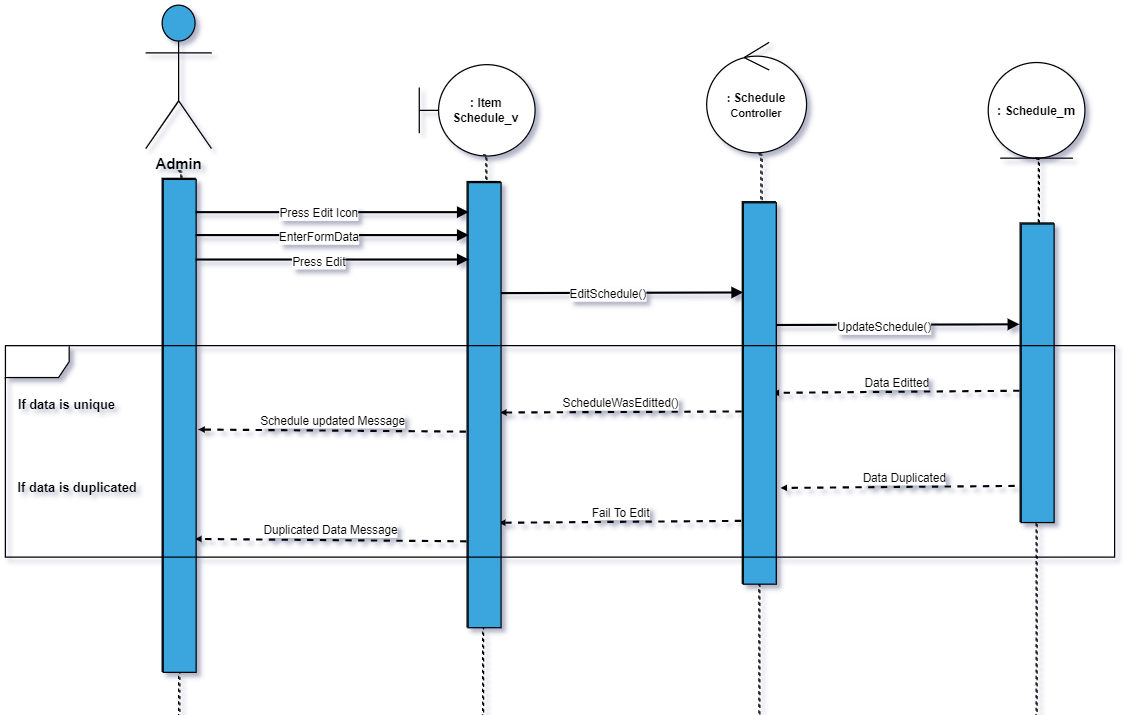
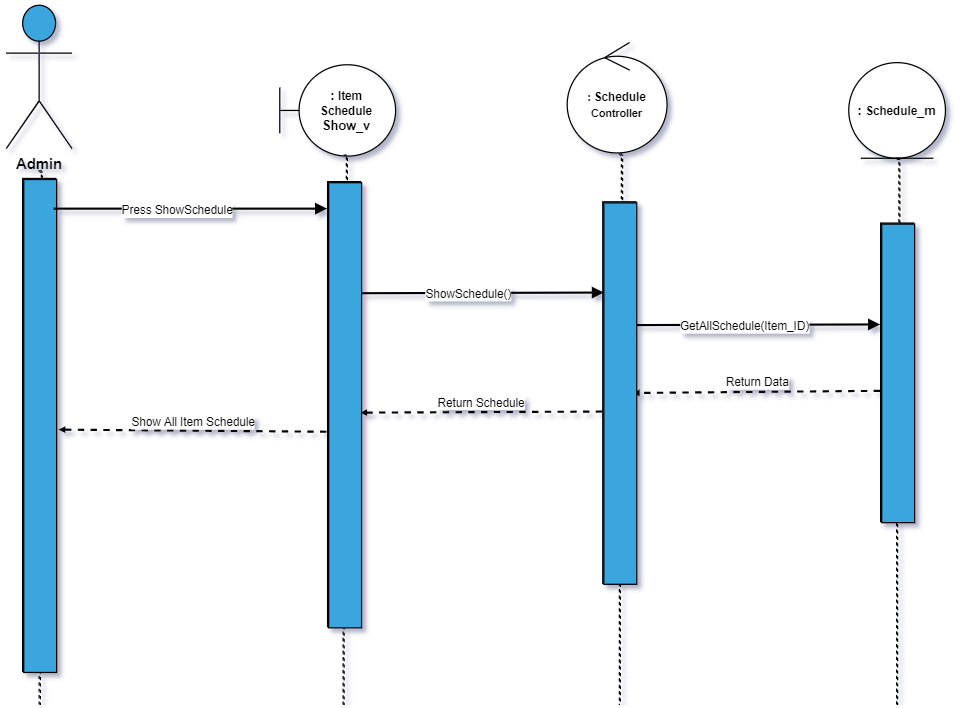
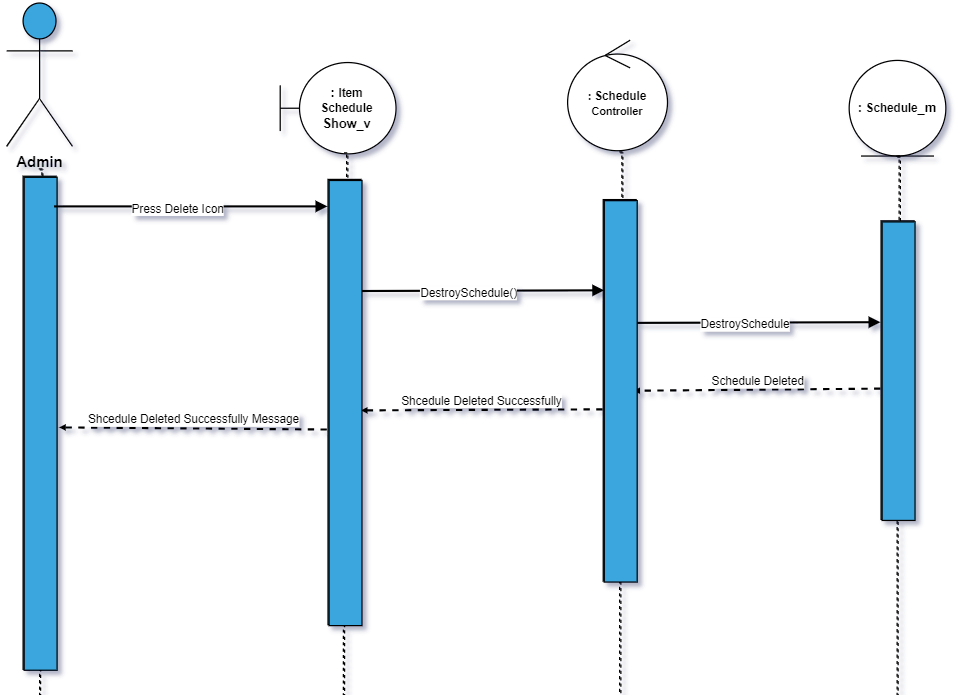
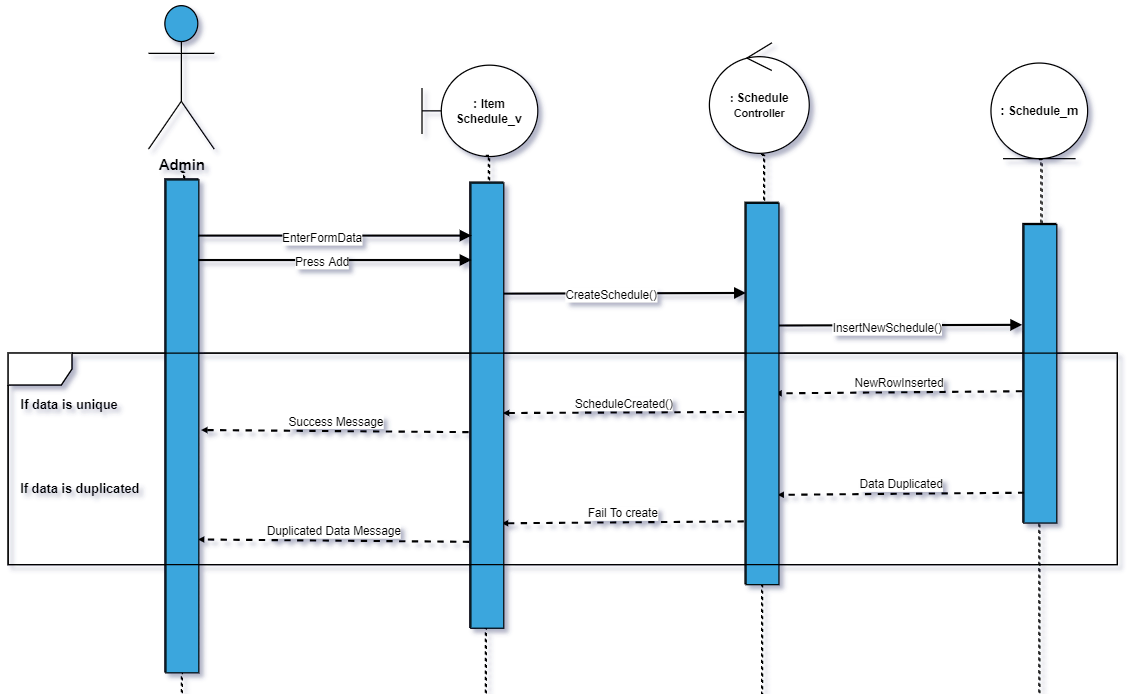


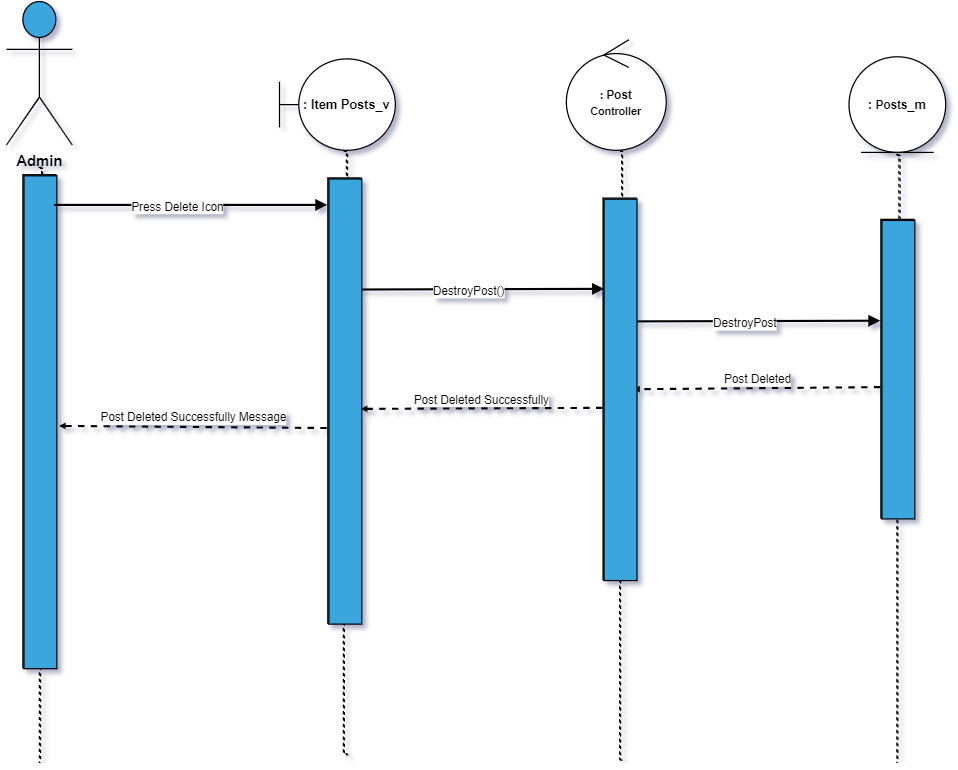
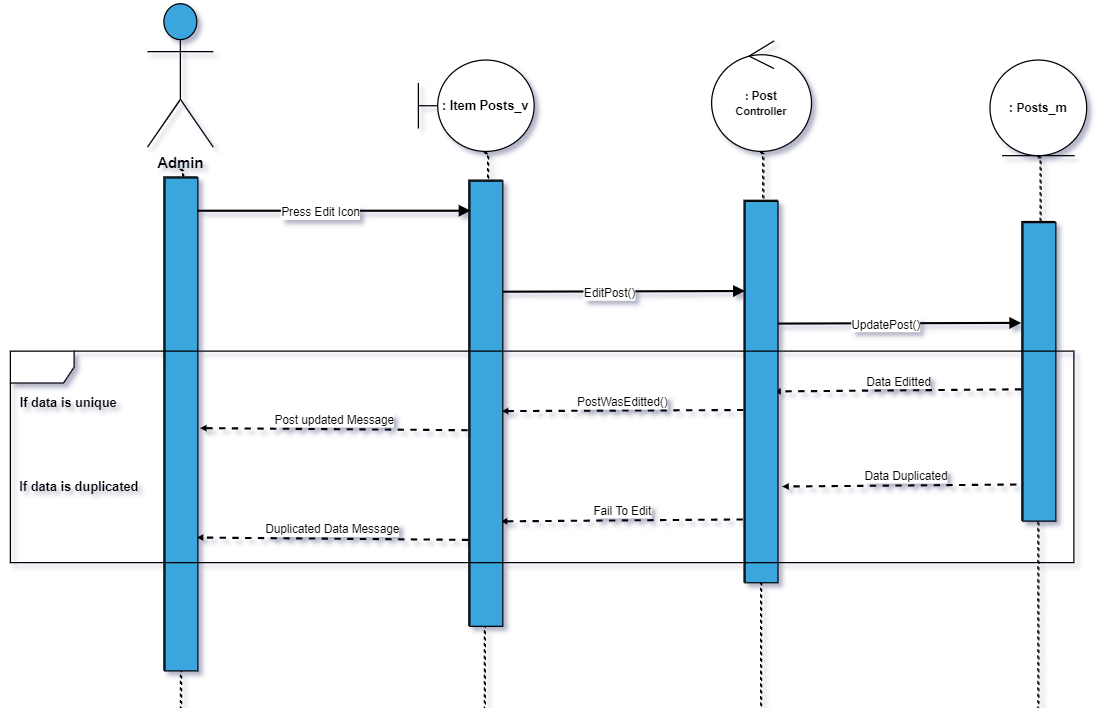
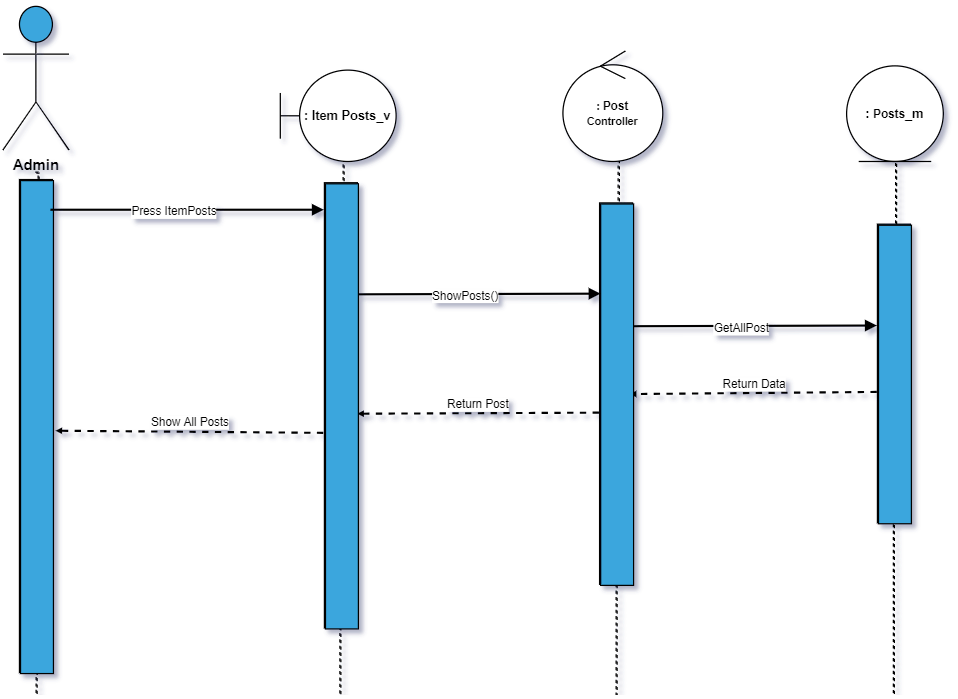
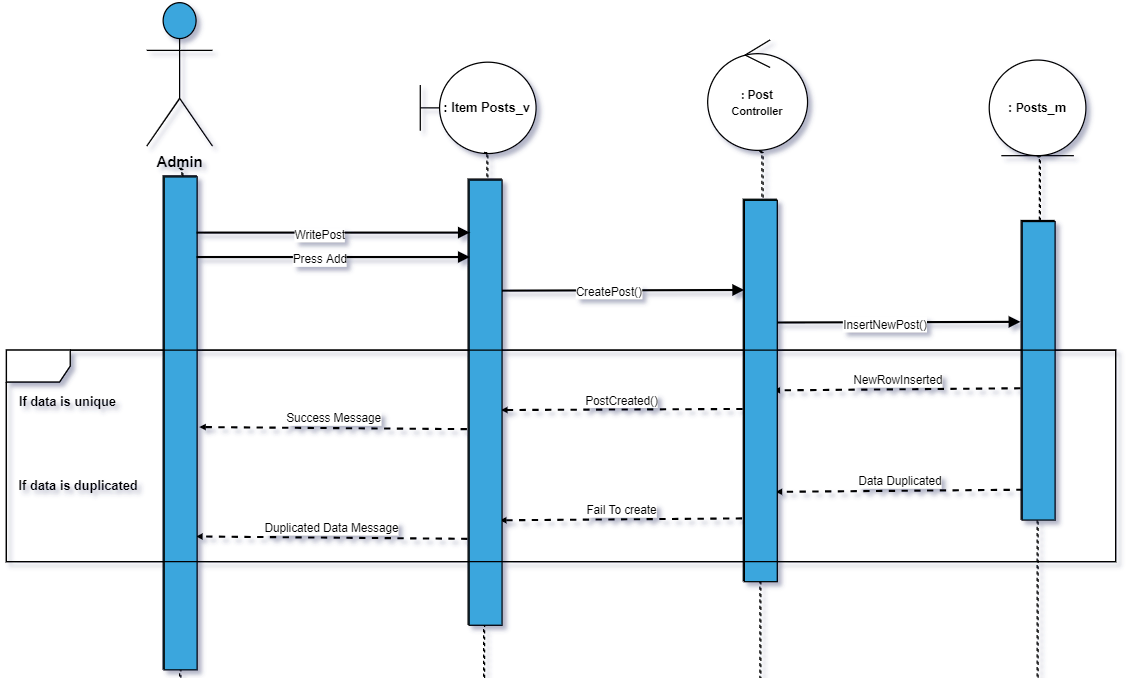


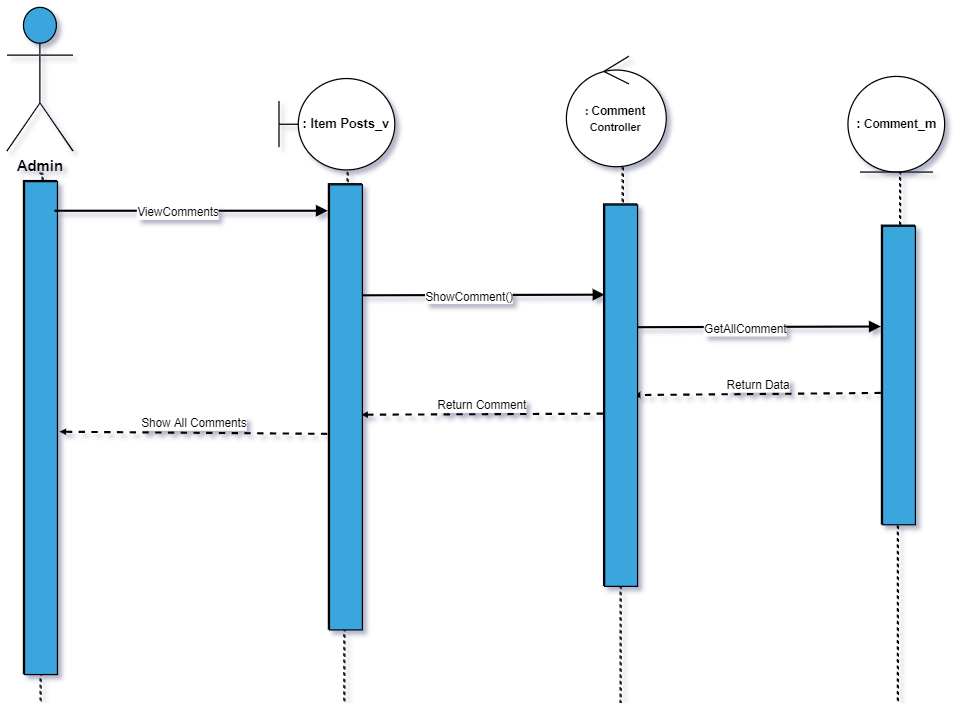
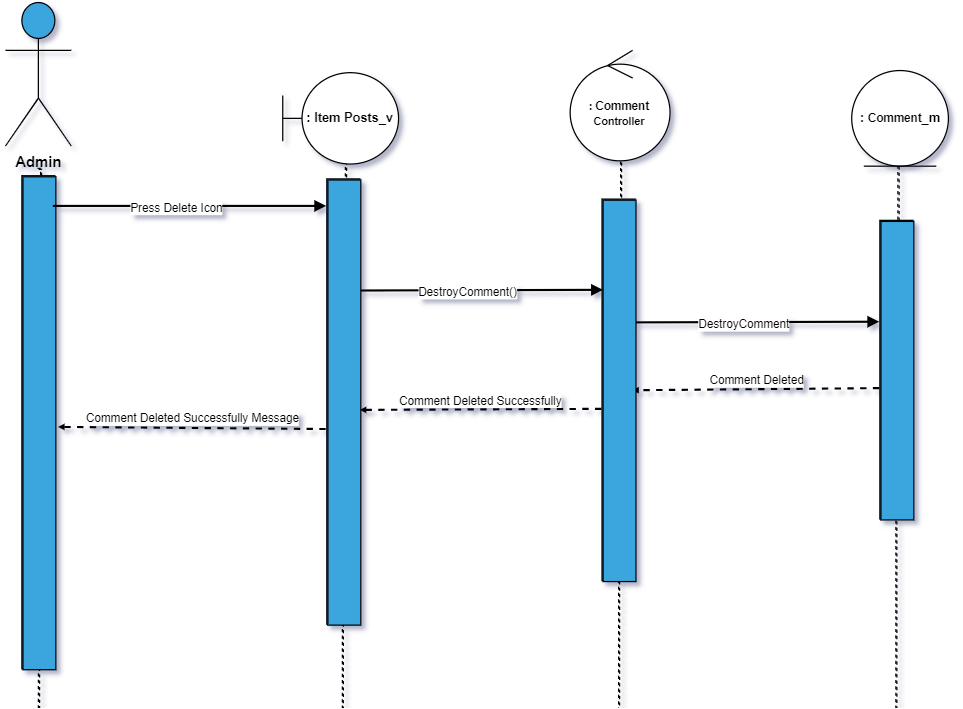


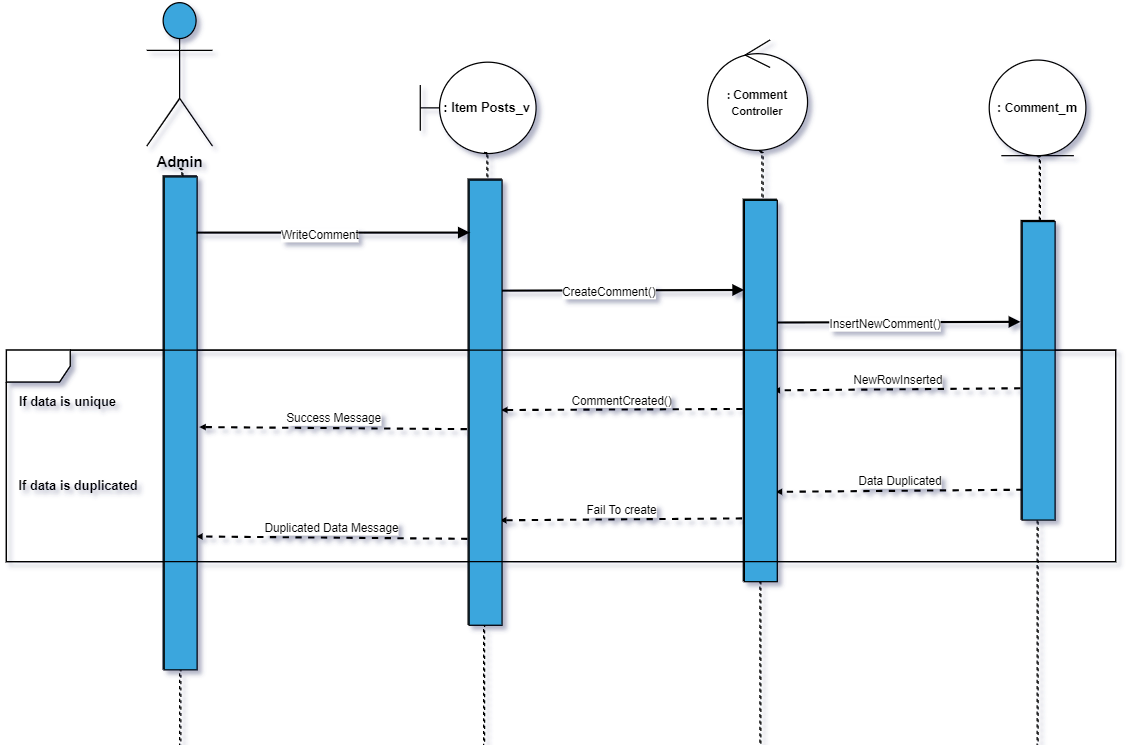


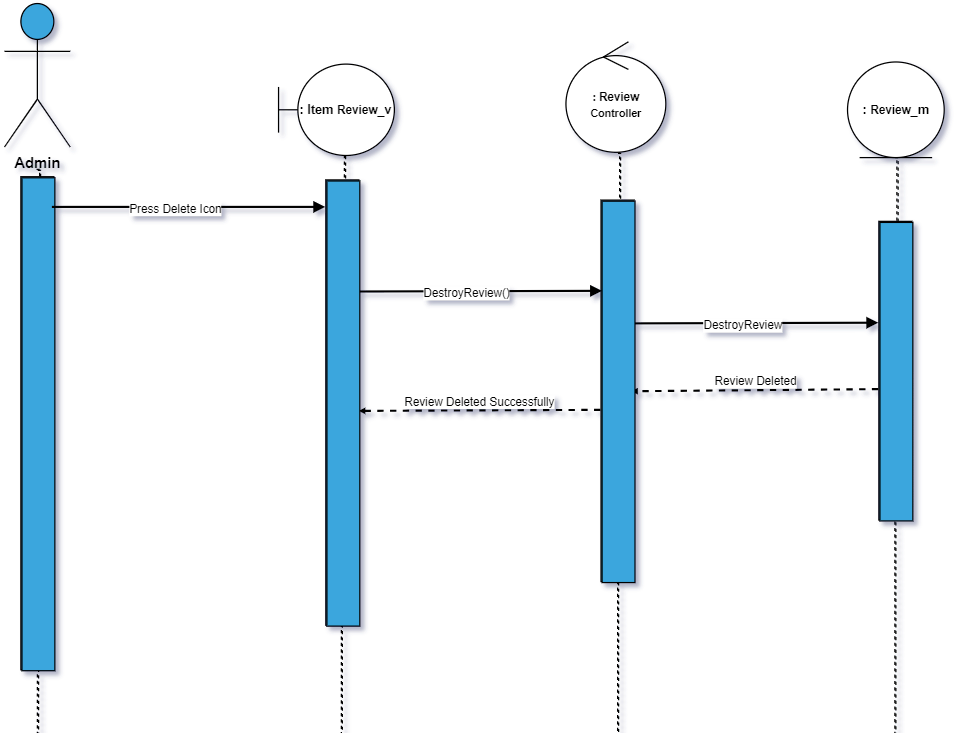
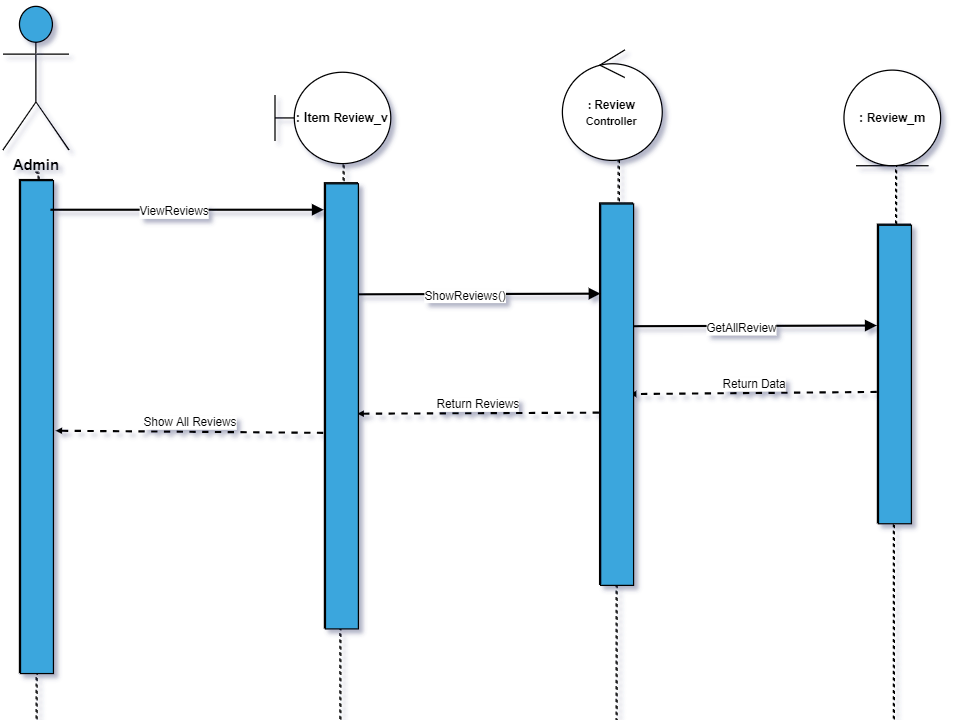
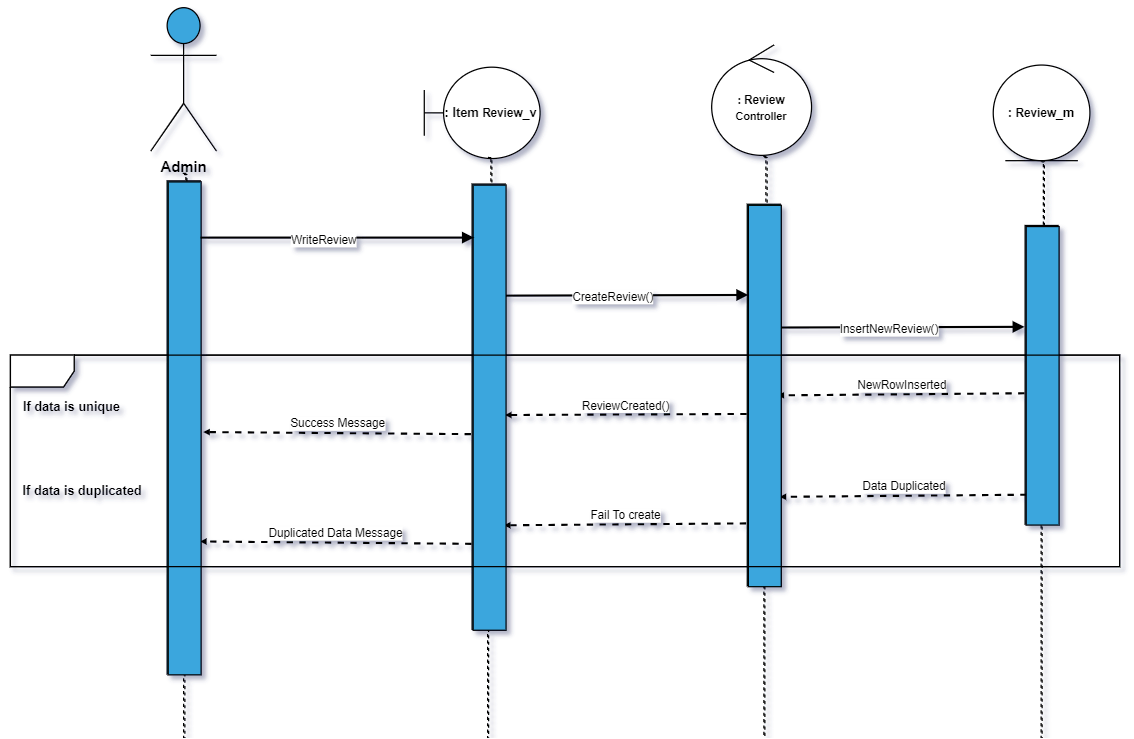


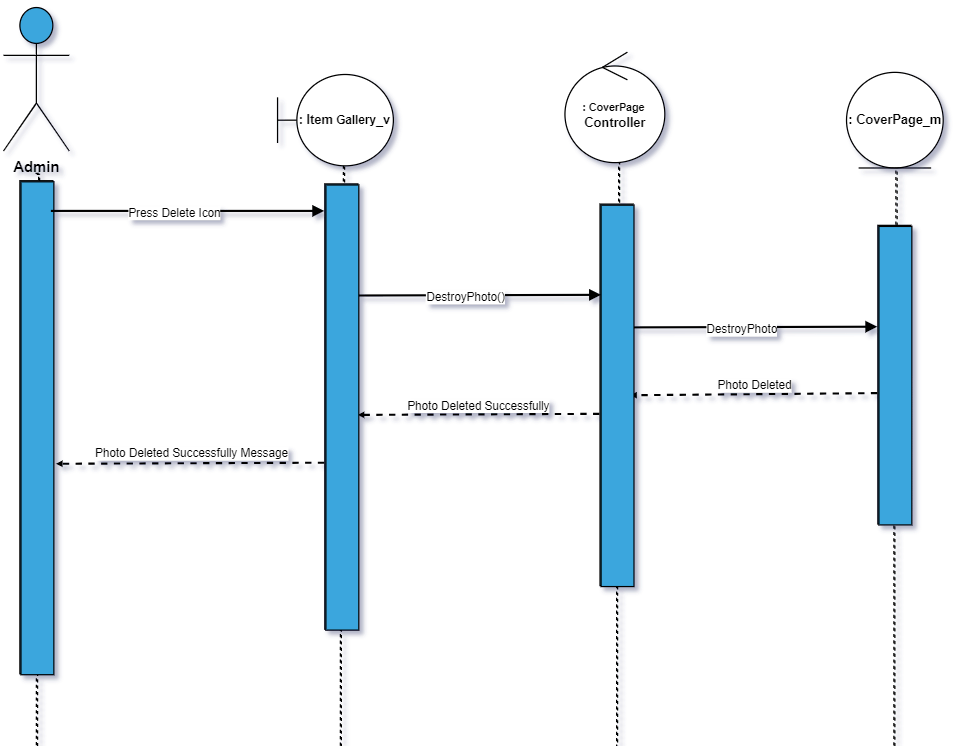
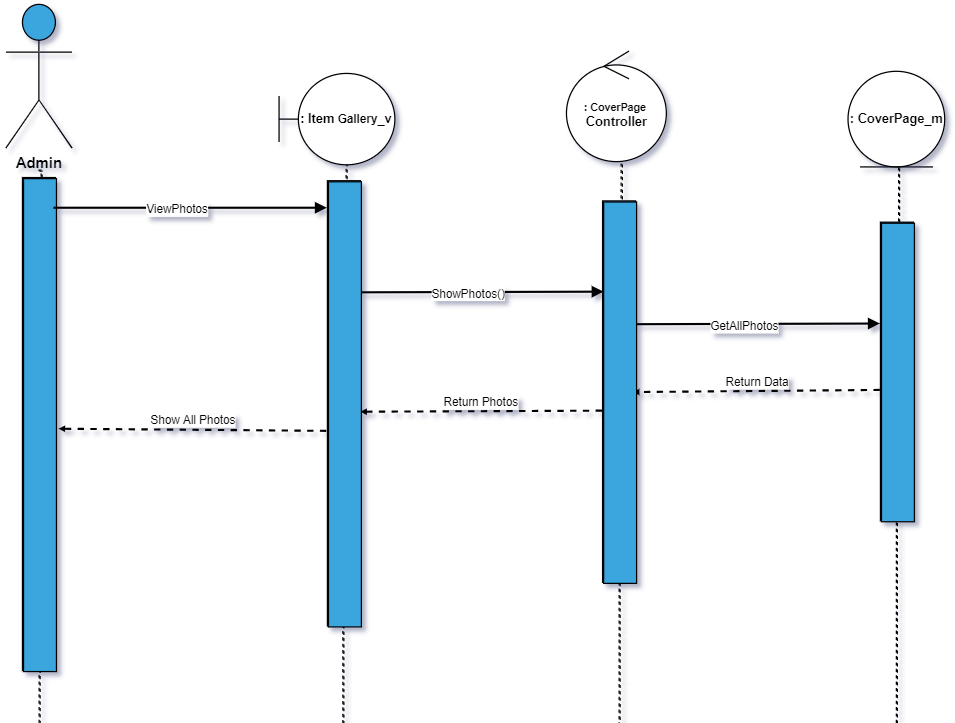
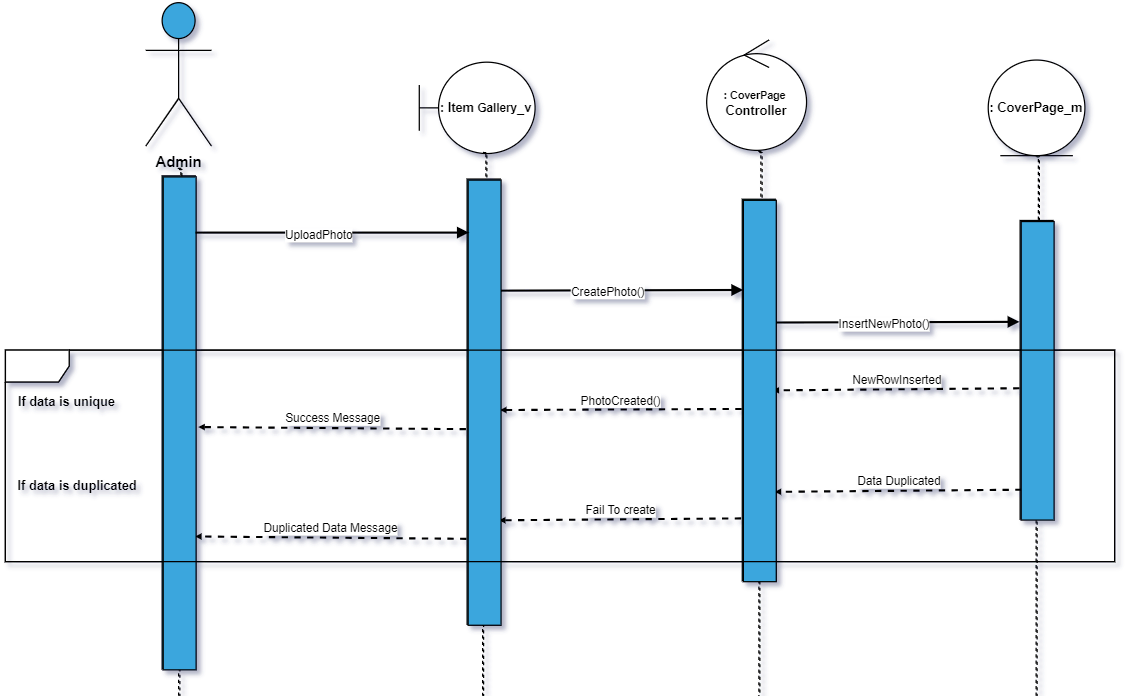






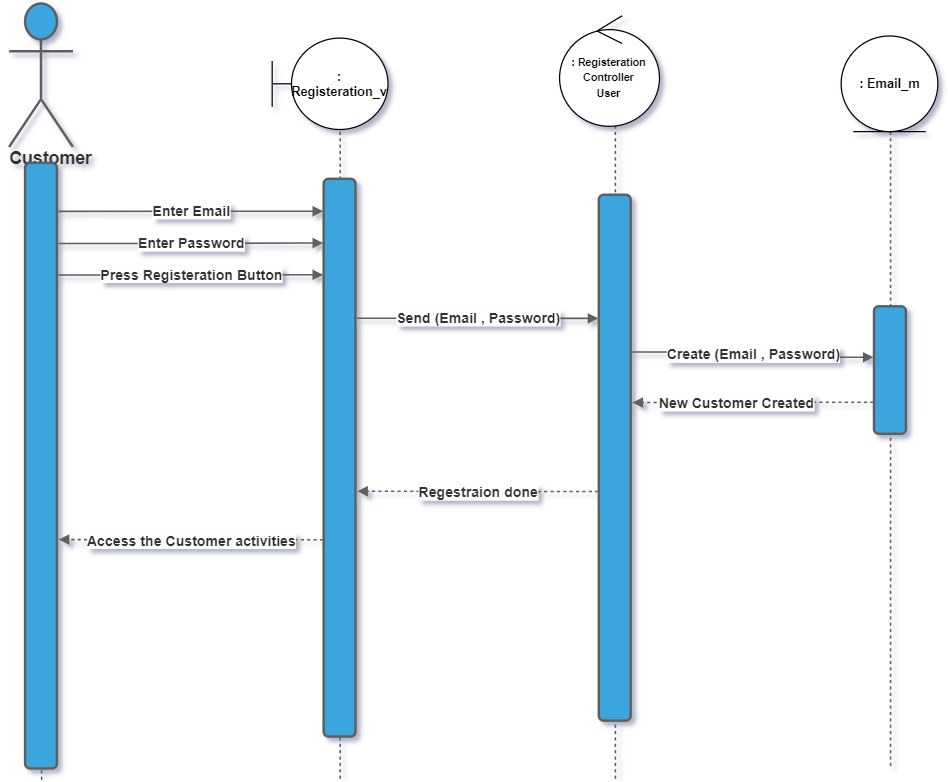






**Customer :**

**Diagram

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| AttachmentController | This controller was created for any process on attachments. | * **Index($id)**   This method takes item id as a parameter and returns all photos used in all posts of this item and return to Item\_Gallery blade. |
|  |  | * **Create($id)**   This method takes item id and create a post attachment for item with the id from the parameter and return back. |
|  |  | * **getAttachment($id)**   This method takes attachment id as a parameter and return attachment |
|  |  | * **getPostAttachments($item\_id)**   This method takes item id as a parameter and return post attachments of all posts of this item. |
|  |  | * **getAttachmentsOfPosts($id)**   This method takes post id as a parameter and return all attachments of a post. |
|  |  | * **destroy($id)**   This method takes attachment id and destroy the attachment with the id from the parameter. |
|  |  | * **getAttachmentOfuser($id)**   This method takes user id as a parameter and return all attachments of the post was uploaded by the user who created the post**.** |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CityController | This controller was created for any process on Cities. | * **Index($id)**   This method returns all cities , countries and states for the drop down list in the blade where to enter a new city. |
|  |  | * **Create()**   This method add a new city to the database. |
|  |  | * **show()**   This method return all cities to show cities blade. |
|  |  | * **destroy($id)**   This method finds all cities with state id from the request and return on json. |
|  |  | * **editCity()**   This method find city with the id from the request and changes city name by the name from the request and save it in the database then return back. |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CommentsController | This controller was created for any process on Comments and replays. | * **create()**   This method for adding a new comments and saving it into database then send notification to the user who created the post and show the comment. |
|  |  | * **reply()**   This method for adding a new replay and saving it into database then send notification to the user who entered the comment and show the replay. |
|  |  | * **CommentCreatedBy($id)**   This method takes user id as a parameter and return comment created by the user |
|  |  | * **destroyComment($id)**   This method takes comment id as a parameter and destroy this comment. |
|  |  | * **destroyReplay($id)**   This method takes replay id as a parameter and destroy this replay. |
|  |  | * **getPostComments($item\_id)**   This method takes item id as a parameter and return all comments of all posts on this item and group them by post id then return the comments. |
|  |  | * **getPostreplies($item\_id)**   This method takes item id as a parameter and return all replaies of all comments of all posts on this item and group them by parent comment then return the replies.   * **getPostCommentHomePage($post\_id)**   This method takes post id as a parameter and return all comments on this post with the name of the users and profile pictures. |
|  |  | * **getPostRepliesHomePage($post\_id)**   This method takes post id as an attachment and return all replies on this post with the name of the users and profile pictures. |
|  |  | * **GetCommentReply()**   This method finds all replies of comment and return with users name , profile pictures and all replies. |
|  |  | * **GetComments()**   This method finds all comment of a post and return them. |
|  |  | * **editComment()**   This method takes comment id , edited comment from the request and update it then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CountryController | This controller was created for any process on Countries. | * **index()**   This method for returning all countries. |
