CS6110 OBJECT ORIENTED ANALYSIS AND DESIGN
SEMESTER – 5
PROJECT

AUTOMOBILE PRUDENT SYSTEM

GOKUL.S 2018103026
SRIHARI.S 2018103601

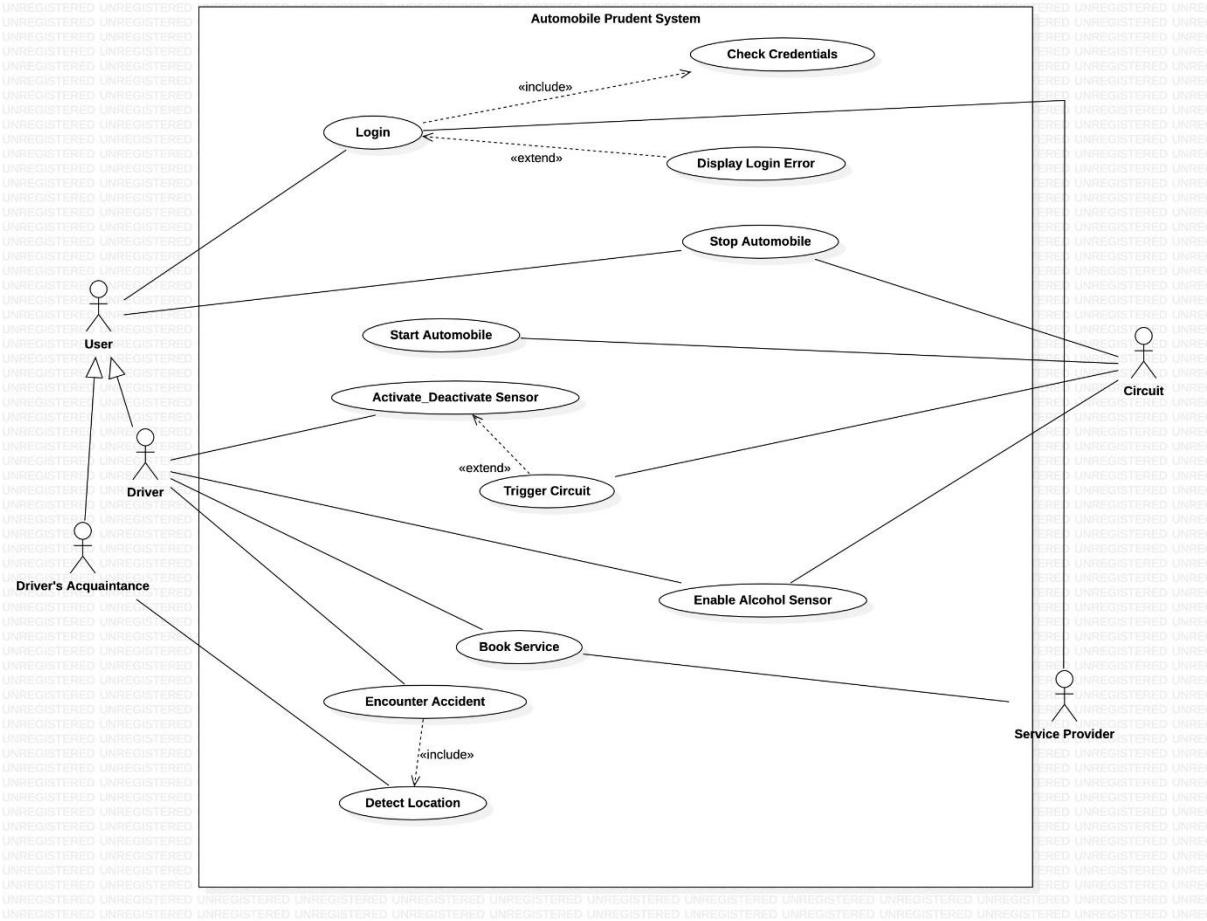
P - BATCH

Description:

