**AUTOMOBILE PRUDENT SYSTEM**

**SEQUENCE MODELLING**

**GOKUL.S 2018103026**

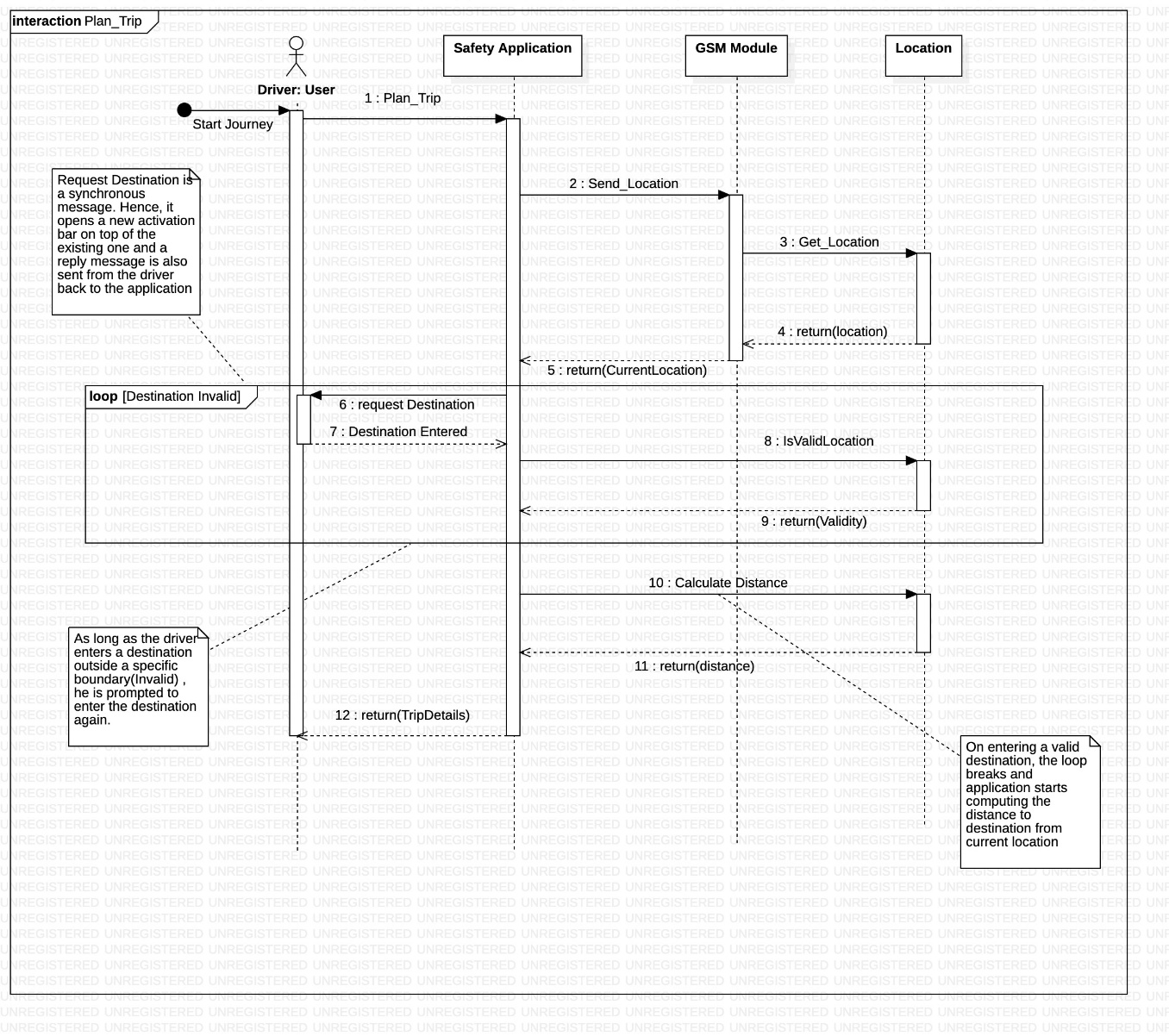
**SRIHARI.S 2018103601**

SD1 **Associated Usecase** – Plan\_Trip

**Collaborating classes** - User, Safety Application, GSM Module, Location

**Steps:** User invokes the Plan\_Trip method of safety application class

* Application finds the current location of the vehicle by invoking the send location method of GSM Module class.
* GSM Module in-turn retrieves the location by contacting the location class.
* Current Location is returned back to the safety application class
* User is prompted to enter the required destination location
* On entering a valid location, safety application class calculates the distance and returns the trip details to the user.

If the entered destination location is invalid, user is prompted to enter a valid destination until he gets one right.