|  |  | * **create()**   This method for adding a new country to database and save it. |
|  |  | * **editCountry()**   This method takes country id and new country name form request and update the row in the database. |
|  |  | * **destroy()**   This method takes country id from the request and destroy this country. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CoverPageController | This controller was created for any process on Coverpages of items. | * **create()**   This method for adding a new coverpage it takes item id as a parameter and saving the new cover page into database then return back. |
|  |  | * **getCoverPhotoOfItem($id)**   This method takes item id as a parameter and return coverpage with this item id. |
|  |  | * **edit($id)**   This method takes coverpage id as a parameter and update a new cover page for this id then return back. |
|  |  | * **destroy($id,$path)**   This method takes coverpage id and the path to delete the coverpage from the database and the folder. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CoverPhotoController | This controller was created for any process on CoverPhotos. | * **create()**   This method takes the file name by the request and creates a new cover photo for the user and save it in database and folder then return back. |
|  |  | * **edit ()**   This method takes user id from authentication and edit his cover photo by the new one from the request and save it in the folder and database then return back. |
|  |  | * **getPhoto($id)**   This method takes user id as a parameter and return his cover photo. |
|  |  | * **update()**   This method get the photo with thephoto id from the request and update filename with the new one from the request and save it in the database and the folder then return back. |
|  |  | * **destroy($id,$path)**   This method takes photo id as a parameter and photo name then delete the photo from the folder and also from the database by it’s id. |
|  |  | * **sendCoverPhotoToProfile($id)**   This method takes user id as a parameter and return cover photo and it’s id. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CustomerHomeController | This controller was created for any process on Customer Home page. | * **index()**   This method for sending all states to the customer home page to make the customer able to search for item in any state preferred and also to check if the user is owner or customer then return to the customer home page with results. |
|  |  | * **indexPhoto()**   This method returns all states’ photos for the customer home page. |
|  |  | * **findItemInState()**   This method takes state name entered from the user by request and returns all items in this state and all states. |
|  |  | * **findItemInStateAndDate()**   This method takes state name entered from the user , arrival and departure by the request and returns all items in this state but also available for rent in TimeLine page . |
|  |  | * **FollowedItemPosts($item\_id)**   This method takes item id as a parameter and returns all posts for this followed item for the TimeLine page. |
|  |  | * **HomePagePosts()**   This method returns all posts of all items the user follows and also all newitems and most popular items to HomePage posts blade. |
|  |  | * **HomePageUserPosts()**   This method returns all posts of all userss the user follows and also all newitems and most popular items to HomePageUser posts blade.   * **showMyProfile()**   This method return user name , cover photo , profile photo , posts and post images to customer own profile page. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| CustomerController | This controller was created for any process on Customer. | * **itemProfile($id)**   This method takes item id as a parameter and return all item profile information for the customer |
|  |  | * **itemDetails($id)**   This method takes item id as a parameter and return all item details for item details page . |
|  |  | * **itemProfileGallery($id)**   This method takes item id as a parameter and return all item galleries for item profile gallery page. |
|  |  | * **itemProfileReviews($id)**   This method takes item id as a parameter and return all item reviews for item profile reviews page. |