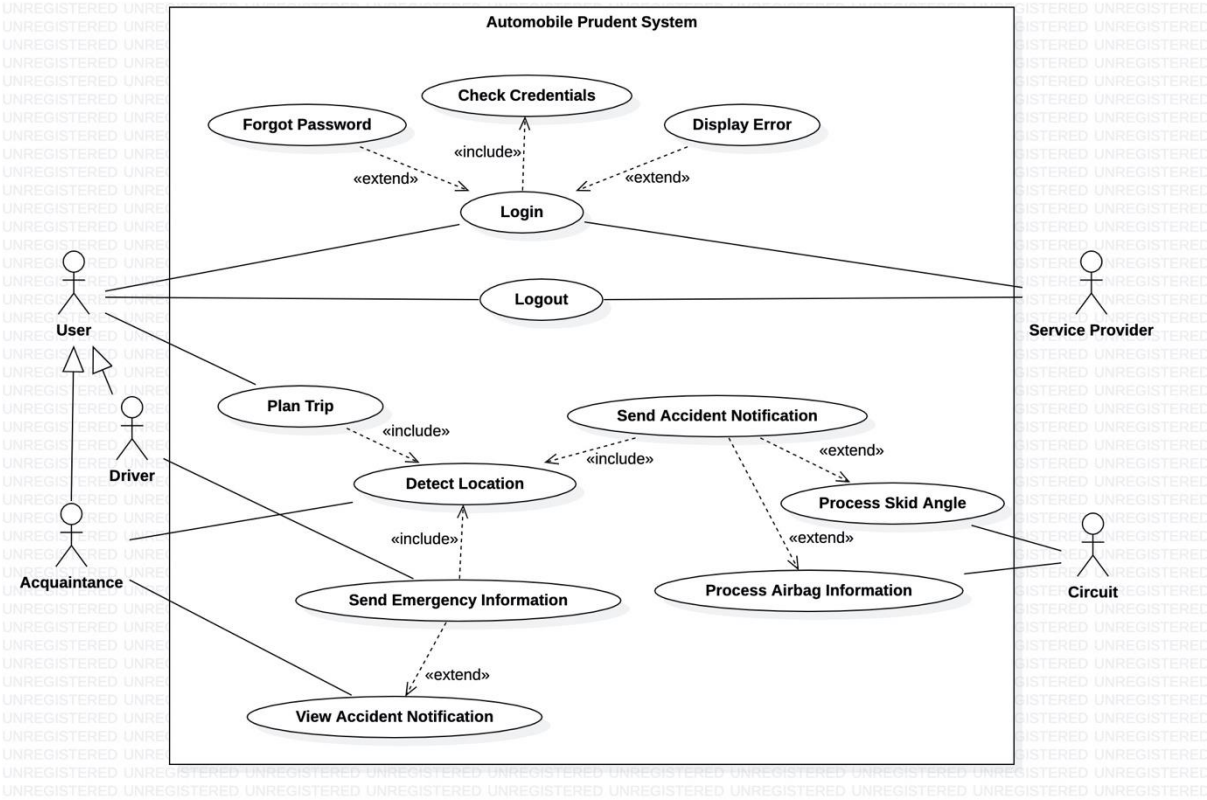
21st century can be infamously described as the era of road accidents and the number of causalities are rising at a dangerously alarming rate. Our project focusses on tackling this very issue by attacking it at its grassroots level. We propose a system that mandates the driver/cabbie to adhere to the basic safety measures i.e, a system that compels the drive to wear the helmet/seatbelt in order to get the vehicle on the go. Our system works based on the collaboration of a number of sensors and a meticulously designed circuit which is integrated with an interactive easy-to-use application. The crux of our proposal is to reduce the exorbitantly high ratio of causalities to accidents by performing a timely emergency act. The system is crafted in a sophisticated manner in order to cater to the requirements of both two-wheelers and four-wheelers. It enables the user to stay connected with his acquaintances at any point of time during his journey.

