**Chapter 5**

**ANDROID BASED PASSENGER’S MONITOR FOR**

**MA. EVENIA VAN SERVICES**

Android Based Passenger’s Monitor is an android application coordinated with a web application that makes travelling with Ma. Evenia Van Services a lot easier. The mobile app allows the passengers to monitor all of the necessary things they might want to know. The web application is accessible only to the administrators of the company. They are responsible for inputting the data that will be presented in mobile application for the users to enjoy.

The proponent’s purpose of creating this system is to ease the way people travel with Ma. Evenia. Not only does it assure security among each individual user, but it can also aid first time users, like tourist as to how the system of van travelling in Ozamiz works. The application offers the user various options like; time management with finding a ride, because users of the application can book a seat at the comforts of their own homes. The app also shows necessary information like the driver’s basic information, and the van’s plate number. Passengers no longer have to worry about getting tricked of paying too much because the app is equipped with a feature that shows how much a passenger has to pay for the trip. They can also view which route the van is taking so as passengers won’t to worry about getting lured into a trap. For a common gain to both the driver and the user, the app shows the exact drop off point where the passenger is going to take. This helps solve the issue for tourists that don’t know the local dialect and driver’s that don’t know that much of the English language. They just have to show the driver their drop off point and the problem is solved.

**System Objectives**

The study aims to:

* Give the passenger the necessary information like the van’s plate number and the driver’s personal information.
* Allow the passenger to monitor his/her current location and the current route taken by the van.
* Pin point the passengers designated drop off point.
* Present the passenger with the exact amount he/she has to pay for the trip.

**General Objectives**

In general, the android application aims to guide the passenger of his/her whereabouts to prevent any security lapses. The android application also seeks to promote honesty and modesty towards the payment and dropping off point of each passenger. With the applications map feature, the passenger can assure his/her safety towards reaching and arriving at his/her designated drop off point while also being able to know the places they’ve passed by.

**Specific Objectives**

**It specifically aims to:**

1. To secure the passenger pays the right amount
2. To track the passengers route and current location
3. Provide the passenger with the van’s and the driver’s basic information

**System Scope and Delimitation**

The study focuses only on the concern of public transportation for passengers of the Ma.Evenia van service. The study is not responsible for driver’s way of gathering more passengers (or lesser) in every trip down the route. The application does not also give off extra information beyond what the route presents neither will it give off any more information about the van and the driver than the data presented. The system is concentrated in making travelling by van easier. To achieve this, the application is built with features that help a user monitor everything while they travel. The mobile app makes for a handy way of seeing things the user’s way. For the driver’s side of the application, the app will show how many passengers he has acquired within the time limit of their parking time. The app also shows how much time the driver has left to gather as much passengers as they can. The basic information that will be given to each user is limited to only how much the driver is willing to give. But a constant thing that the system will give is a short background of the driver (years of experience, driver’s license, address, age, etc). Information about the Van is also limited to only the van’s plate number and the number of seats that van is built with. The web application administrated by a company employee is only tasked to update information about a driver or a van, and register a driver or a van. The administrator (of the company) cannot access personal information of each user registered in the system. Any other information besides the above mentioned is not covered by the system or the company.

**Physical Environment and Resources**

These are the system specifications used by the proponents in developing the research:

**Developer Side System/Application**

**Hardware**

**Platforms to be used:**

-RAM : 4.00 GB

-Hard Disk Drive : 1000 GB

-Processor : Intel CoreI5 1.70 GHz

-Laptop

-Mouse

**Software**

-OS : Windows 7 and versions after.

-Platform : Android Studio, Google Maps API, Firebase

-Language : HTML, PHP, JAVASCRIPT, JAVA, MYSQL

**Deployment Specification**

**Recommended system requirements**

**Mobile Phone:**

**-**Mobile Phone OS : Android

-Version : API 9: Android 4 and versions after

-RAM : 512MB or higher

**Desktop Computer:**

-Computer OS : Windows 7 versions after

-Processor : Intel CoreI3 or later

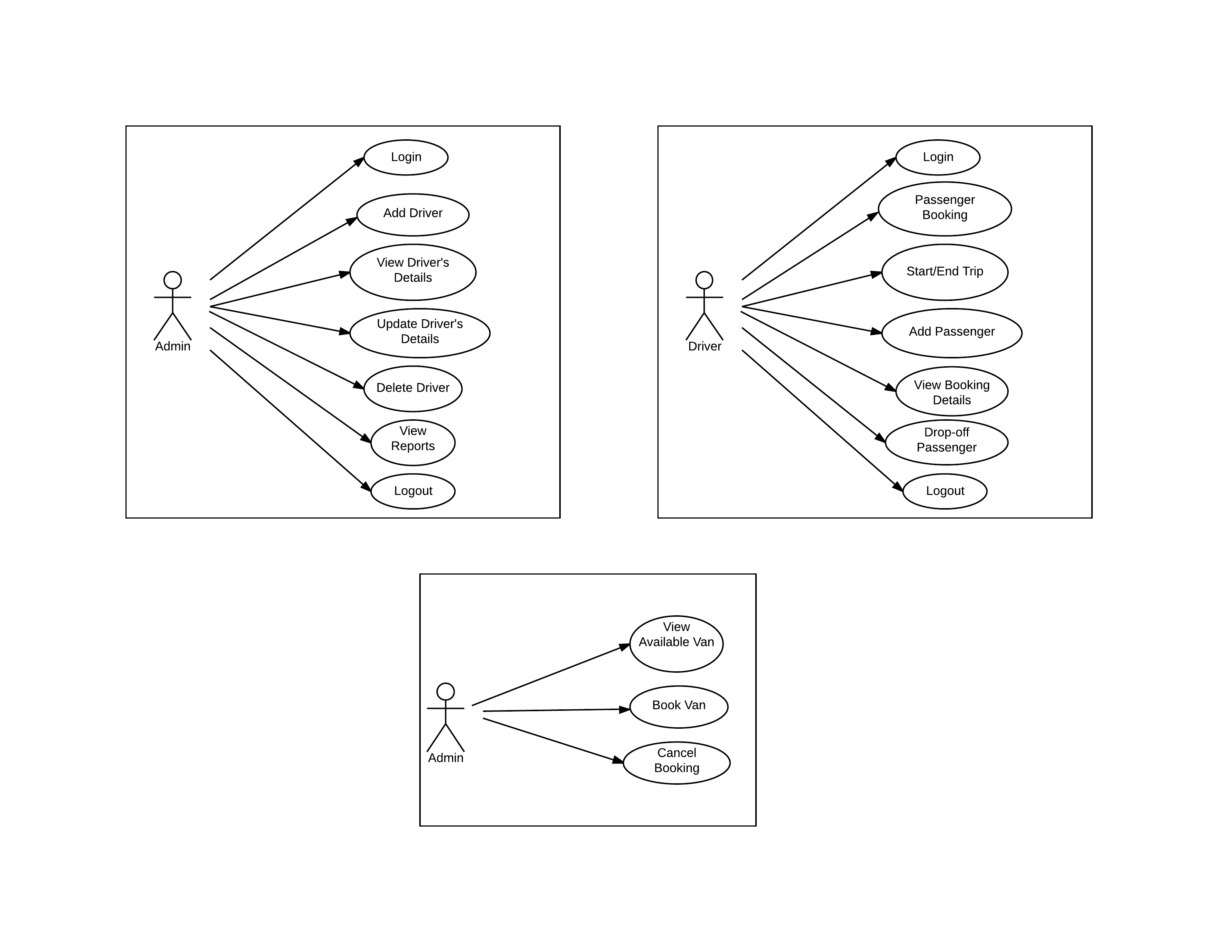
-RAM : Minimum of 2GB

-Graphics Card : Minimum of 2GB

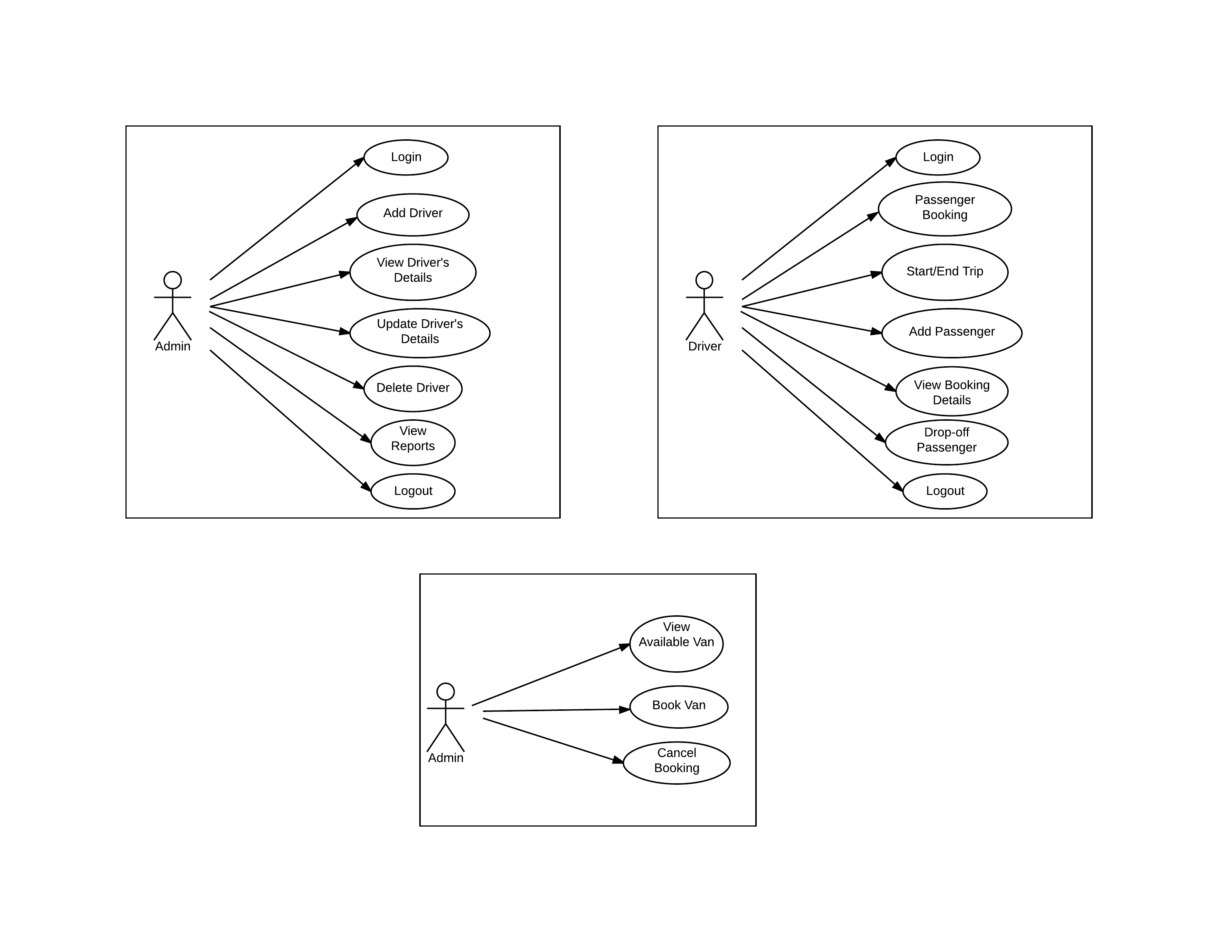
**Peopleware**

Assigned employees of the company are the only ones (besides the developers) that can modify changes in the web application for the system.