|  |  | * **Show($id)**   **This method takes user id as a parameter and return all his posts , profile photo , cover photo and images to customer own profile if the user id is from the authentication else return to customer profile.** |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| OwnerController | This controller was created for any process on Owner. | * **itemProfile($id)**   This method takes item id as a parameter and return all item profile information for the Owner. |
|  |  | * **itemDetails($id)**   This method takes item id as a parameter and return all item details for item details page to the owner . |
|  |  | * **itemProfileGallery($id)**   This method takes item id as a parameter and return all item galleries for item profile gallery page to the owner. |
|  |  | * **itemProfileReviews($id)**   This method takes item id as a parameter and return all item reviews for item profile reviews page to the owner. |
|  |  | * **itemReservations($item\_id)**   This method takes item id as a parameter and returns all reservations and all it’s details of this item for the item profile reservation page for the owner. |
|  |  | * **itemManageSchedule($item\_id)**   This method takes item id as a parameter and returns all schedule of this item for the item profile manage schedule page for the owner. |
|  |  | * **index()**   This method returns all states to Customer Home blade for the owner. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| DataTypeController | This controller was created for any process on DataType. | * **index()**   This method return to data type page in admin pages. |
|  |  | * **create()**   This method takes data type name from the request and save it in the data base then return back. |
|  |  | * **show()**   This method return all data types to show data type page in admin pages. |
|  |  | * **edit()**   This method takes data type id and the new name from the request and edit it in the data base then return back. |
|  |  | * **destroy()**   This method takes data type id from the request and delete this data type from the data base then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| DetailsController | This controller was created for any process on Details. | * **index()**   This method return all main types , sub types , property and property details for details page in admin pages. |
|  |  | * **create()**   This method takes all details by the request and save each detail in the data base then return back. |
|  |  | * **editDetails()**   This method takes details from the request and edit each detail and save it in the data base then return back. |
|  |  | * **AddImage()**   This method takes file name from the request and save it in the database if the user want to enter the detail value as a image then return back. |
|  |  | * **show()**   This method return all details for show details page in admin pages. |
|  |  | * **edit()**   This method takes detail id and the new detail name from the request and edit it then save it in the database and return back. |
|  |  | * **destroy()**   This method takes detail id from the request and delete this detail from the database |
|  |  | * **destroydetails()**   This method takes all details of property and delete all its details then return back. |
|  |  | * **FindDetailsForShow()**   This method takes property diff id from the request and get all it’s details for the show. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| FollowedUsersController | This controller was created for any process on Follow. | * **FollowedUser($id)**   This method takes the id of the user followed as a parameter and save a relation ship “follow” between the user from the authentication and the user he want to follow then send a notification to the user he was followed then return back. |
|  |  | * **UnfollowUser($id)**   This method takes id of the user was followed as a parameter and delete this relation from the database then return back. |
|  |  | * **checkFollow($id)**   This method takes id of the user was followed as a parameter and check if there is a relation ship between him and the user from the authentication. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| MainTypesController | This controller was created for any process on MainTypes. | * **index()**   This method return to main types page in admin pages. |
|  |  | * **create()**   This method takes Main Type name from the request and save it in the database then return back. |
|  |  | * **show()**   This method returns all main types to show them in main types show page. |
|  |  | * **Edit()**   This method takes the new main type and it’s id then edit it and save it in the data base then return back. |