USE CASE MODELLING

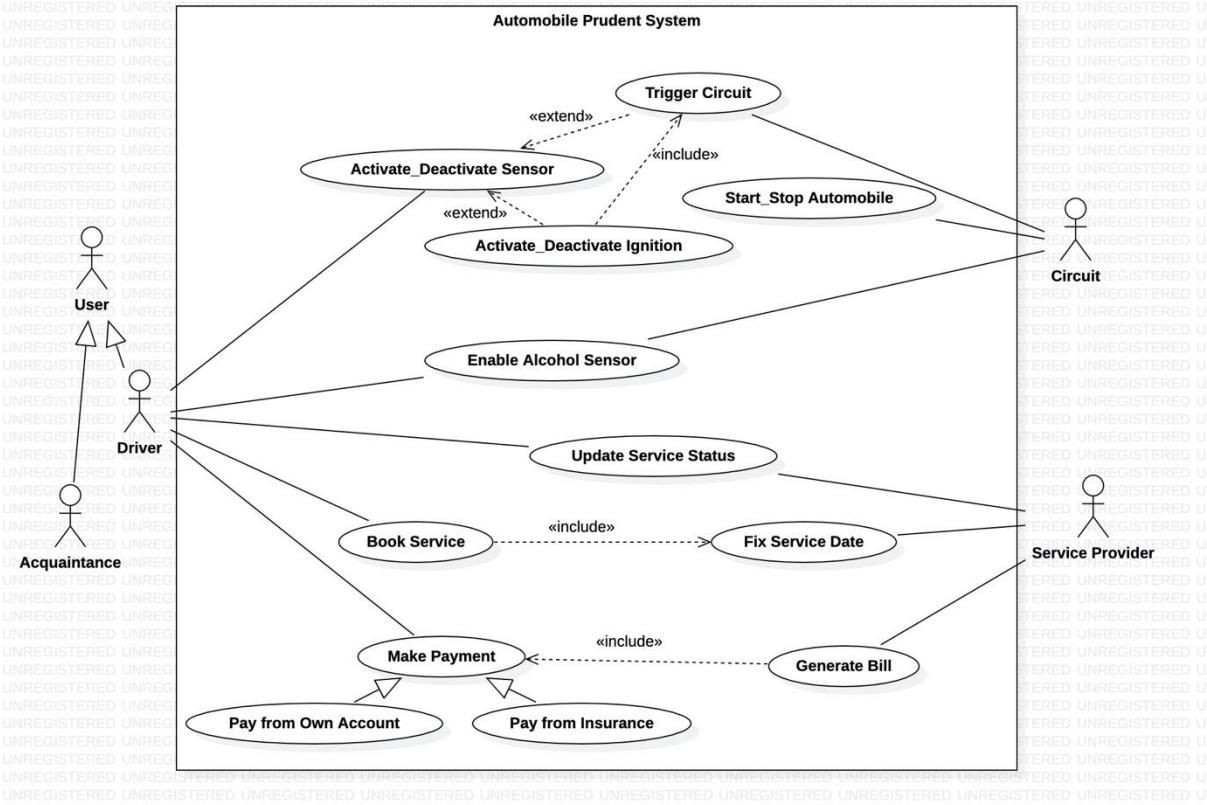
Use Case Diagram – 1



Use Case Diagram – 2



Use Case Diagram – 3



Use Case: Activate/Deactivate Sensor

Scope: System
Level: User goal level
Primary Actor: Driver
Stakeholders and Interests:

- Driver – Wears helmet/seatbelt to initiate the sensor.
- Co-passengers – Reach destination on time.

Preconditions:

Should be a registered and authenticated user.

Success Guarantee:

Should be allowed to start/stop the vehicle based on the state of the scenario.

Main Success Scenario:

- User wears the seat belt.
- Sensor detects the heart-beat and gets activated.
- User is able to start the vehicle.

Extensions:

- User inserts the seatbelt into the buckle.
- Heartbeat isn't detected as he isn't positioned properly.
- Tries starting vehicle but fails.

Special Requirements:

- Requires a state-of-the-art heartbeat sensor (four wheelers)/pressure sensor (two wheelers).

Frequency of Occurrence:

Continuous

Use Case: Activate/Deactivate Ignition

Scope: System
Level: User goal level
Primary Actor: Driver
Stakeholders and Interests:

- Driver – Uses the application to start vehicle without the keys.
- Co-passengers – Reach destination on time.