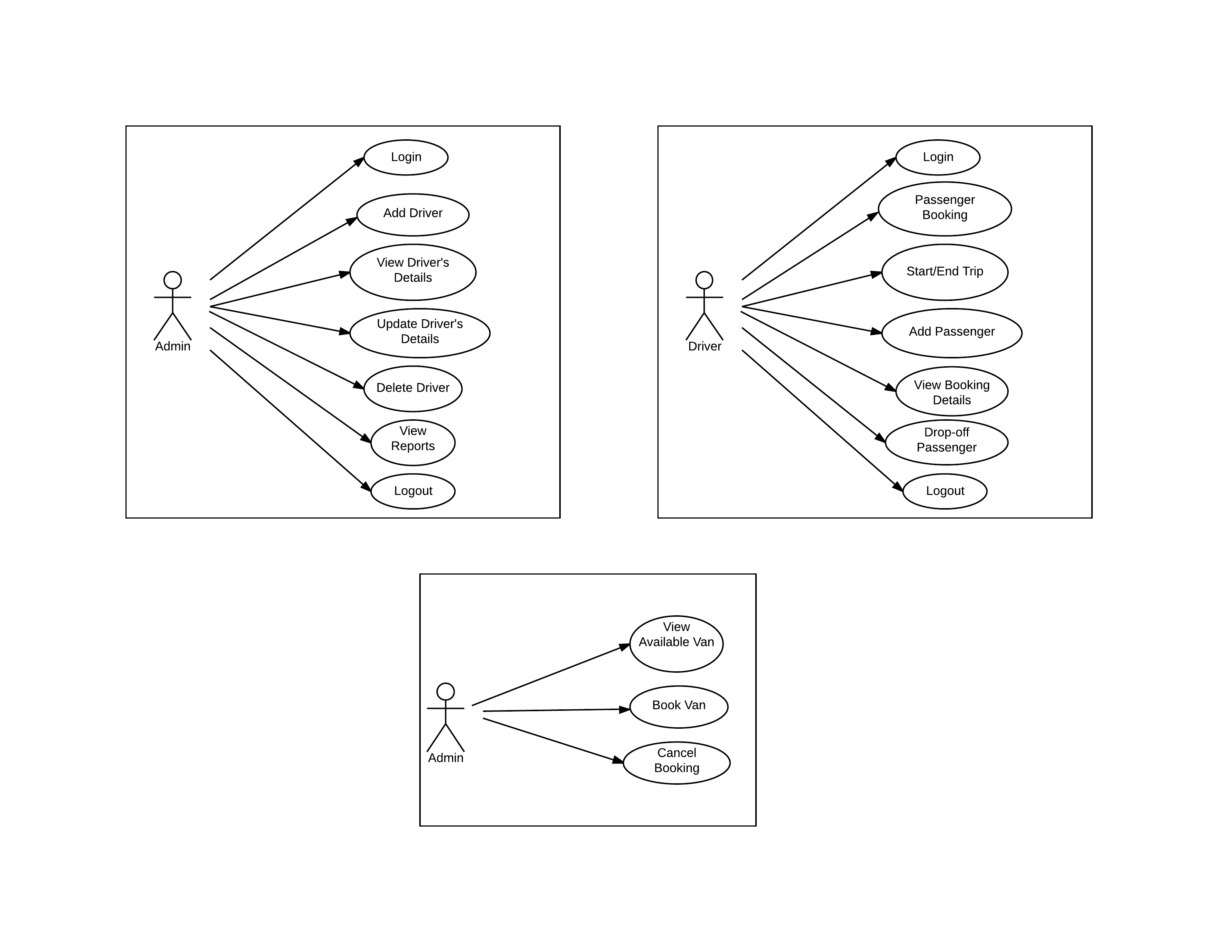
**Use Case Modelling**

****

**Figure 5.1: Admin Use Case Diagram**

****

**Figure 5.2: Driver Use Case Diagram**

****

**Figure 5.3: Passenger Use Case Diagram**

**Actor List**

* **Admin –** refers to the van service owner who manages the operations of Maria Evenia Van Services. He/she is responsible for the applications nevessary information.
* **Driver –** refers to the person who drives the Maria Evenia vans and makes use of the application wherein he/she inputs the details of the passengers.
* **Passenger –** refers to the person who uses Maria Evenia vans as their mode of transportation and views the availability of van in the area where he/she can ride on/

**Admin Use Case List**

**UC1 Login:** The admin must login his/her account before he/she can access the website which shows the transactions

**UC2 Add Driver:** The admin can add a driver including its corresponding details such as basic information, the assigned van and the specific route

**UC3 View Driver’s Details:** The admin can view the details of the driver on the landing page

**UC4 Update Driver’s Details:** When there are changes when it comes to the driver’s van or his/her routes, the admin can change the specific van and route of which the driver was originally assigned to

**UC5 Delete Driver:** The admin can delete existing driver if he/she no longer works for the company

**UC6 View Reports:** The admin can view the reports

**UC7 Logout:** To secure some privacy, the admin can log out from his/her account respectively

**Admin Use Case Scenarios**

**Table 5.1**

Login

|  |  |
| --- | --- |
| UC1:Login | |
| Precondition: None | |
| System User | System Response |
| 1. The admin enters the username and password | 2. The system checks if the information entered is correct |
| Post-condition: Welcome Page appears | |

**Table 5.2**

Add Driver

|  |  |
| --- | --- |
| UC2:Add Driver | |
| Precondition: Login | |
| System User | System Response |
| 1. The admin entered the information needed on each fields | 2. The system checks if the entered information existed already in the database  3. The system will alert “Successfully added. Please click the button to proceed.” |
| Post-condition: Welcome Page | |

**Table 5.3**

View Driver’s Details

|  |  |
| --- | --- |
| UC3:View Driver’s Details | |
| Precondition: Login | |
| System User | System Response |
| 1. The admin clicks a driver to view its information details | 2. The system retrieves the information from the database |
| Post-condition: Update Driver’s Details, Delete Driver or back to Welcome Page | |

**Table 5.4**

Update Driver’s Details

|  |  |
| --- | --- |
| UC4:Update Driver’s Details | |
| Precondition: View Driver’s Details | |
| System User | System Response |
| 1. The admin clicks the update driver button   3. The admin enters the new information | 2. The system loads a screen where the existing details is already seen in the fields  4. The system stores the new information to the database.  5. The system will alert “Successfully updated. Please click the button to proceed.” |
| Post-condition: Welcome Page | |

**Table 5.5**

Delete Driver

|  |  |
| --- | --- |
| UC5:Delete Driver | |
| Precondition: View Driver’s Details | |
| System User | System Response |
| 1. The admin clicks the delete button | 2. The system deletes the driver from the database  3. The system will alert “Successfully deleted. Please click the button to proceed.” |
| Post-condition: Welcome Page | |

**Table 5.6**

View Reports

|  |  |
| --- | --- |
| UC6:View Reports | |
| Precondition: Login | |
| System User | System Response |
| 1. The admin clicks the view report button | 2. The system retrieves information from the database and displays it |
| Post-condition: Admin can view the reports | |

**Table 5.7**

Logout

|  |  |
| --- | --- |
| UC7:Logout | |
| Precondition: Login | |
| System User | System Response |
| 1. The user clicks “Logout” button if he/she wishes to logout | 2. The system logs the user out |
| Post-condition: Login Page | |

**Driver Use Case List**

**UC8 Login:** The driver can login to its account given by the admin

**UC9 Passenger Booking:** The driver can see if a passenger would want to ride his van which includes the drop-off point

**UC10 Start/End Trip:** The driver clicks start or end trip to signify when he/she starts his shift

**UC11 Add Passenger:** The driver can add a passenger by clicking the add button and supply the necessary details needed including their drop-off point and faire

**UC12 View Booking Details:** The driver can view the details of the passenger’s booking

**UC13 Drop-off Passenger:** The driver can swipe left a passenger if he/she is in already in their respective drop-off point

**UC14 Logout:** The driver can logout from their account respectively to preserve their accounts privacy

**Driver Use Case Scenarios**

**Table 5.8**

Login

|  |  |
| --- | --- |
| UC8:Login | |
| Precondition: None | |
| System User | System Response |
| 1. The user signs in through the input field provided | 2. The system will validate the information and checks if the information is correct and it matches. |
| Post-condition: The user will be directed to the driver’s home page where the driver can start his/her shift. | |

**Table 5.9**

Passenger Booking

|  |  |
| --- | --- |
| UC9:Passenger Booking | |
| Precondition: Start/End Trip | |
| System User | System Response |
| 4. The user accepts or declines the passenger who wants to ride the van | 1. The system receives a notification that a passenger wants to ride on the van  2. The system checks if there is still availability  3. The system notifies the user that a passenger wants to ride the van |
| Post-condition: | |

**Table 5.10**

Start/End Trip

|  |  |
| --- | --- |
| UC10:Start/End Trip | |
| Precondition: Login | |
| System User | System Response |
| 1. The user will click the start trip to notify that the current van he/she is using is available | 2. The system will store the information to the database  3. The button then changes from start trip to end trip whenever the user ends his/her voyage |
| Post-condition: The user will be allowed to see the list of passengers on his/her trip. | |

**Table 5.11**

Add Passenger

|  |  |
| --- | --- |
| UC11:Add Passenger | |
| Precondition: Start/End trip | |
| System User | System Response |
| 1. The user clicks the add passenger and its specific details | 2. The system will validate the information being inputted and will prompt the user that the passenger has been added.  3. The system will notify the user if the passenger is near his/her drop-off point. |
| Post-condition: The user is able to view the booking details or drop the passenger off when he/she is near the drop off point. | |

**Table 5.12**

View Booking Details

|  |  |
| --- | --- |
| UC12:View Booking Details | |
| Precondition: Add Passenger | |
| System User | System Response |
| 1. The user clicks the passenger of which he/she wants to view | 2. The system will retrieve the specific information needed and displays it on the screen. |
| Post-condition: The user will be able to swipe the specific passenger to drop him/her off or stay at the same page | |

**Table 5.13**

Drop-off Passenger

|  |  |
| --- | --- |
| UC13:Drop-off Passenger | |
| Precondition: Start/End trip | |
| System User | System Response |
| 2. The user swipes the passenger to the left to signify that the passenger is already out of the bus | 1. The system notifies the user if the passenger is near his drop-off point.  3. The system saves the information of the passenger’s booking to the database.  4. The system will notify the user that the information is stored. |
| Post-condition: The user will be led back to the passenger list of he/she may end trip | |