|  |  | * **Destroy()**   This method takes main types id and delete it from the database then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| HomeController | This controller was created for any process on MainTypes. | * **index()**   This method return to Home page. |
|  |  | * **\_\_construct()**   This method allow to access all pages if user is authenated. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| NotificationController | This controller was created for any process on Notification. | * **index()**   This method return all notifications |
|  |  | * **create($from\_id,$to\_id,$notification)**   This method takes notification , from and to as a parameter and save it in the data base then return back. |
|  |  | * **create($from\_id,$to\_id,$notification,$redirect\_to)**   This method takes notification , from , to and redirect to as a parameter and save it in the data base then return back. |
|  |  | * **viewNotification()**   This method takes notification id from the request and return the notification as a viewed then return back. |
|  |  | * **show()**   This method return all notifications. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| PaymentController | This controller was created for any process on Payment. | * **create()**   This method return all notifications |
|  |  | * **create($from\_id,$to\_id,$notification)**   This method takes notification , from and to as a parameter and save it in the data base then return back. |
|  |  | * **create($from\_id,$to\_id,$notification,$redirect\_to)**   This method takes notification , from , to and redirect to as a parameter and save it in the data base then return back. |
|  |  | * **viewNotification()**   This method takes notification id from the request and return the notification as a viewed then return back. |
|  |  | * **show()**   This method return all notifications. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| PostAttachmentController | This controller was created for any process on PostAttachment. | * **deleteImgFromGallery()**   This method takes attachment id from request and return all attachment |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| PostsController | This controller was created for any process on Posts. | * **index()**   This method takes item id as a parameter and return all posts of this item to item posts page . |
|  |  | * **create($id)**   This method takes item id as a parameter and create post for this item with it’s attachments then return back. |
|  |  | * **postCreatedBy($id)**   This method takes post id as a parameter and return the id of the user created this post. |
|  |  | * **userPosts($id)**   This method takes user id as a parameter and return all posts of this user. |
|  |  | * **Destroy($id)**   This method takes post id as a parameter and delete this post from the data base then return back. |
|  |  | * **getItemPosts($Item\_Id)**   This method takes item id as a parameter and return all posts of this item. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| ProfilePhotoController | This controller was created for any process on Profile Photo. | * **getPhoto($id)**   This method takes profile photo id as a parameter and return this photo. |
|  |  | * **create()**   This method takes profile photo name from the request and save it in the database and the folder then return back. |
|  |  | * **edit()**   This method takes profile photo name from the request and edit this photo in the data base by user id with the new photo name and return back. |
|  |  | * **update()**   This method takes profile photo and photo id from the request and update the photo with this id with the new profile photo then return back. |
|  |  | * **Destroy($id,$path)**   This method takes profile photo id and the path the delete this photo from the folder and the database then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| PropertyDetailsController | This controller was created for any process on Property Details. | * **index()**   This method return all property details for property details page. |
|  |  | * **create()**   This method takes property details from the request and save it in the database then return back. |
|  |  | * **show()**   This method return all property details for show in property details show page. |
|  |  | * **find()**   This method takes property detail id from the request and return property detail. |
|  |  | * **FindDetailsForForm()**   This method takes property id and return all property details of this property. |