Preconditions:

Sensor should be activated/deactivated respectively.

Success Guarantee:

Circuit gets triggered to start/stop the automobile.

Main Success Scenario:

- User logs into the application.
- Driver wears the helmet/seatbelt.
- Uses application to start vehicle.
- Vehicle starts.

Extensions:

- User attempts logging into the application.
- Gets flagged with an error due to incorrect credentials.

Special Requirements:

- Needs a well-designed circuit and an interactive application that integrates it with the vehicle's ignition.

Frequency of Occurrence:

Rare

Use Case: Start Automobile

Scope: System
Level: User goal level
Primary Actor: Circuit
Stakeholders and Interests:

- Driver – Gets the vehicle on the fly.
- Co-passengers – Reach destination on time.
- Circuit – Enables start of the automobile depending on the state of the system.

Preconditions:

Triggered and activated circuit which enables to driver to start the vehicle.

Success Guarantee:

Drivers drives the vehicle.

Main Success Scenario:

- Sensor gets activated.
- Circuit gets triggered.
- Driver gets the vehicle moving.

Extensions:

- Sensor gets activated.
- Circuit gets triggered.
- Vehicle doesn’t start due to insufficient battery charge.

Special Requirements:

- Circuit should be synchronized with the ignition of the vehicle.

Frequency of Occurrence:

Continuous.

Use Case: Stop Automobile

Scope: System
Level: User goal level
Primary Actor: Circuit
Stakeholders and Interests:

- Driver – Gets the vehicle to a halt.
- Co-passengers – Remain safe during the journey.
- Circuit – Enables stop of the automobile depending on the state of the system.

Preconditions:

Triggered and de-activated circuit which enables to driver to stop the vehicle.

Success Guarantee:

Drivers unable to drive the vehicle.

Main Success Scenario:

- Vehicle is moving.
- Sensor gets de-activated as the user removes seatbelt/helmet.
- Circuit gets triggered.
- Vehicle comes to a halt.

Extensions:

- Sensor gets de-activated.
- Circuit gets triggered.
- Vehicle doesn’t stop due to insufficient battery charge.

Special Requirements:

- Circuit should be synchronized with the ignition of the vehicle.

Frequency of Occurrence:

Continuous.

Use Case: Trigger Circuit

Scope: System
Level: User goal level
Primary Actor: Circuit
Stakeholders and Interests:

- Driver – Start/Stop the vehicle.
- Co-passengers – Reach destination on time.
- Circuit – Enables start/stop of the automobile depending on the state of the system.

Preconditions:

An activated/de-activated sensor.

Success Guarantee:

Vehicle starts/stops depending on why the circuit got triggered.

Main Success Scenario:

- User wears the seat belt.
- Sensor detects the heart-beat and gets activated.
- Circuit gets triggered.
- User is able to start/stop the vehicle.

Extensions:

- User wears the seat belt.
- Sensor detects the heart-beat and gets activated.
- Circuit doesn't get triggered due to inconsistent communication with the ignition.

Special Requirements:

- Activate/deactivate functionality of the application should be synchronized with the circuit.

Frequency of Occurrence:

Always

Use Case: Login

Scope: System
Level: User goal level
Primary Actor: User, Service Provider
Stakeholders and Interests:

- User – Avail the services of the system
- Service Provider – Provide services to the users of the system
- Helpline workers – Help the users during crisis.

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- User can avail system services.
- Service provider offers required services.

Main Success Scenario:

- User enters login credentials.
- Logs in successfully.

Extensions:

- User enters login credentials.
- Log in unsuccessful.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

As and when required.

Use Case: Forgot Password

Scope: System
Level: Sub-functional level
Primary Actor: User, Service Provider
Stakeholders and Interests:

- User – Reset the forgotten password
- Service Provider - Reset the forgotten password, offer services to the user

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

Enter valid new password.

Main Success Scenario:

- User logs in successfully.