**Table 5.14**

Logout

|  |  |
| --- | --- |
| UC14:Logout | |
| Precondition: Login | |
| System User | System Response |
| 1. The user clicks “Logout” button if he/she wishes to logout | 2. The system logs the user out |
| Post-condition: Login scene | |

**Passenger Use Case List**

**UC15 View Available Van:** The passenger can view vans that are available for them to ride on

**UC16 Book Van:** The passenger can click an icon to notify the driver of his/her bookings

**UC17 Cancel Booking:** The passenger can cancel his/her bookings

**Passenger Use Case Scenario**

**Table 5.15**

View Available Van

|  |  |
| --- | --- |
| UC15:View Available Van | |
| Precondition: None | |
| System User | System Response |
| 1. The passenger opens the application | 2. The system loads the vans that are available at the moment |
| Post-condition: Ride Van | |

**Table 5.16**

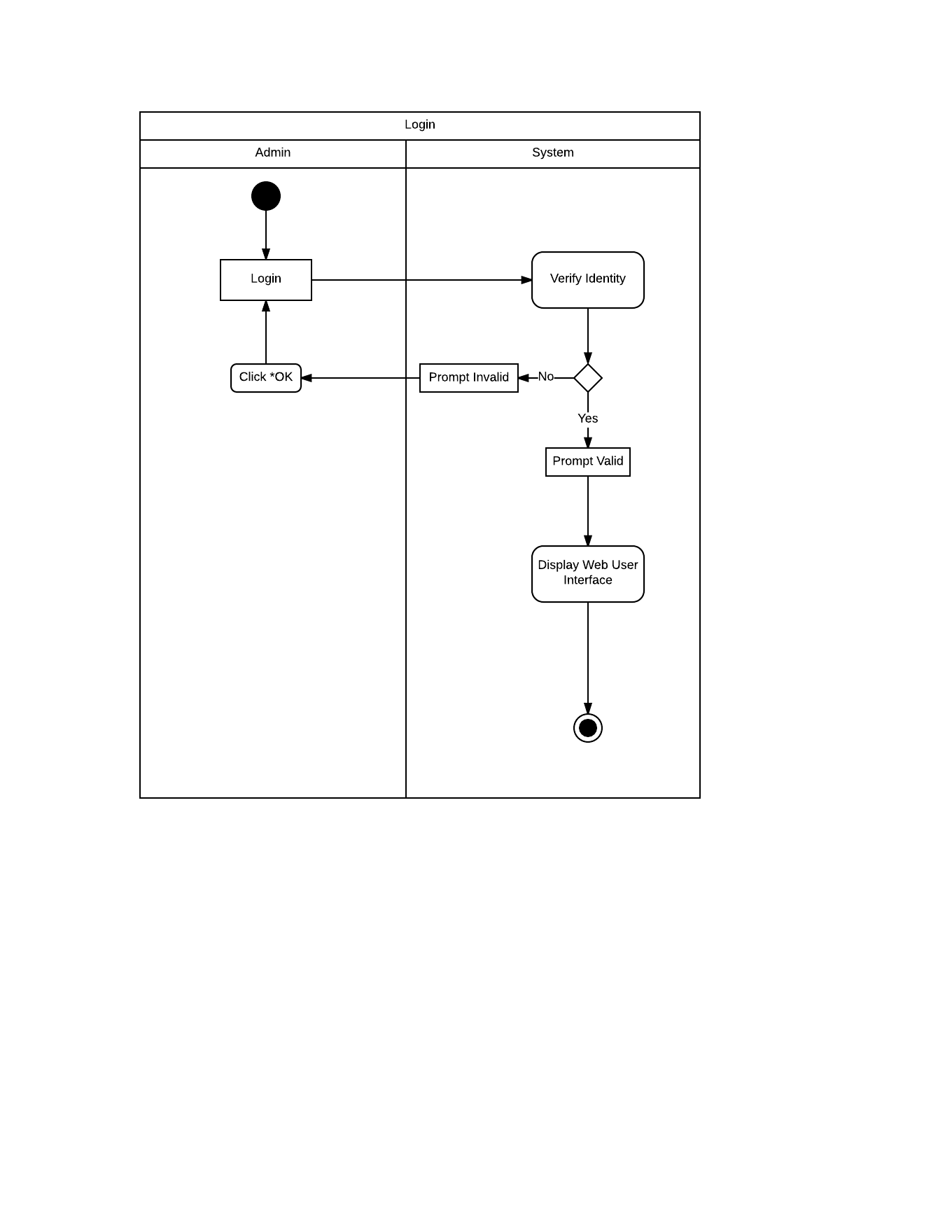
Ride Van

|  |  |
| --- | --- |
| UC16:Ride Van | |
| Precondition: View Available Van | |
| System User | System Response |
| 1. The user clicks ride van | 2. The system will notify the driver from the van that a passenger wants to ride his van  3. The system loads the booking details from the and displays it |
| Post-condition: Home page | |

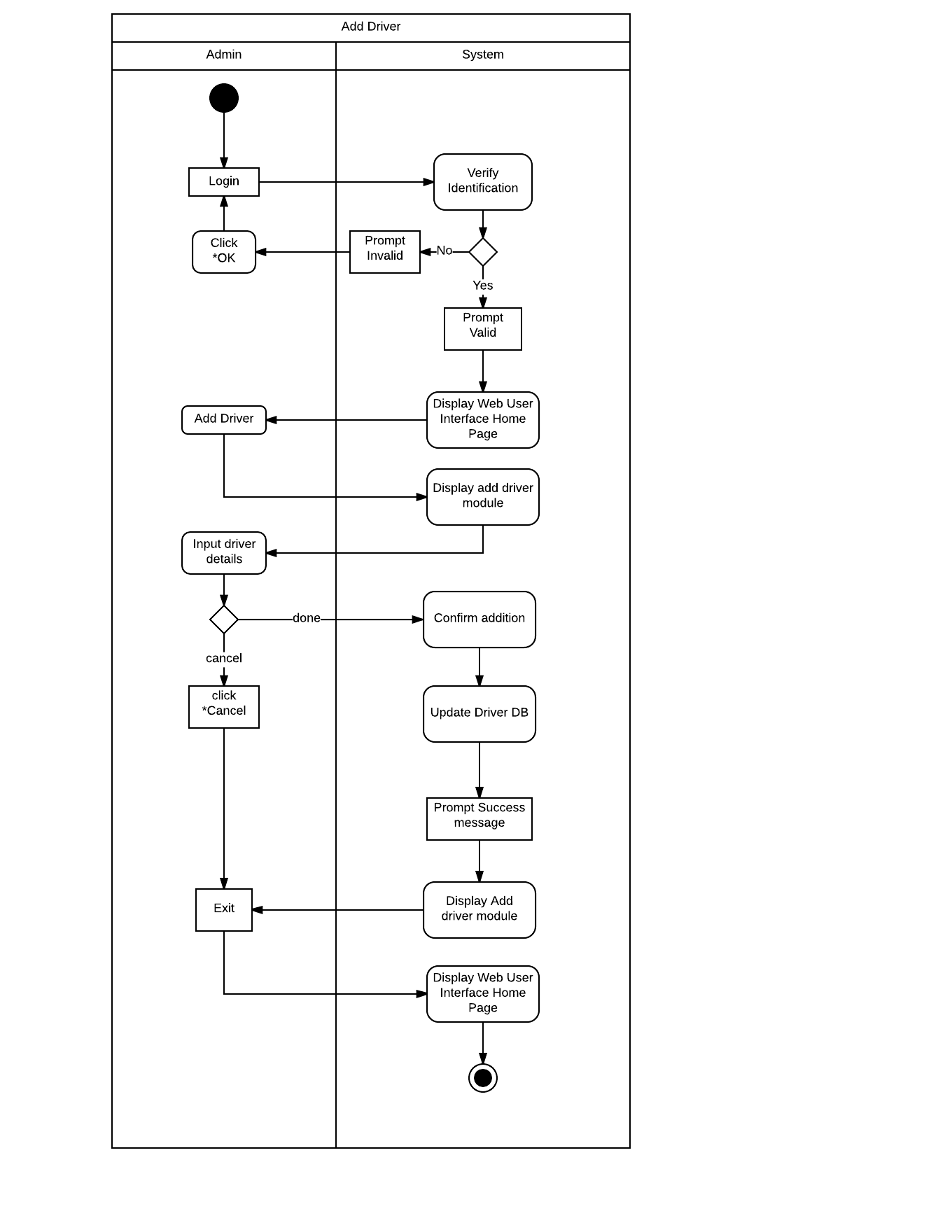
**Table 5.17**

Cancel Booking

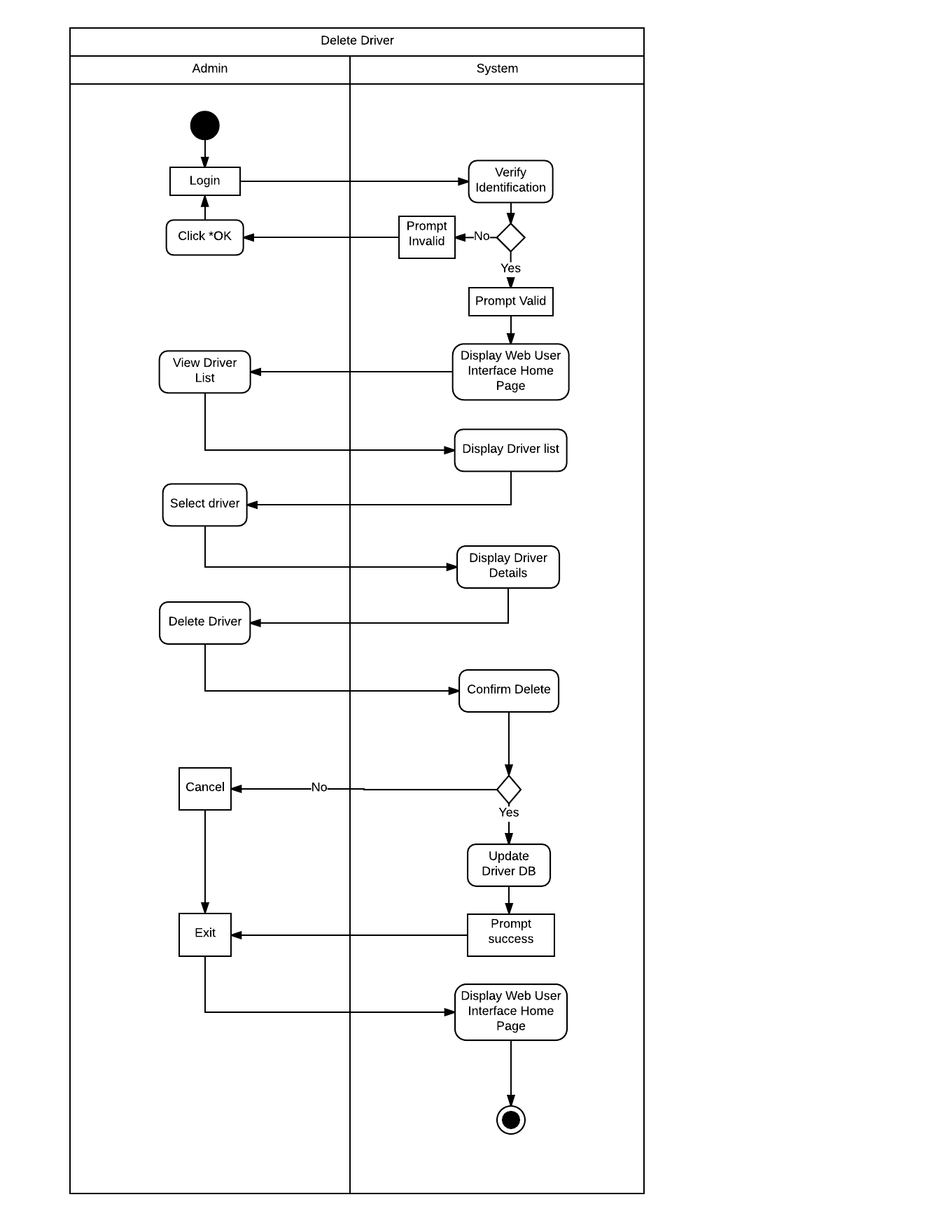
|  |  |
| --- | --- |
| UC17: Cancel Booking | |
| Precondition: Ride Van | |
| System User | System Response |
| 1. The user will click “Cancel Booking” if he/she wants to cancel his/her ride | 2. The system will notify the driver from that van that a passenger will cancel his booking |
| Post-condition: View Available Van | |