|  |  | * **findDetailsForFormInOwner()**   This method takes property diff id and property id from the request and return all property details and all properties. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| RegionController | This controller was created for any process on Regions. | * **index()**   This method return all countries , states and cities for region page in admin pages. |
|  |  | * **create()**   This method takes country , state , city and region from the request and save region in the database then return back. |
|  |  | * **show()**   This method return all regoins for show in region show page. |
|  |  | * **Destroy(id)**   This method takes region id as a parameter and delete this region from the database then return back. |
|  |  | * **Findregion()**   This method takes city id from the request and return all regions in this city. |
|  |  | * **EditRegion()**   This method takes region id and region name from the request and edit it in the database then return back. . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| UserTypesController | This controller was created for any process on UserTypes. | * **index()**   This method return all UserTypes for User Types Page in admin pages. |
|  |  | * **create()**   This method takes User Type Name from the request and save User Type Name in the database then return back. |
|  |  | * **show()**   This method return all UserTypes for User Types Show Page in admin pages. |
|  |  | * **edit ()**   This method takes user type id from the request and edit it’s name by the new one from the request and save it in the database then return back. |
|  |  | * **destroy()**   This method takes user type id from the request and delete it from the database then return back to user types show page. |
|  |  | * **get\_user\_types()**   This method return all user types and all users for users show page.   * **getUser()**   This method takes user type id from the request and check in table type of users for all users with this user type id then return all these users and their phone number , email by json . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| ReservationController | This controller was created for any process on Reservations. | * **show()**   This method return all reservation operation details to reservation show page . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| ReviewController | This controller was created for any process on Reviews. | * **index()**   This method takes item id as a parameter and return all item reviews to item reviews page. |
|  |  | * **create()**   This method takes item id and user id from the request and check if there is a review with these two ids and edit it with the new review content and stars from the request and save it if there is no review with these two ids will create a new review and fill it’s information by the data from the request , save it in the data base and return done. |
|  |  | * **getItemReviews($item\_id)**   This method takes item id as a parameter and return all item reviews . |
|  |  | * **getItemRate($item\_id)**   This method takes item id as a parameter and calculate the rate of item by summing all stars for this item and how many reviews on this item then calculating the rate . |
|  |  | * **destroy($id)**   This method takes review id as a parameter and delete this review from the data base and return back. |
|  |  | * **EditRegion()**   This method takes region id and region name from the request and edit it in the database then return back. . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| StateController | This controller was created for any process on States. | * **index()**   This method return all countries for add state page in admin pages. |
|  |  | * **create()**   This method takes country id and state name from the request and save state in the database then return back. |
|  |  | * **show()**   This method return all states for show in states show page. |
|  |  | * **Destroy(id)**   This method takes state id as a parameter and delete this state from the database then return back. |
|  |  | * **findstate()**   This method takes country id from the request and return all states in this country. |
|  |  | * **edutState()**   This method takes state id and new state name from the request and edit it in the database then return back. |
|  |  | * **Findstatebyname($statename)**   This method takes state name as a parameter and return state id with this state name. |