Extensions:

- User enters new password.
- Password requirements not satisfied.
- User unable to reset password.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

Rare

Use Case: Check Credentials

Scope: System
Level: Sub-functional level
Primary Actor: User, Service Providers
Stakeholders and Interests:

- User – Avail the services of the system
- Service Provider – Provide services to the users of the system
- Helpline workers – Help the users during crisis.

Preconditions:

- Registered user should have entered login credentials.

Success Guarantee:

- If login credentials match, user can avail services of the system.

Main Success Scenario:

- Registered user enters login credentials.
- System checks its validity.
- User successfully logs in upon entering consistent details.

Extensions:

- Registered user enters login credentials.
- System checks its validity.
- User is unable to log-in due to inconsistency in the entered details.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

Whenever user tries to log in.

Use Case: Display Error

Scope: System
Level: Sub-functional level
Primary Actor: User, Service Providers
Stakeholders and Interests:

- User – Avail a flawless entry into the application.
- Service Provider – Provide services to the users of the system
- Helpline workers – Help the users during crisis.

Preconditions:

- Registered user should have entered login credentials.

Success Guarantee:

- Enters incorrect login credentials.

Main Success Scenario:

- Registered user enters login credentials.
- System checks its validity.
- User unsuccessful in logging in upon entering inconsistent details.
- User flagged with an error.

Extensions:

- Registered user enters login credentials.
- System checks its validity.
- User successfully logs in upon entering consistent details.
- User not flagged with any error.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

As and when required.

Use Case: Logout

Scope: System
Level: Sub-functional level
Primary Actor: Users, Service Providers
Stakeholders and Interests:

- User – Logout of the system
- Service Provider – Logout of the system

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- Logged out of the system.

Main Success Scenario:

- User avails Plan Trip usecase of the application.
- Calculates the distance of the destination from his current location.
- Logs out of the application.

Extensions:

- User books service for his vehicle.
- Tries logging out of the system.
- But unable to do so due to bugs in the mobile.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

As and when required.

Use Case: Enable Alcohol Sensor

Scope: System
Level: User goal level
Primary Actor: Driver
Stakeholders and Interests:

- Driver – To be notified when drunk.
- Co-passengers – Ensure safe journey.

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- Vehicle is stopped.

Main Success Scenario:

- Drunk driver wears his helmet/seatbelt.
- Alcohol Sensor detects the consumption of alcohol.
- Primary sensor gets deactivated.
- Circuit gets triggered.
- Driver is unable to start his vehicle.

Extensions:

- Sober driver wears his helmet/seatbelt.
- Alcohol Sensor isn't enabled.
- Primary sensor gets activated.
- Circuit gets triggered.
- Driver is able to start his vehicle.

Special Requirements:

- System requires a state-of-the-art alcohol sensor

Frequency of Occurrence:

Rare

Use Case: Plan Trip

Scope: System
Level: User goal level
Primary Actor: User
Stakeholders and Interests:

- User – Plan trip in advance
- Co-passengers – Can join the user on his trip.

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- User gets precise knowledge of the destination.

Main Success Scenario:

- User avails the plan trip usecase.
- Enters the destination in the application.
- System reports back with information about the trip.

Extensions:

- User avails the plan trip usecase.
- Enters the destination in the application.
- System reports back an error due to invalid destination.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

As and when required

Use Case: Detect Location

Scope: System
Level: User goal level
Primary Actor: Driver's Acquaintances
Stakeholders and Interests:

- Driver – Helps him plan the trip
- Co-passengers – Can join the user on his trip.
- Helpline workers – Can receive accident location and provide timely help
- Driver's Acquaintances – Can track the driver's current location.

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- Acquaintance gets precise knowledge of the driver's current location.

Main Success Scenario:

- Driver goes on a long drive.
- There isn't any information about him for a long period of time.
- His mother tracks his current location.

Extensions:

- Driver goes on a long drive.
- There isn't any information about him for a long period of time.
- His mother tries to track his current location.
- But is unable to do so due to bugs in the GSM module.

Special Requirements:

- System requires a GSM module.