**Activity Diagram**

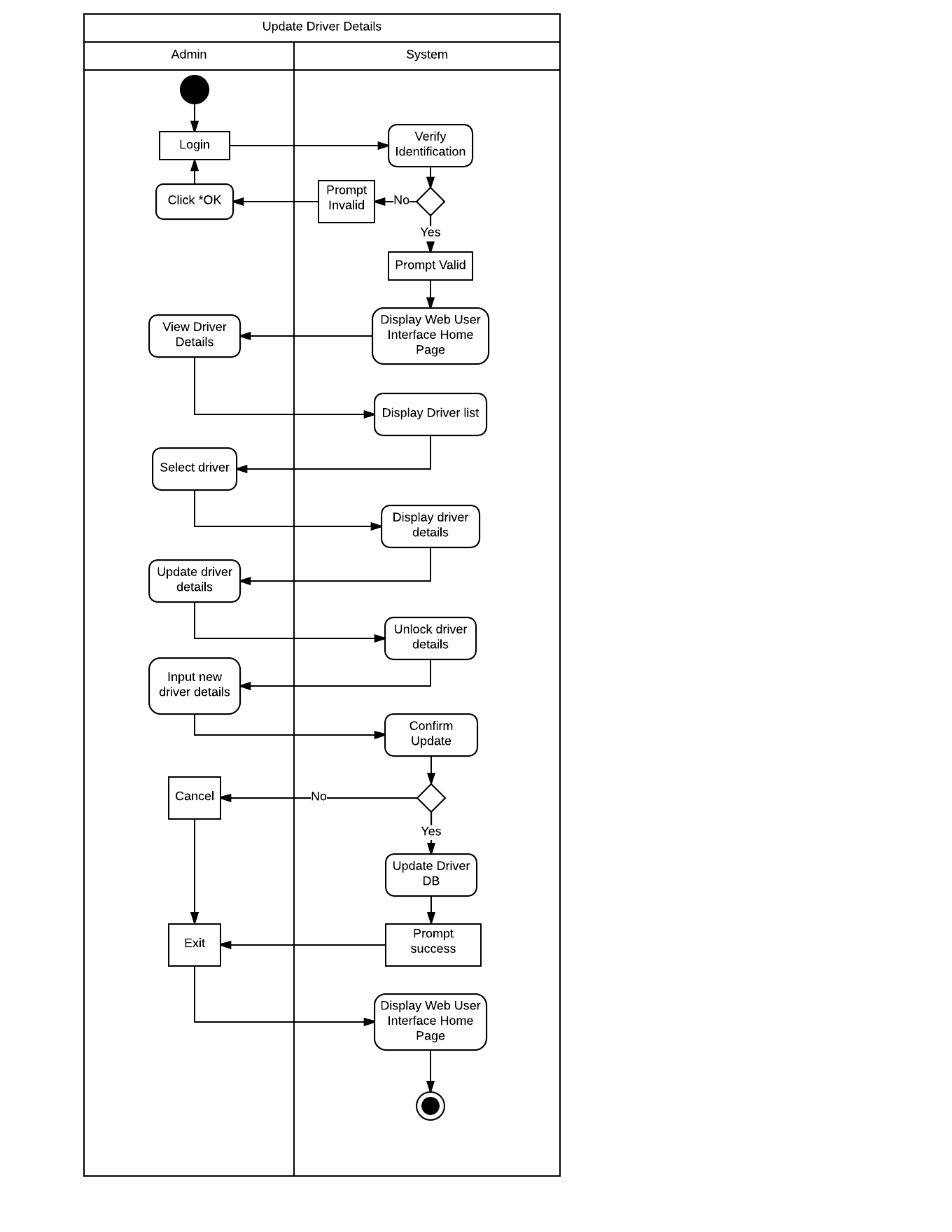
**Figure 5.4 Admin Login Activity Diagram**



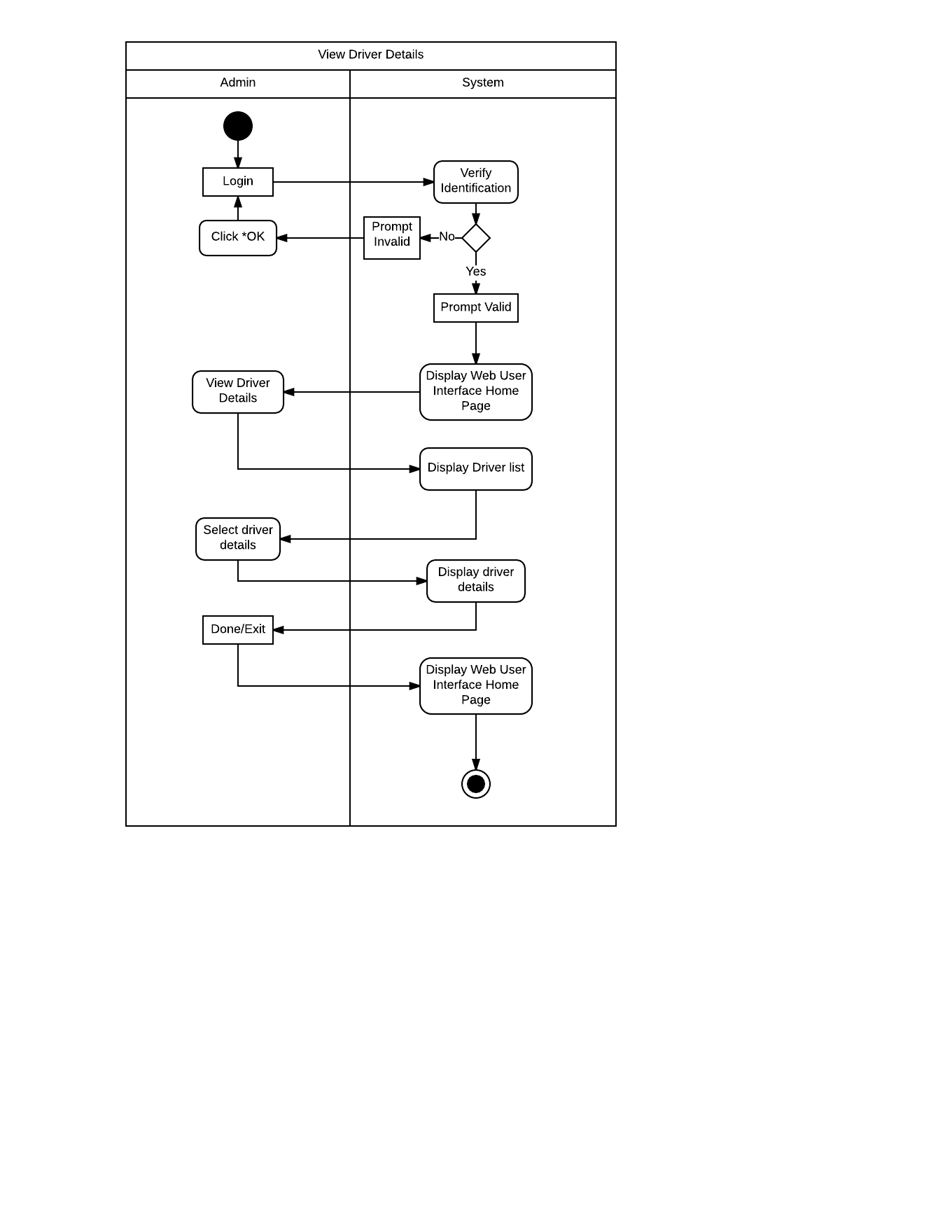
**Figure 5.4 Admin Add Driver Activity Diagram**