|  |  | * **getStates()**   This method return all state names. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| StatePhotoController | This controller was created for any process on StatePhotos. | * **index()**   This method return all states for state photo page. |
|  |  | * **create()**   This method takes file name from the request and save it in the folder and create a new attachment then take the attachment id to save it in state photo table by state id and create a state photo then return back |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| StreetController | This controller was created for any process on Streets. | * **index()**   This method return all countries , states , cities and regions for add street page in admin pages. |
|  |  | * **create()**   This method takes country , state , city , region ids and street name from the request and save street in the database then return back. |
|  |  | * **show()**   This method return all streets for show in street show page. |
|  |  | * **Destroy(id)**   This method takes street id from the request and delete this street from the database then return back. |
|  |  | * **findstreet()**   This method takes region id from the request and return all streets in this region. |
|  |  | * **editStreet()**   This method takes street id and new street name from the request and edit it in the database then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| SubTypePropertyController | This controller was created for any process on SubTypeProperties. | * **index()**   This method return all maintypes and sub types for SubTypeProperty page. |
|  |  | * **create()**   This method takes maintype id and subtype id and Property name from the request and save Property name in the database then return back. |
|  |  | * **show()**   This method return all Property names for show in SubTypeProperty show page. |
|  |  | * **Destroy(id)**   This method takes property id from the request and delete this property from the database then return back. |
|  |  | * **find()**   This method takes subtype id from the request and return all properties in this region. |
|  |  | * **editStreet()**   This method takes property id and new property name from the request and edit it in the database then return back. |
|  |  | * **Property\_select($item\_id,$sub\_type\_id)**   This method takes subtype id as a parameter and return all properties for this sub type with item id to properties select page . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| SubTypeController | This controller was created for any process on SubType s. | * **index()**   This method return all maintypes for SubType page. |
|  |  | * **create()**   This method takes maintype id and subtype name from the request and save subtype name in the database then return back. |
|  |  | * **show()**   This method return all maintypes names and subtypes names for show in SubType show page. |
|  |  | * **Destroy(id)**   This method takes subtype id from the request and delete this subtype from the database then return back. |
|  |  | * **find()**   This method takes maintype id from the request and return all subtypes in this maintype. |
|  |  | * **update($id)**   This method takes subtype id as a parameter and maintype id and subtype name from the request and edit the main type of this sub type then return back. |
|  |  | * **Property\_select($item\_id,$sub\_type\_id)**   This method takes subtype id as a parameter and return all properties for this sub type with item id to properties select page . |
|  |  | * **getSubTypeById($id)**   This method takes subtype id as a parameter and return sub type name. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| UserController | This controller was created for any process on User. | * **create()**   This method for adding a new user information (name , gender , password , email , phone number and user type) saving it into database and return back. |
|  |  | * **index()**   This method return all user types for add user page. |
|  |  | * **AdminProfile()**   This method takes user id from the authentication and return user name , email , phone number to admin profile page. |
|  |  | * **EditUserProfileVeiw()**   This method takes user id from the authentication and return user name , email , phone number and profile photo to edit user profile page. |
|  |  | * **EditUserProfile($id)**   This method takes user id from the authentication and user name , email and phone number from request to edit user profile page and return back. |