Frequency of Occurrence:

As and when required

Use Case: Send Emergency Information

Scope: System
Level: User goal level
Primary Actor: User
Stakeholders and Interests:

- User – Requests help under emergency situation.
- Co-passengers – Avail help along with the driver during emergency situations.
- Helpline workers – Can receive accident location and provide timely help

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- Helpline workers reach spot of accident.

Main Success Scenario:

- Driver is on a trip.
- Encounters an accident.
- His brother gets notified.
- He shares the spot of accident to the helpline workers.
- Helpline workers reach spot of accident and rescues the driver.

Extensions:

- Driver is on a trip.
- Encounters an accident.
- He becomes unconscious.
- He dies.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

Rare

Use Case: View Accident Notification

Scope: System
Level: User goal level
Primary Actor: Driver's Acquaintances
Stakeholders and Interests:

- Driver – Requests help under emergency situation.
- Co-passengers – Avail help along with the driver during emergency situations.
- Driver's Acquaintance – Can perform a timely action in case of an emergency.

Preconditions:

Should be a registered and authenticated user of the system.

Success Guarantee:

- Can use the send emergency usecase.

Main Success Scenario:

- Driver is on a trip.
- Encounters an accident.
- His brother receives notification from the system.
- He shares the spot of accident to the helpline workers.
- Helpline workers reach spot of accident and rescues the driver.

Extensions:

- Driver is on a trip.
- Encounters an accident.
- His brother receives notification from the system.
- He hasn't checked the application.
- Helpline workers aren't informed.
- Driver dies.

Special Requirements:

- User needs to be registered to the application.

Frequency of Occurrence:

Rare.

Use Case: Send Accident Notification

Scope: System
Level: Sub-functional level
Primary Actor: Circuit
Stakeholders and Interests:

- Driver – Requests help under emergency situation.
- Co-passengers – Avail help along with the driver during emergency situations.
- Driver's Acquaintance – Can perform a timely action in case of an emergency.
- Helpline workers – Can receive accident location and provide timely help

Preconditions:

- Should be a registered and authenticated user of the system.
- The airbag state/ skid angle should be in the danger level.

Success Guarantee:

- The location of the driver gets tracked.
- His acquaintances get notified.

Main Success Scenario:

- Driver is on a trip.
- Encounters an accident.
- His car's airbag gets ejected.
- His brother receives notification from the system.
- He shares the spot of accident to the helpline workers.
- Helpline workers reach spot of accident and rescues the driver.

Extensions:

- Driver is on a trip.
- Encounters an accident.
- Due to bugs in his airbag system, it doesn't get ejected.
- His brother doesn't receive any notification.
- Driver dies.

Special Requirements:

System requires a state-of-the-art level detector/ airbag system that is synchronized with the circuit.

Frequency of Occurrence:

Rare.

Use Case: Process Airbag Information
Use Case: Process Skid Angle

Scope: System
Level: User goal level
Primary Actor: Circuit
Stakeholders and Interests:

- Driver – Requests help under emergency situation.
- Co-passengers – Avail help along with the driver during emergency situations.
- Driver's Acquaintance – Can perform a timely action in case of an emergency.
- Helpline workers – Can receive accident location and provide timely help

Preconditions:
Should be a registered and authenticated user of the system.
Success Guarantee:
Can use send accident notification use case.
Main Success Scenario:

- Driver is on a trip.
- Encounters an accident
- His bike gets thrown off the road.
- Level detector senses an accident.
- Acquaintances receive accident notification.
- Helpline workers reach the spot of the accident.
- Driver is saved.

Extensions:

- Driver is on a trip.
- Encounters a minor accident.
- Level detector doesn't sense any mishap.
- His brother doesn't receive any notification.

Special Requirements:
System requires a state-of-the-art level detector/ airbag system that is synchronized with the circuit.
Frequency of Occurrence: Continuous.

Use Case: Update Service Status

Scope: System
Level: User goal level
Primary Actor: Service Provider
Stakeholders and Interests:

- Driver – To get his vehicle serviced.
- Service Provider – Can provide timely updates during the service process.

Preconditions:

Driver and the Service Provider should be registered and authenticated users of the system.