****

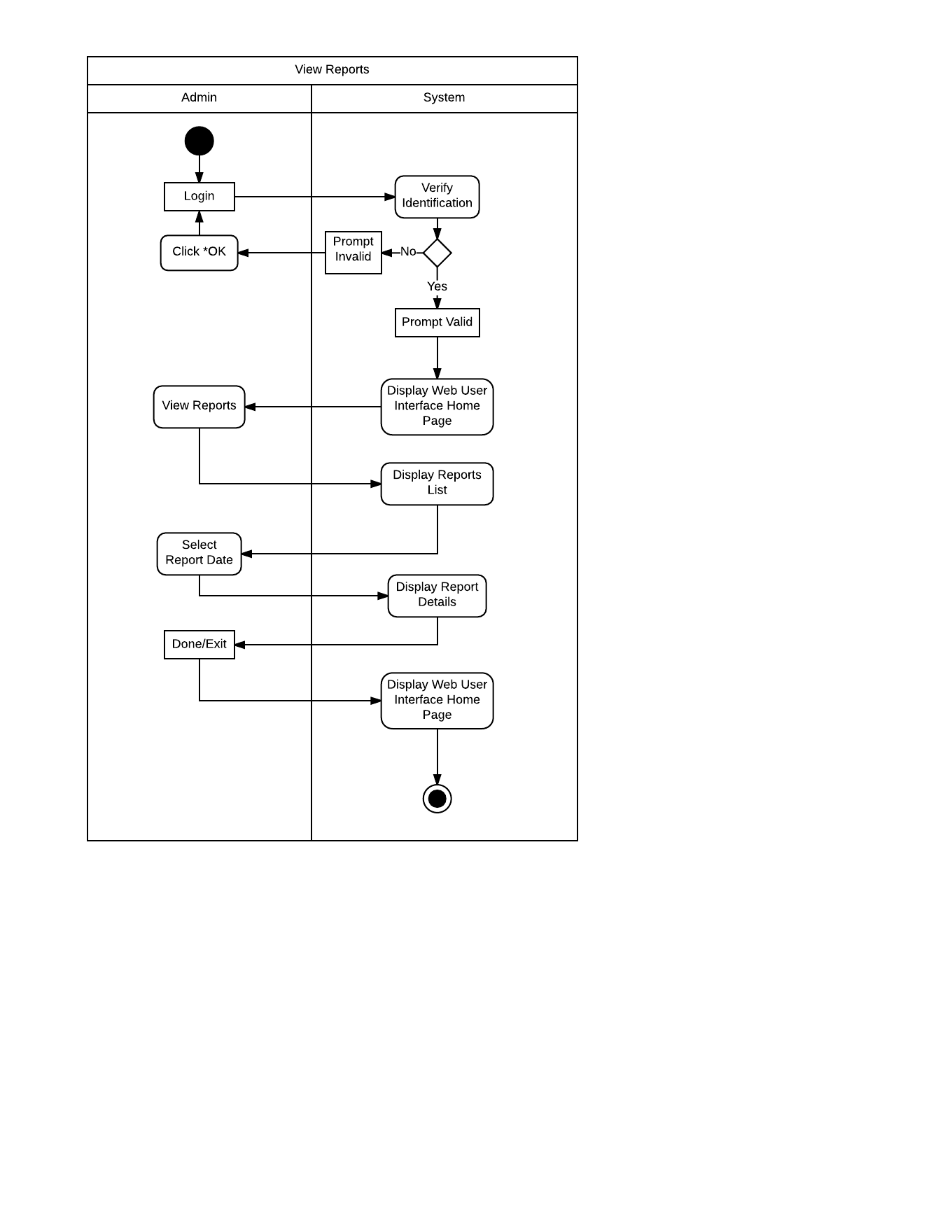
**Figure 5.5 Admin Delete Driver Activity Diagram**

****

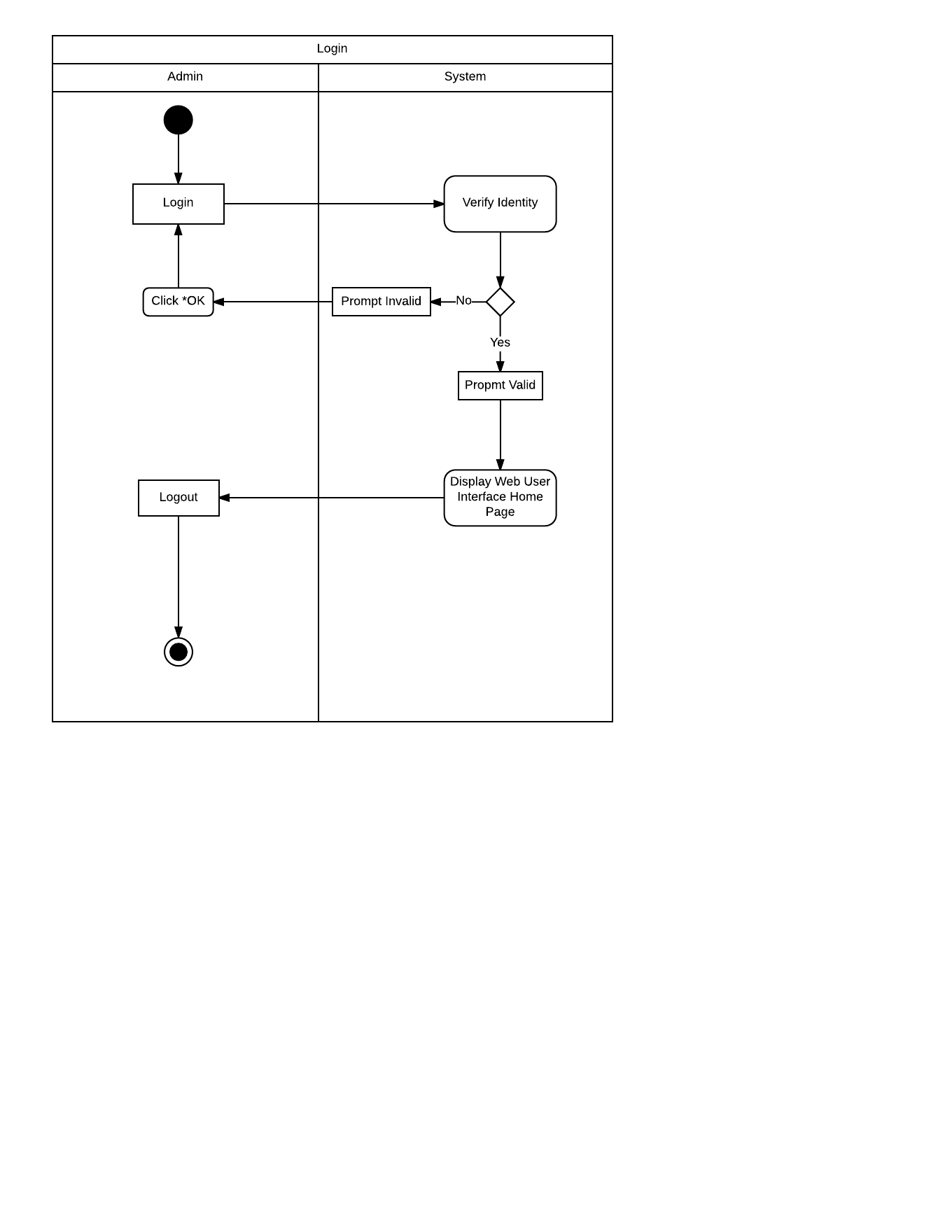
**Figure 5.6 Admin Update Driver Details Activity Diagram**

****

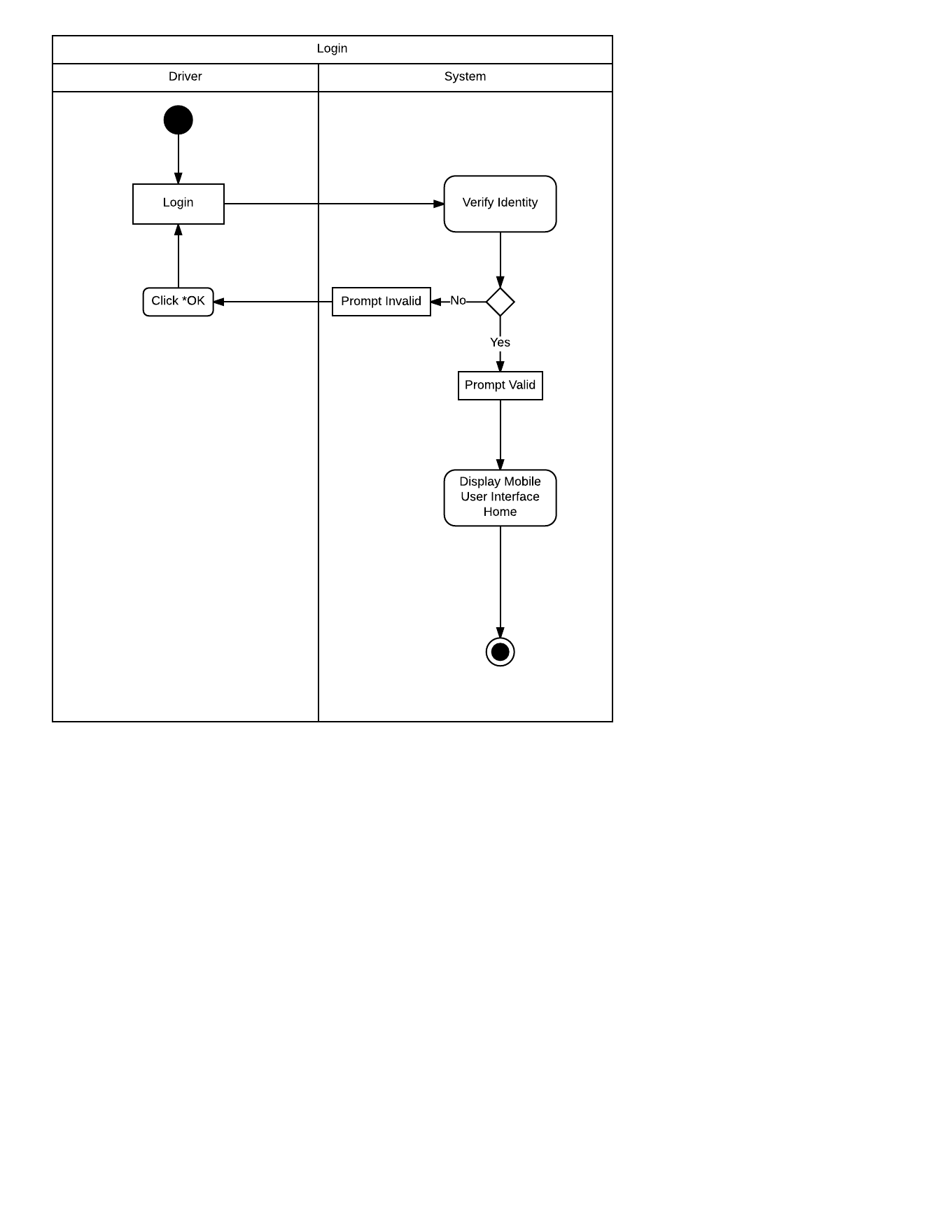
**Figure 5.7 Admin View Driver Details Activity Diagram**