|  |  | * **checkIfOwner()**   This method takes user id from the authentication and check in table type of users if this user id has a usertype id of the owner or not. |
|  |  | * **BeOwner($user\_id)**   This method takes user id as a parameter and create a row data for this id in table type of user making his user type equal owner id and filling all his data then return back.   * **getItemWithOwnerName($item\_id)**   This method takes item id as a parameter and return item with the owner name of this item. |
|  |  | * **search()**   This method takes email from the request and return it from the data base if found . |
|  |  | * **destroy()**   This method takes user id from the request and delete this user , his phone number and his email from data base then return back. |
|  |  | * **editUserName()**   This method takes user id , new first , middle and last name then edit it in the database then return back. |
|  |  | * **editUserEmail()**   This method takes user id and the new email then edit it in the database then return back. |
|  |  | * **editUserPhoneNumber()**   This method takes user id and the new phone number then edit it in the database then return back. |
|  |  | * **getUserName($user\_id)**   This method takes user id as a parameter and return user name. |
|  |  | * **FollowedItem($item\_id)**   This method takes item id as a parameter and make relation between the user from the authentication and this item “follow” then send notification to the owner of this item that this user started to follow his item. |
|  |  | * **UnFollowedItem($item\_id)**   This method takes item id as a parameter and the user id from the authentication and delete row of data of these two ids from table followed item by user then return back. |
|  |  | * **changePassword()**   This method takes user id from authentication and the new password from the request and changes it in the database then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| ItemController | This controller was created for any process on Items. | * **create()**   This method for adding a new user information (name , gender , password , email , phone number and user type) saving it into database and return back. |
|  |  | * **index()**   This method return all main types and items for item page in admin pages. |
|  |  | * **Index1()**   This method return all user types and countries for add item steps. |
|  |  | * **Show()**   This method takes item id from the request and return owner of item , item location and item details to show item page in admin pages. |
|  |  | * **itemShow()**   This method return to item page. |
|  |  | * **SubTypeShow()**   This method return all main types and subtypes to item sub type show. |
|  |  | * **OwnerSelectProperty($item\_id,$subtype\_id)**   This method takes item id and subtype id as a parameter and return subtype and properties to owner select details page.   * **SelectSubType($item\_id)**   This method takes item id as a parameter and return subtypes and main types to owner select sub type. |
|  |  | * **OwnerAddItem()**   This method takes user id from authentication , street id , item name , longitude and latitude from request and create item with owner id of this user id then return to function SelectSubType($item\_id) to continue adding item. |
|  |  | * **create()**   This method takes user id ,street id and item name from the request and add item in the data base then return to function SubTypeShow($item\_id). |
|  |  | * **ShowEditLocation(id)**   This method return all regions , countries , states , cities and streets to edit item location page. |
|  |  | * **EditLocation($id)**   This method takes item id as a parameter and find the item then edit street id by the new one from the request and return to function show($item\_id). |
|  |  | * **showEditUser($id)**   This method takes item id as a parameter and return back to edit item user page. |
|  |  | * **EditUser($item\_id)**   This method takes item id as a parameter and find the item then edit user id by the new one from the request and return to function show($item\_id). |
|  |  | * **searchEmail()**   This method takes email from the request and frind it in the database then return back. |
|  |  | * **destroy($item\_id)**   This method takes item id as a parameter and delete this item from the database then return back. |
|  |  | * **EditItemMap($Item\_Id)**   This method takes item id as a parameter and the new latitude and longitude from the request and changes it in the database then return back. |