Success Guarantee:

Driver gets notified when his vehicle is ready to be picked up.

Main Success Scenario:

- Driver books service for his vehicle using the application.
- Checks the status of the service.
- Collects the vehicle once its ready.

Extensions:

- Driver books service for his vehicle using the application.
- Checks the status of the service.
- Service Provider fails to update the status regularly.
- Driver is unable to catch-up with the service status.

Special Requirements:

- Interactive application for the Service Provider to update service details.

Frequency of Occurrence:

As and when required.

Use Case: Fix Service Date

Scope: System
Level: User goal level
Primary Actor: Driver
Stakeholders and Interests:

- Driver – To get his vehicle serviced.
- Service Provider – Can provide timely updates during the service process.

Preconditions:

- Should be a registered and authenticated user of the system.
- Driver should use the book service functionality.

Success Guarantee:

- Date of service gets fixed.

Main Success Scenario:

- Driver books service for his vehicle using the application.
- Date of service is fixed.
- Driver drops his vehicle for service.

Extensions:

- Driver books service for his vehicle using the application.
- Service Provider has a tight schedule on the desired date of service.
- Date of service isn't confirmed.

Special Requirements:

- Interactive application for the user to fix the date of service.

Frequency of Occurrence:

As and when required.

Use Case: Generate Bill

Scope: System
Level: User goal level
Primary Actor: Service Provider
Stakeholders and Interests:

- Driver – To get his vehicle serviced.
- Service Provider – Generating the final bill for the service.

Preconditions:

- Should be a registered and authenticated user of the system.
- Driver should have scheduled a service beforehand.

Success Guarantee:

- Driver makes the payment.

Main Success Scenario:

- Driver books service for his vehicle using the application.
- Date of service is fixed.
- Service Provider generates the bill after the service is over.
- Driver makes payment and picks his vehicle.

Extensions:

- Driver books service for his vehicle using the application.
- Date of service is fixed.
- He doesn't drop the vehicle at the specified date
- Service process gets halted.
- Bill isn't generated.

Special Requirements:

- Interactive application for the Service Provider to update the bill.

Frequency of Occurrence:

As and when required.

General Use Case: Make Payment
Specialized Use Case: Pay From Own Account
Specialized Use Case: Pay From Insurance

Scope: System
Level: User goal level
Primary Actor: Driver
Stakeholders and Interests:

- Driver – To get his vehicle serviced.
- Service Provider – Enable the user to make payment

Preconditions:

- Should be a registered and authenticated user of the system.
- Driver should have scheduled a service beforehand.
- Service Provider should have generated the bill.

Success Guarantee:

- Driver picks up the vehicle.

Main Success Scenario:

- Driver books service for his vehicle using the application.
- Date of service is fixed.
- Service Provider generates the bill after the service is over.
- Driver makes payment and picks his vehicle.

Extensions:

- Driver books service for his vehicle using the application.
- Date of service is fixed.
- Service Provider generates the bill after the service is over.
- Transaction made by the driver fails.

Special Requirements:

- Interactive application for the driver to make payment.

Frequency of Occurrence:
As and when required.

Use Case: Encounter Accident

Scope: System
Level: User goal level
Primary Actor: Driver.
Stakeholders and Interests:

- Driver – Requests help under emergency situation.
- Co-passengers – Avail help along with the driver during emergency situations.
- Driver's Acquaintance – Can perform a timely action in case of an emergency.

Preconditions:

- Should be a registered and authenticated user of the system.

Success Guarantee:

- Location of the injured driver is sent to his acquaintances and helpline workers.

Main Success Scenario:

- Driver is on a trip.
- Encounters an accident.
- His brother receives notification from the system.
- He shares the spot of accident to the helpline workers.
- Helpline workers reach spot of accident and rescues the driver.

Extensions:

- Driver is on a trip.
- Encounters an accident.
- His brother receives notification from the system.
- He hasn't checked the application.
- Helpline workers aren't informed.
- Driver dies.

Special Requirements:

- User needs to be registered to the application.
- System requires a state-of-the-art level detector/ airbag system that is synchronized with the circuit.