****

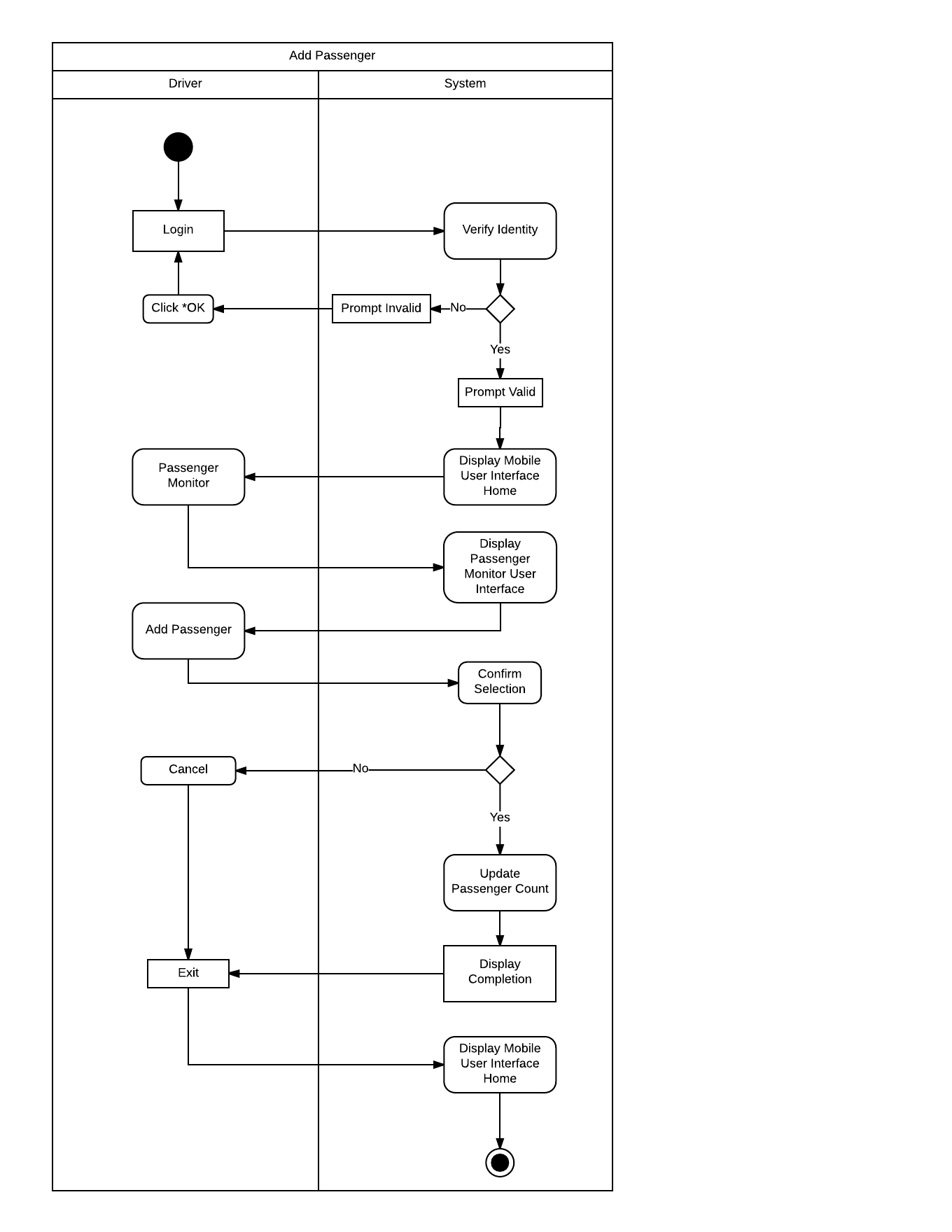
**Figure 5.8 Admin View Report Activity Diagram**

****

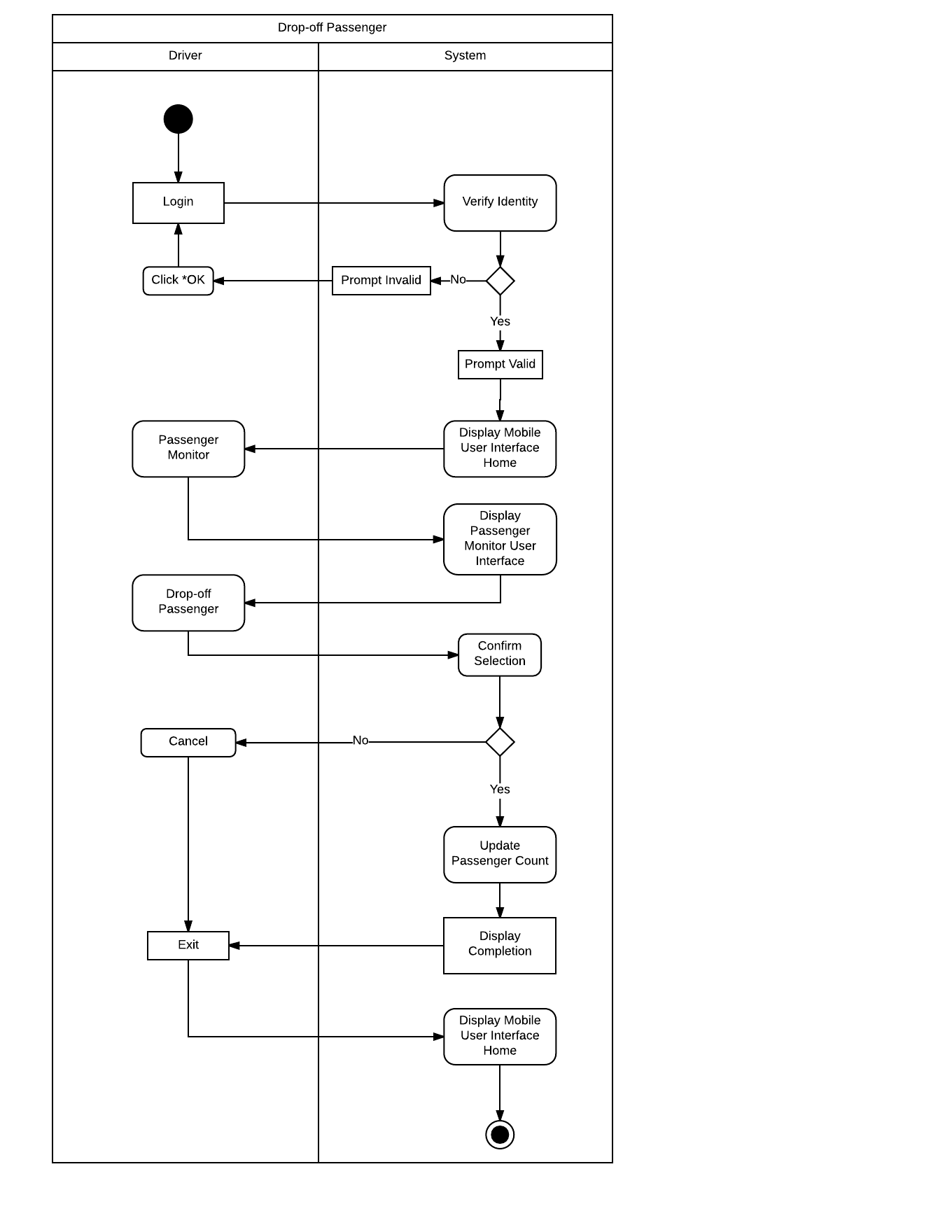
**Figure 5.9 Admin Logout Activity Diagram**

****

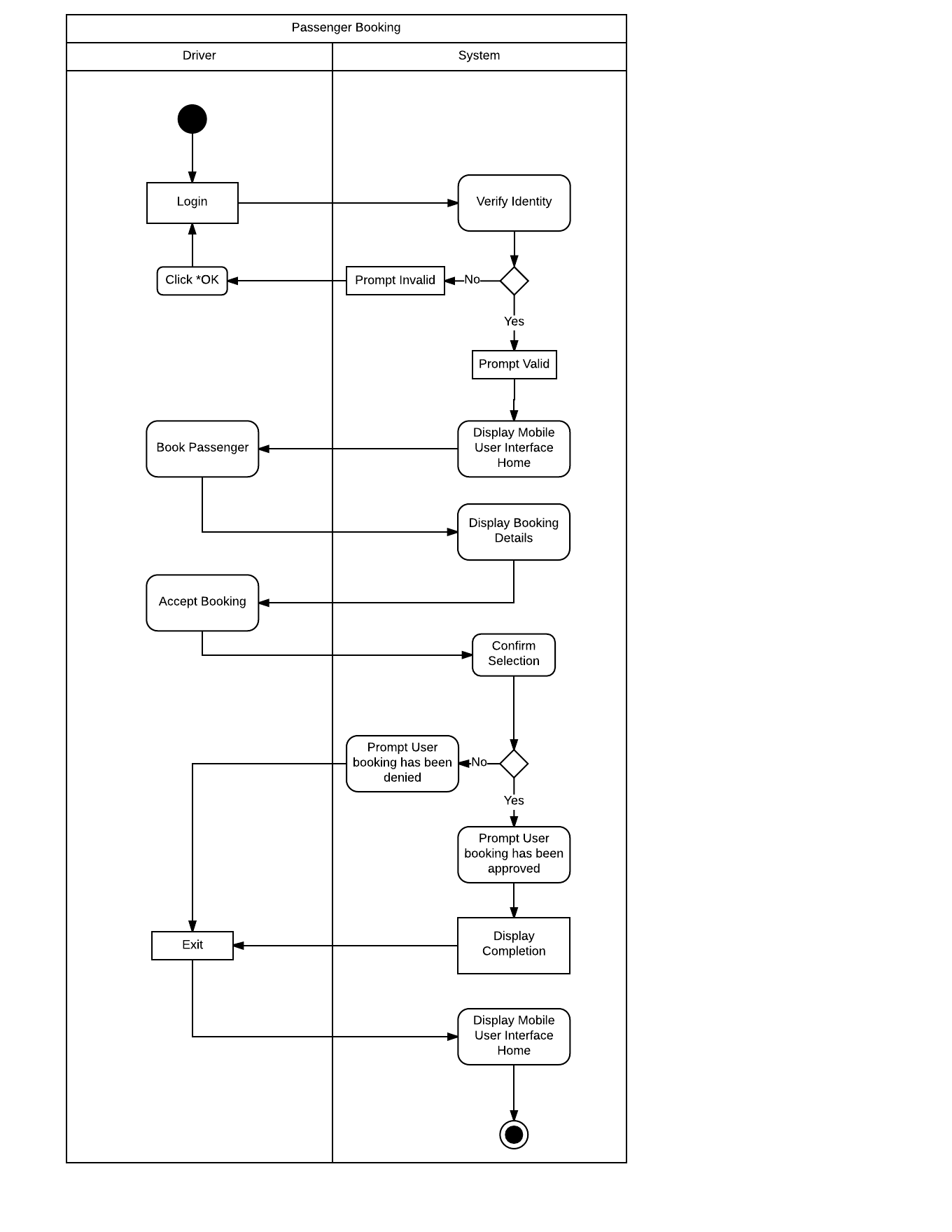
**Figure 5.10 Driver Login Activity Diagram**