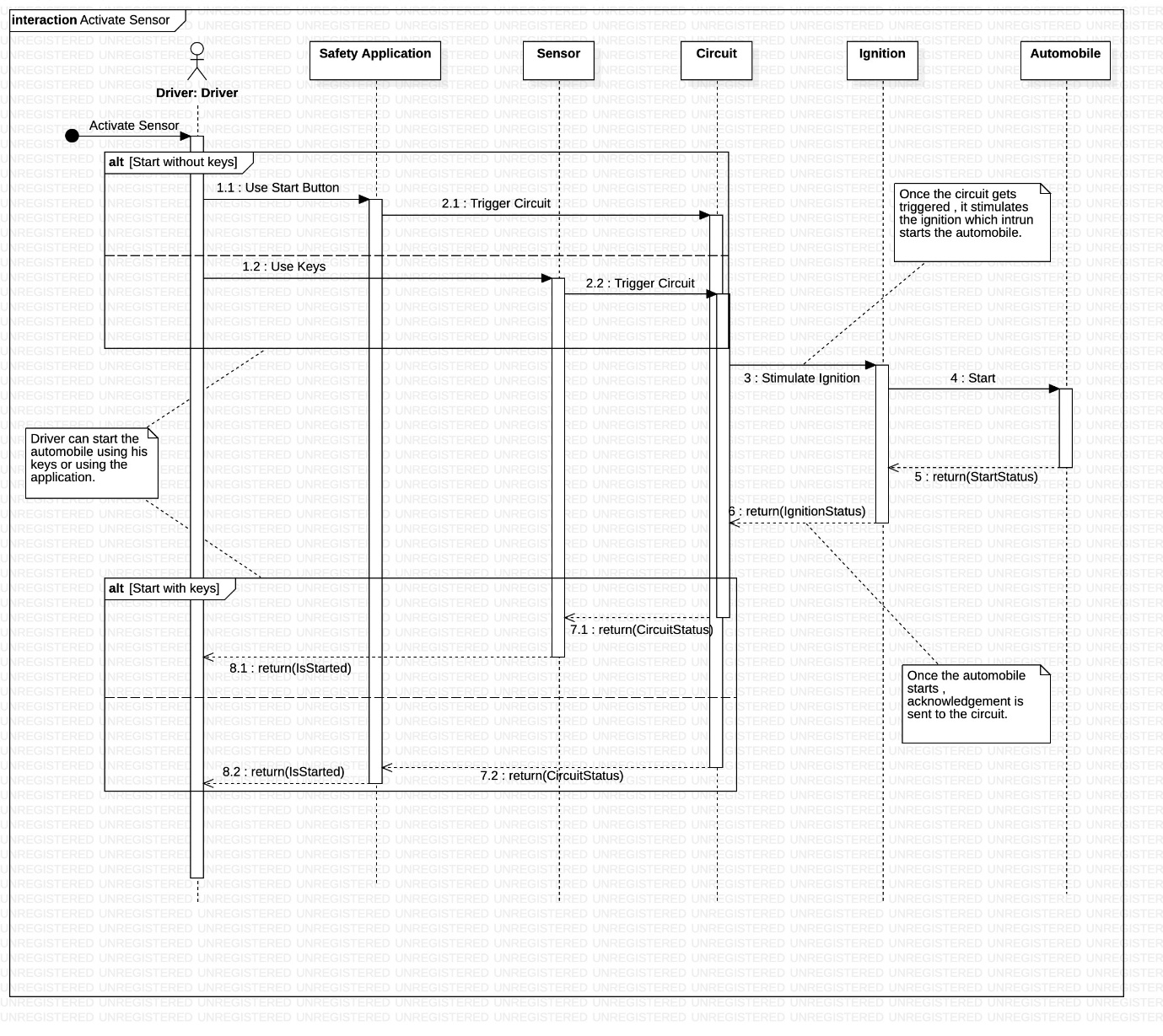
SD2

**Associated Usecase** – Activate Sensor

**Collaborating classes** - Driver, Safety Application, Sensor, Circuit, Ignition, Automobile

**Steps:**

* Driver has two options to start his automobile, i.e. With keys / Using the application.
* If he uses the application to start it directly triggers the circuit
* If he uses his keys to start it activates the sensor which in-turn triggers the circuit
* This is determined by the use of an alt frame.
* Once the circuit gets triggered, it stimulates ignition which in-turn starts the automobile.
* Since each message is synchronous, we’re obliged to return a reply message.
* If the driver had started with application, he gets the return status from the safety application class, which in-turn got an acknowledge from the circuit class.
* If the driver had started using his keys, he gets the return status from the sensor class, which in-turn got an acknowledge from the circuit class.
* This is determined by the use of an alt frame.