|  |  | * **Getnewestitems()**   This method order items by created at date then return all items from the newest to the oldest |
|  |  | * **Getpopularitems()** |
|  |  | * **Getowner($id)**   This method takes item id as a parameter and return user id of this item . |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| OperationControlle | This controller was created for any process on Operations. | * **create($item\_id)**   This method takes item id as a parameter and user id from the authentication then create a new operation the data base then return back. |
|  |  | * **index()**   This method return all operation types for operation detail. |
|  |  | * **calculatedays()**   This method takes schedule id start and end date from the request then calculate total price and price per date and return them. |
|  |  | * **Show()**   This method return all operation types to operation types show page. |
|  |  | * **finddetail()**   This method takes operation detail id from the request and return operation detail |
|  |  | * **edit()**   This method takes operation types id , operation type name and edit operation type then save it in the data base and return back. |
|  |  | * **editDetail()**   This method takes operation detail id , operation detail name and edit operation detail then save it in the data base and return back.   * **destroy()**   This method takes operation type id and delete this operation type then return back. |
|  |  | * **destroydetail()**   This method takes operation detail id and delete this operation detail then return back. |
|  |  | * **showDetail()**   This method return operation details and operation types to operation details show SubTypeShow($item\_id). |
|  |  | * **Showreservations($item\_id)**   This method takes item id as a parameter and return this item to reservation show page. |
|  |  | * **Showuserreservations()**   This method takes user id from authentication and get all operations of this user then return to show reservation page. |
|  |  | * **destroyoperation($id)**   This method takes operation id as a parameter and delete this operation from the database then return back. |
|  |  | * **createType()**   This method takes operation type name from the request and add it in the data base then return back. |
|  |  | * **createDetail()**   This method takes operation detail and operation type id from the request then save it in the database then return back. |
|  |  | * **createValue($operation\_id,$type\_id,$detail\_id,$value)**   This method takes operation\_id, type\_id, detail\_id, value as a parameters and create a new value and save it in the database then return back. |

|  |  |  |
| --- | --- | --- |
| Controller Name | Description | Method & Description |
| ScheduleControlle | This controller was created for any process on Schedules. | * **checkIfScheduleExists($item\_id,$start)**   This method takes item id and start date as a parameter and check if this schedule is exists. |
|  |  | * **index($item\_id)**   This method return to item schedule page with item id. |
|  |  | * **create($item\_id)**   This method takes item id as a parameter and check if this schedule not exists then create if is exists will create error message. |
|  |  | * **ownercreate()**   This method takes item id from the request and check if this schedule not exists then create if is exists will create error message. |
|  |  | * **createWithVriables($id,$start$end$price)**   This method takes item id , start date , end date and price as a parameter and check if this schedule not exists then create if is exists will create error message. |
|  |  | * **getWholeSchedule($item\_id)**   This method takes item id and return schedule of this item. |
|  |  | * **getAvailbleTime()**   This method return all available schedules of this   * **getdays($star,$end , $schedule\_id)**   This method takes schedule id , start and end date as a parameter and return all available days between start date and end date. |
|  |  | * **cutSchedule($schedule\_id,$start,$end)**   This method takes schedule\_id,start,end as a parameter and cut the reserved dates from the schedule |
|  |  | * **showl($id)**   This method takes item id and return schedule of this item to item schedule show page. |
|  |  | * **edit()**   This method takes item id , start date , end date and price from the request and update it in the data base then return back. |
|  |  | * **destroy()**   This method takes schedule id from the request and delete it from the data base then return back |

**e**