****

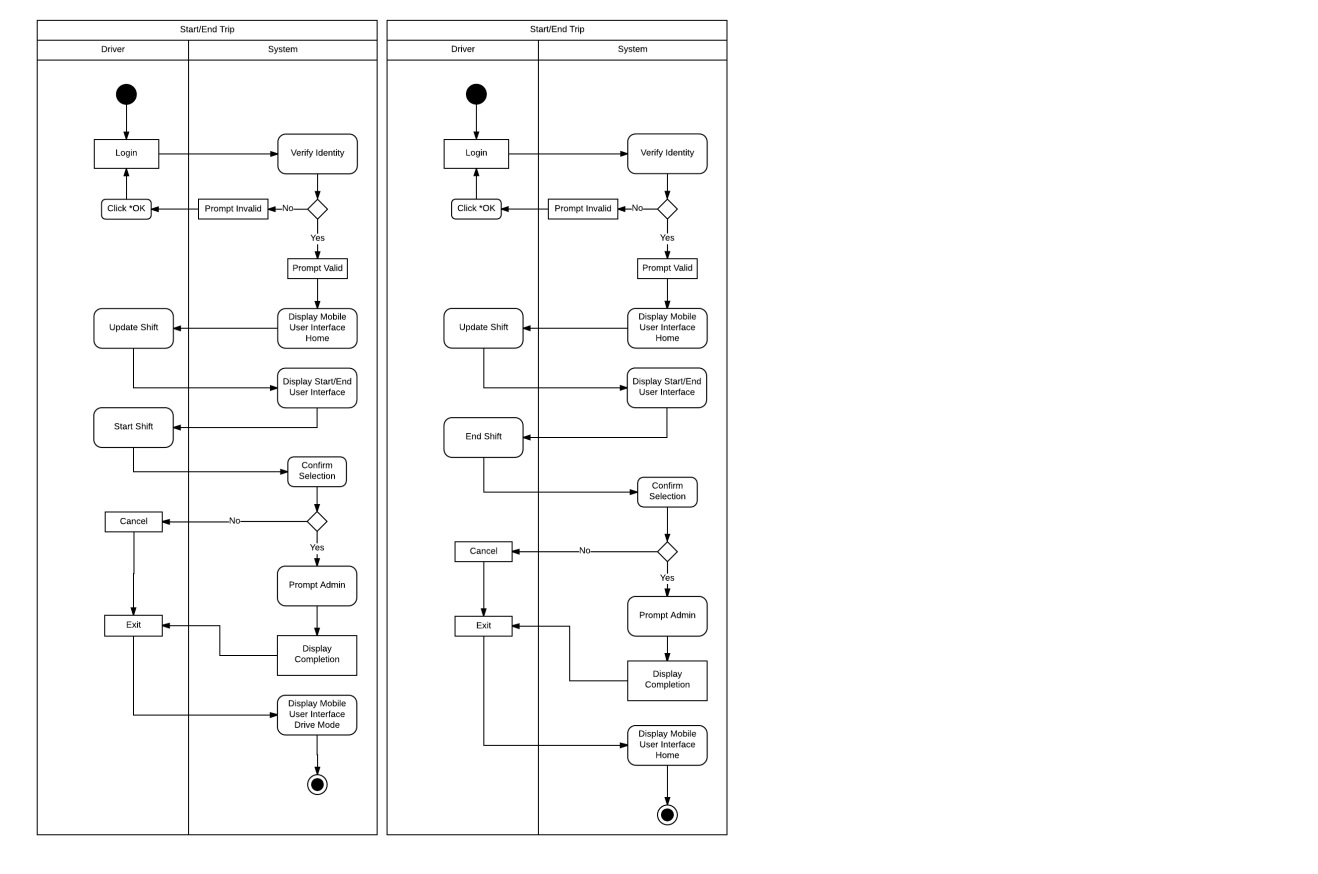
**Figure 5.11 Driver Add Passenger Activity Diagram**

****

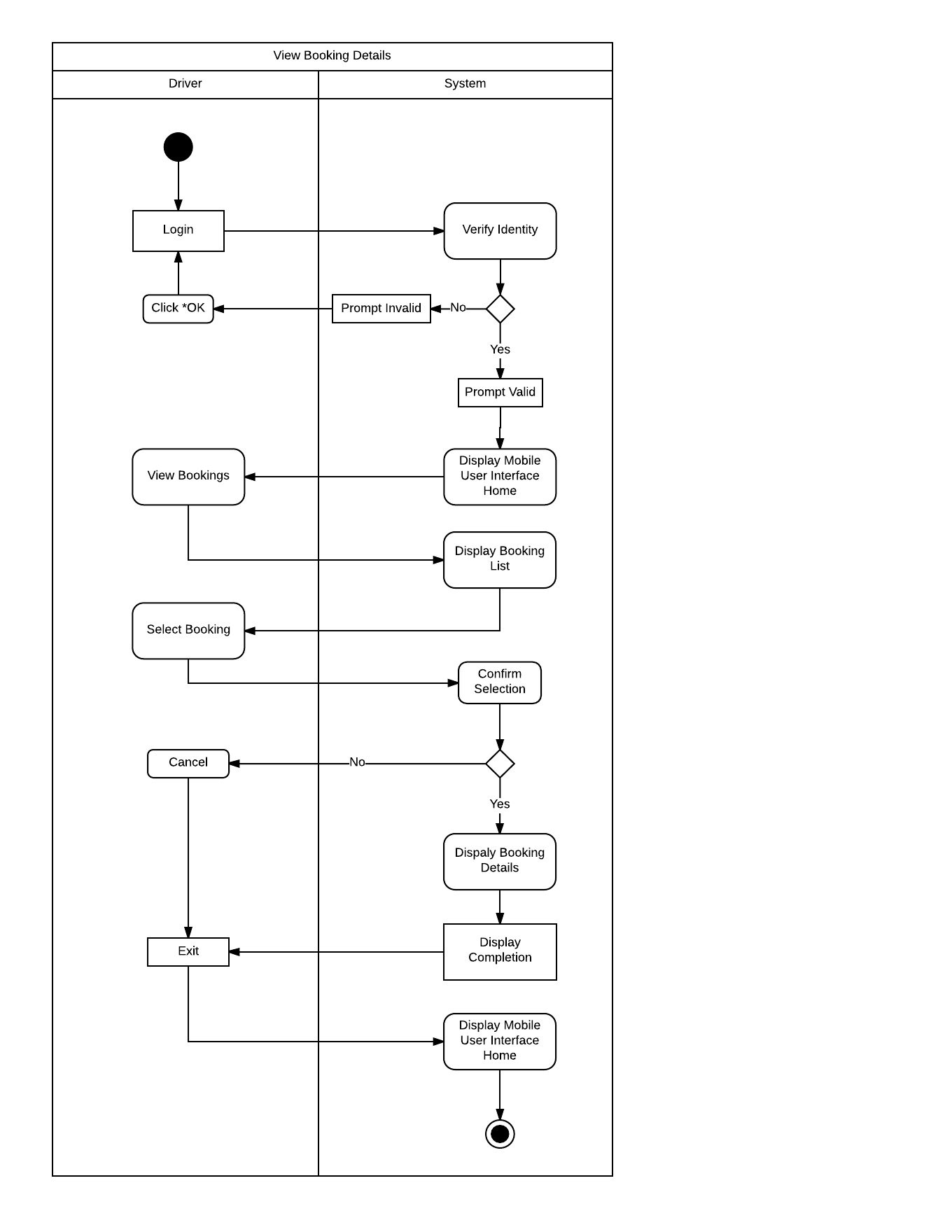
**Figure 5.11 Driver Drop Off Passenger Activity Diagram**

****

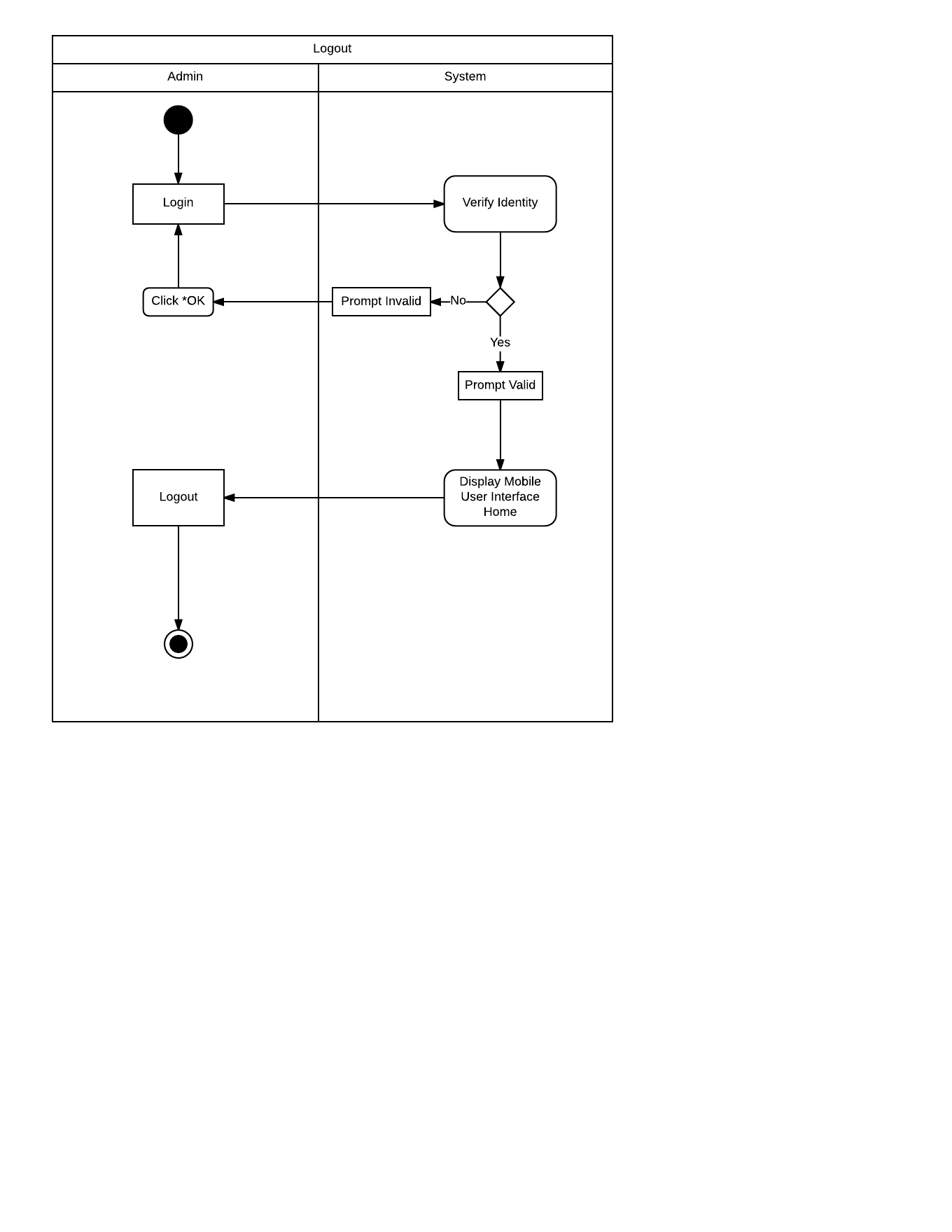
**Figure 5.12 Driver Passenger Booking Activity Diagram**