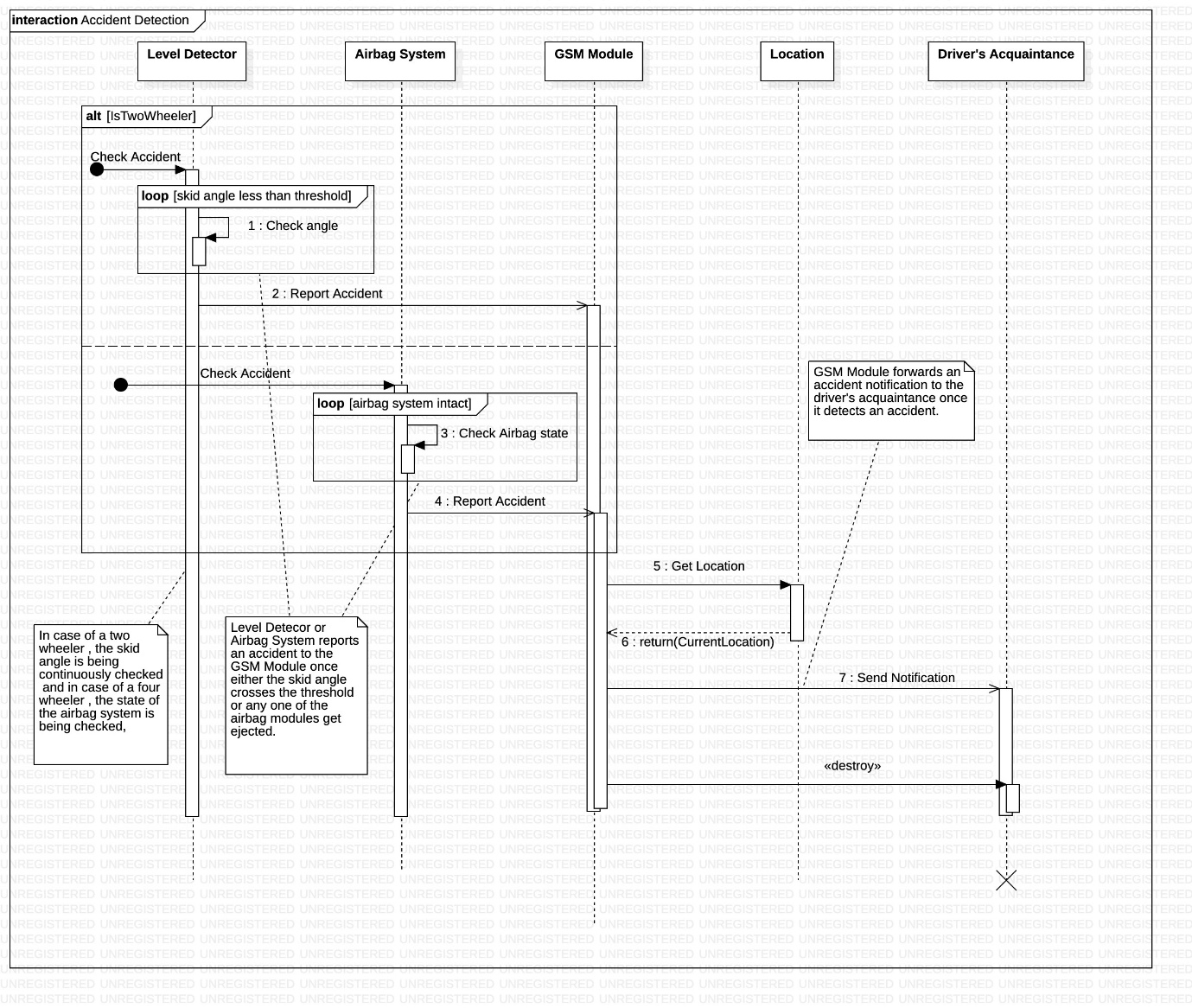
SD3

**Associated Usecase** – Encounter Accident, Detect Location

**Collaborating classes** - Level Detector, Airbag System, GSM Module, Location, Driver’s Acquaintance

**Steps:**

* If the driver uses a two wheeler, the skid angle is continuously checked.
* If it crosses the threshold, the loop breaks (loop frame) and the level detector class reports an accident to the GSM Module class.
* If the driver uses a four wheeler, the state of his airbag system is continuously checked.
* If the airbag gets ejected at any point, the loop breaks (loop frame) and the airbag system class reports an accident to the GSM Module class.
* This is determined using an alt frame.
* GSM Module contacts the location class to retrieve the current location of the driver.
* It then sends a notification to the Driver’s Acquaintance class, that contains the location.

Finally, the GSM Module destroys the already created object of Driver’s Acquaintance class.

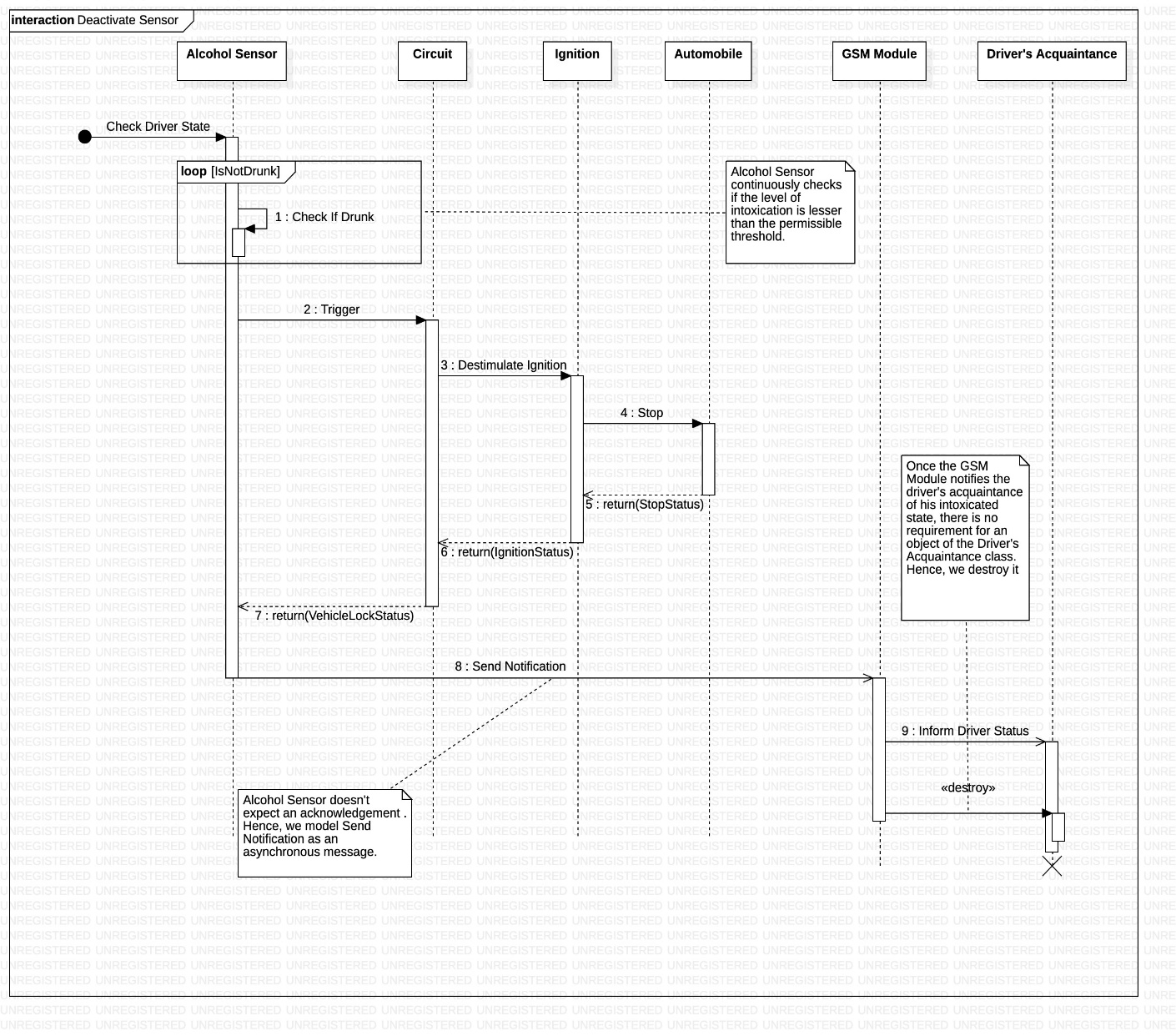
SD4

**Associated Usecase** – Deactivate Sensor, Enable alcohol sensor

**Collaborating classes** – Alcohol sensor,Circuit, Ignition, Automobile, GSM Module, Driver’s Acquaintance

**Steps:**

* The level of intoxication of the driver is continuously checked.
* Loop breaks (loop frame) once the alcohol sensor detects that the driver is drunk.
* It triggers the circuit, which in-turn de-stimulates the ignition.
* As a result, the automobile comes to a halt.
* Since the messages sent till now are synchronous, corresponding reply messages are being returned.
* The alcohol sensor class sends a notification to the GSM Module class which in-turn informs the driver’s acquaintance about his state.

Finally, the GSM Module destroys the already created object of Driver’s Acquaintance class.