****

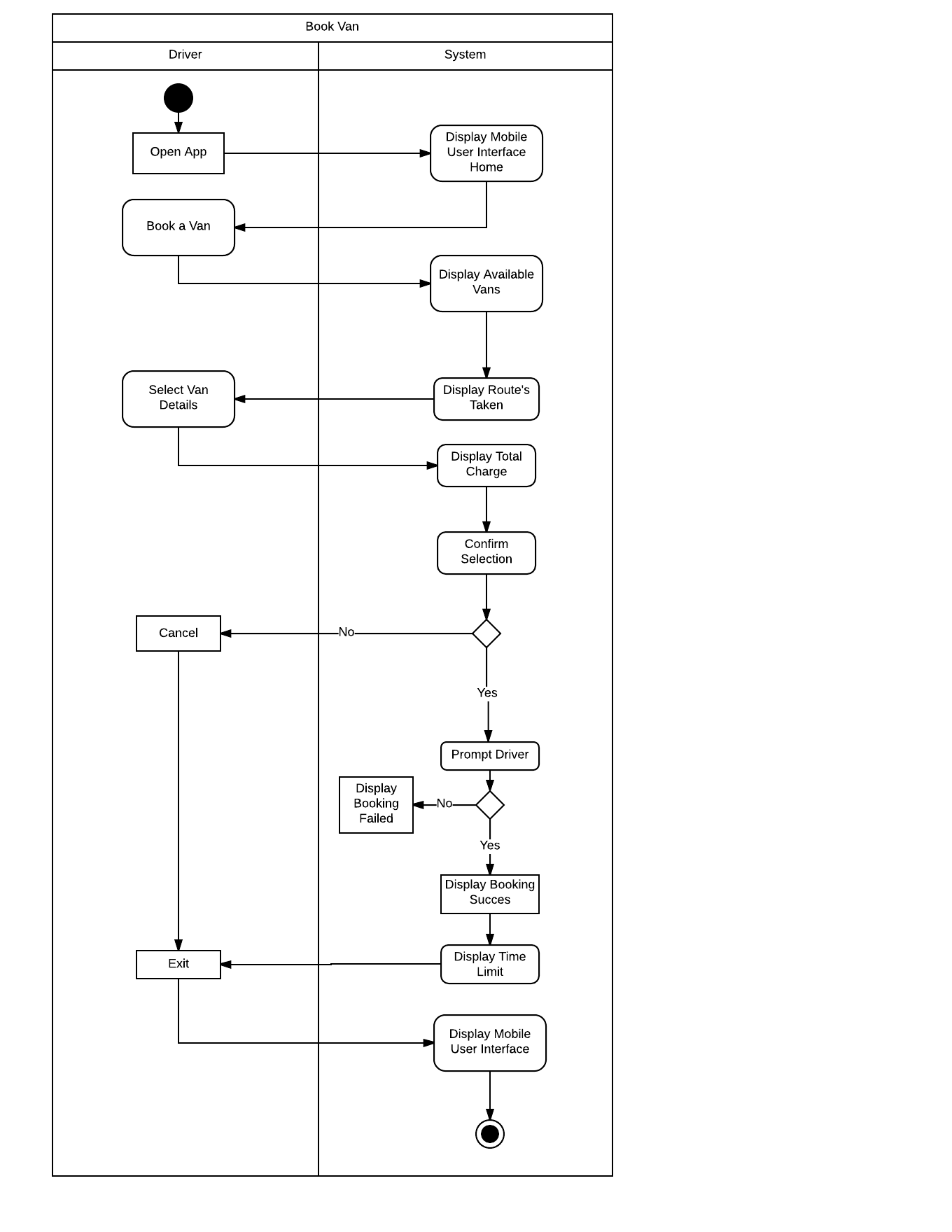
**Figure 5.13 Driver Start and End Trip Activity Diagram**

****

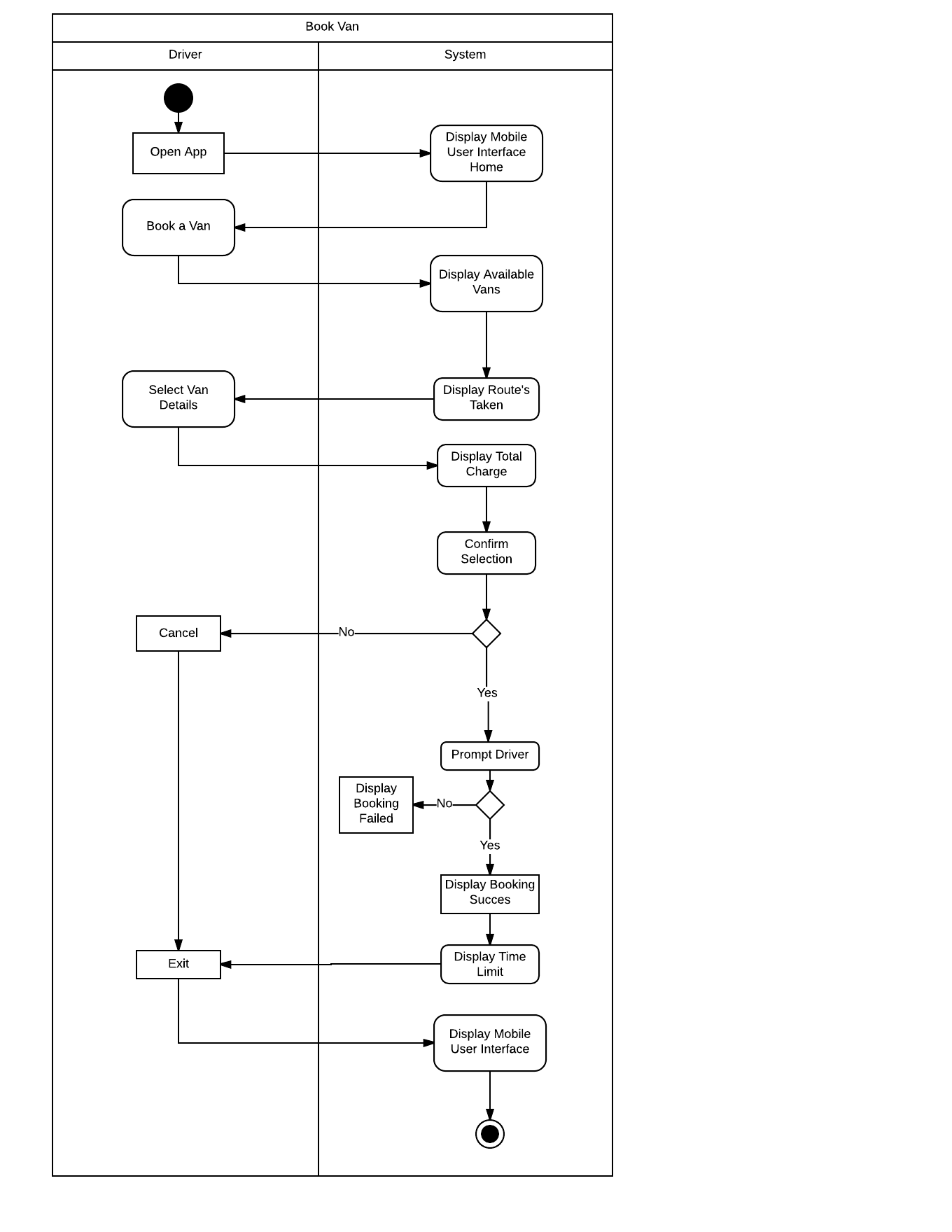
**Figure 5.14 Driver Booking Details Activity Diagram**

****

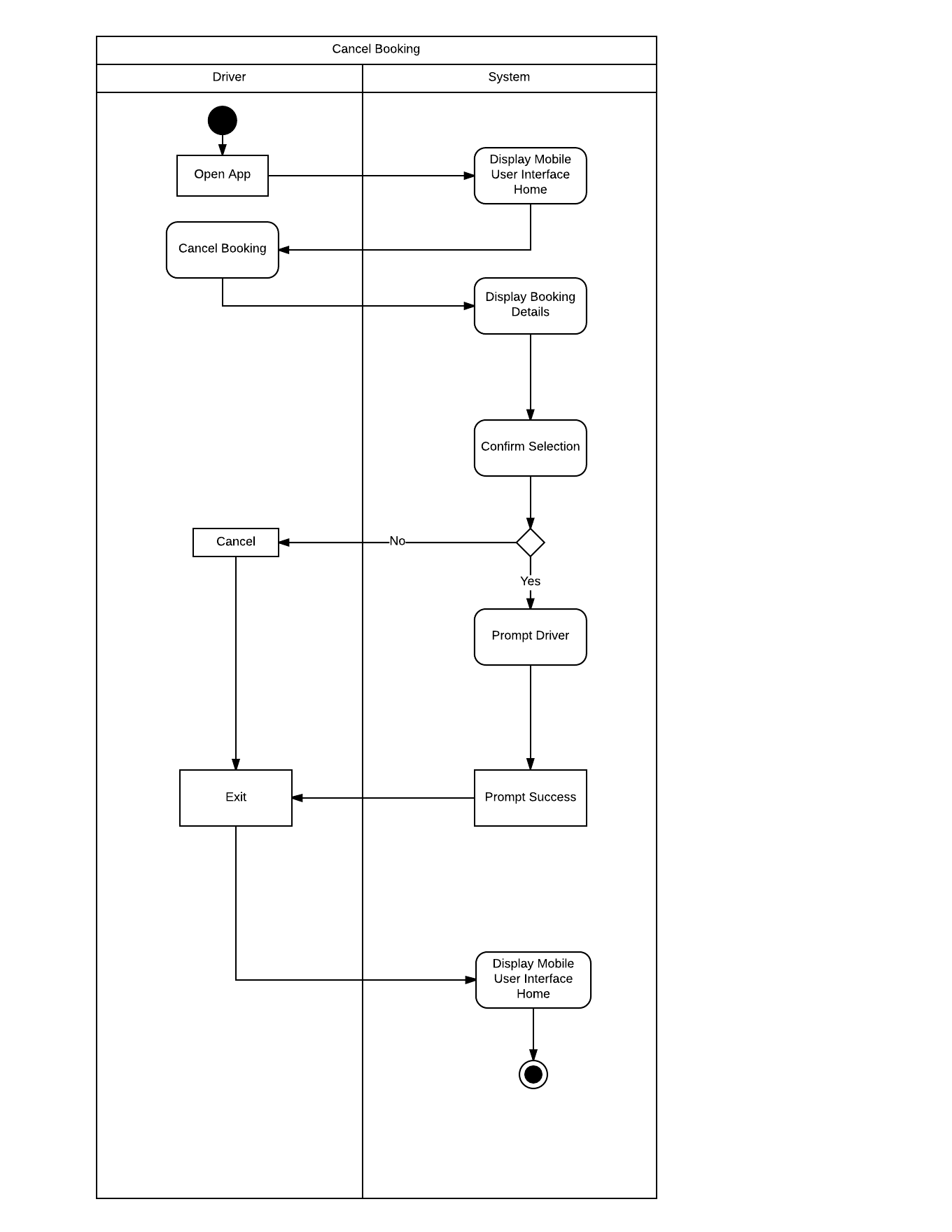
**Figure 5.15 Driver Logout Activity Diagram**

****

**Figure 5.16 Passenger View Available Van Activity Diagram**

****

**Figure 5.17 Passenger Book Van Activity Diagram**

****

**Figure 5.18 Passenger Cancel Booking Activity Diagram**