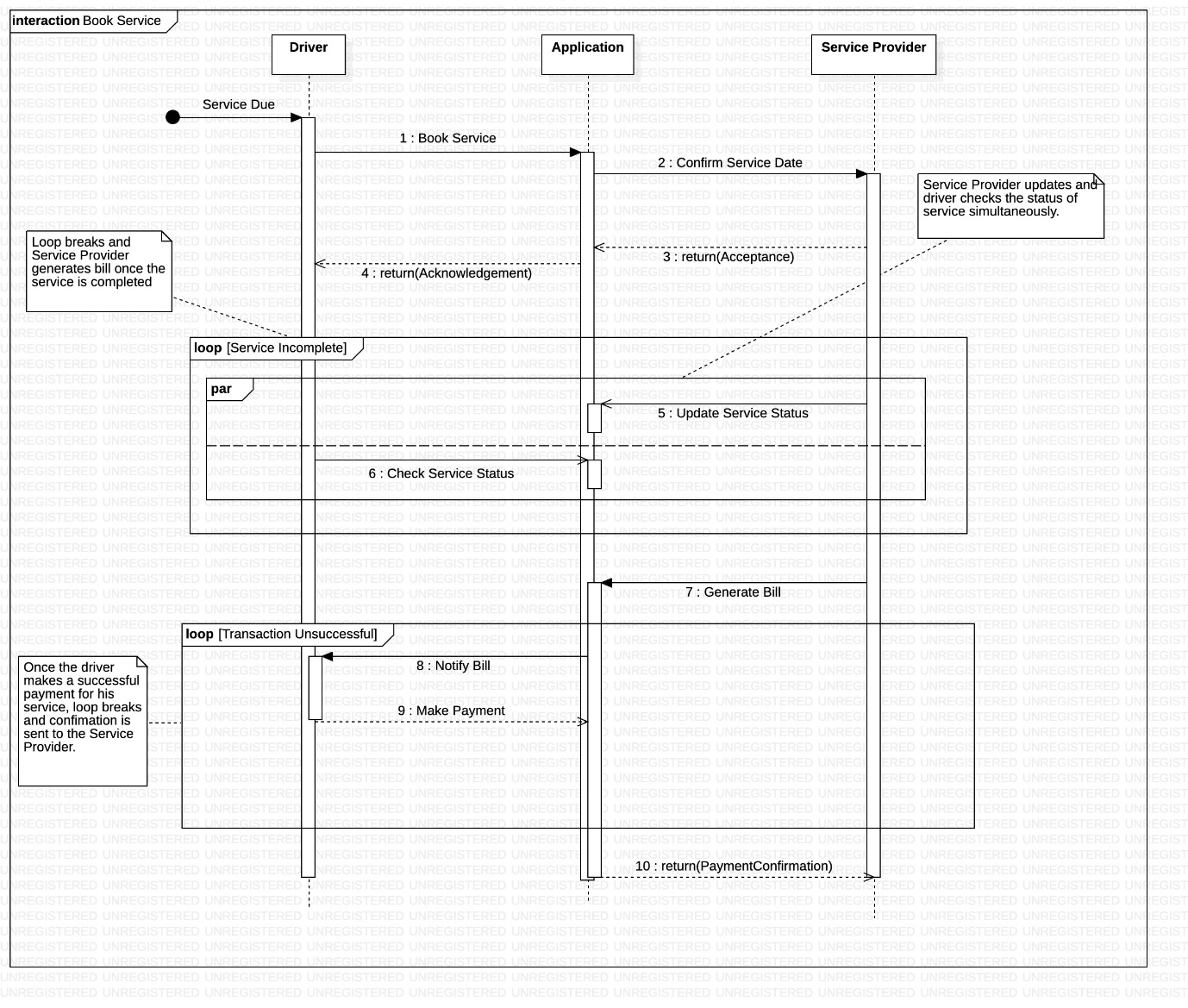
SD5

**Associated Usecase** – Book Service

**Collaborating classes** – Driver, Safety Application, Service Provider

**Steps:**

* Driver books service for his vehicle when its on due.
* The safety application class contacts the service provider class to confirm the required service date.
* Driver receives an acknowledgement from the application class, if the service provider returns an acceptance to the safety application class.
* While the status of the service is incomplete (loop frame), the service provider can update the status of the automobile and the driver can check this status parallelly (par frame).
* Once the service is complete, service provider generates the bill to the safety application class, which in-turn notifies the driver class.
* Finally driver makes his payment and a confirmation is sent to the service provider class.
* If the payment transaction is unsuccessful, the driver can attempt to make another transaction until he gets it successful.



**Safety application**