Frequency of Occurrence:

Rare.

DOMAIN MODELING AND PARTIAL CLASS MODELLING

List of Domain Terms:

Automobile, accident, safety, vehicle, two-wheeler, brake, four-wheeler, passengers, acquaintance, injury, seat-belt, helmet, maintenance, mobile safety application, speedometer, circuit, odometer, location, speed-breaker, pillion-rider, level detector, traffic-signal , driver, zebra crossing, ignition, Antilock breaking system, User, airbag system, tyre, Global Positioning System, Rear View Camera, alcohol sensor, road safety helpline, accelerator, insurance company, insurance, Ambulance, Heartbeat Sensor, Traffic Police, Service Provider, Central lock system, GSM module. Accident detector, Sensor

Step 1: Finding Classes by extracting nouns from the list of domain terms as-well-as using a category list:

Automobile, Vehicle, Two-wheeler, Brake, Four-wheeler, Passengers, Acquaintance, Seat-belt, Helmet, Speedometer, Mobile safety application, Circuit, Odometer, Speed breaker, Pillion rider, Level detector, Traffic signal, Driver, Ignition, Antilock breaking system, User, Airbag system, tyre, Global Positioning System, Rear View Camera, alcohol sensor, Accelerator, Insurance, Ambulance, Heartbeat sensor, Traffic Police, Service Provider, Central lock system, GSM module, Location. Accident detector, Sensor

Step 2: Refining the above list by eliminating spurious classes:

Automobile, Two-wheeler, Four-wheeler, Acquaintance, Helmet Sensor, Mobile safety application, Circuit, Level detector, Driver, Ignition, User, Airbag system, Alcohol Sensor, Heartbeat Sensor, Service Provider, GSM Module, Location. Accident detector, Sensor

Step 3: Preparation of Data Dictionary:

- **Automobile –** Consists of the technical specifications of the automobile. Has an in-built GSM module and a circuit. Driver can start the automobile and reach the destination.
- **Two-wheeler -** Consists of the technical specifications of the two-wheeler.
- **Four-wheeler -** Consists of the technical specifications of the four-wheeler.
- **Acquaintance –** Related to the driver. A user of the system.
- **Sensor -** Triggers the circuit to start/stop the ignition depending upon the situation.
- **Helmet Sensor –** Triggers the ignition of the two-wheeler. Gets activated/de-activated depending upon the state of the helmet.
- **Alcohol Sensor –** Triggers the stoppage of the vehicle in-case the driver is intoxicated. Gets activated/de-activated depending upon the state of the driver.
- **Heartbeat Sensor –** Trigger the ignition of the four-wheeler. Gets activated when the driver wears the seatbelt properly.
- **GSM Module –** Tracks the location of the vehicle. It is built-onto the automobile during the manufacturing process.
- **Ignition –** Holds the state of ignition of the vehicle. Can be triggered using the keys as-well-as the mobile safety application.
- **Level Detector –** Detects an accident. It triggers the circuit when the skid angle crosses the threshold, which in-turn informs the app to send a notification.
- **Airbag System –** Detects an accident. It triggers the circuit when the airbags get ejected, which in-turn informs the app to send a notification.
- **Accident Detector -** Detects an accident when the parameters of the vehicle cross the threshold
- **User –** Makes use of the functionalities provided by the system.

- **Circuit –** Stimulates the ignition of the vehicle depending upon the state of the sensors. Sends notification to the driver's acquaintances depending upon the state of the level detector/ airbag system.
- **Service Provider –** A user of the system. Deals with the service of the vehicles whenever requested by the driver. Updates the status of the service and generates bill when its over.
- **Location –** Indicates the position of the vehicle.
- **Driver –** A user of the system. Has the ability to start/stop the vehicle.
- **Mobile safety application –** Integrates the entire system. Tracks the location of the vehicle. Driver/Acquaintances can send notifications to helpline workers in-case of an emergency. Driver can fix the date of service of his vehicle. Service Provider can update the service status as-well-as generate the bill when its over. Consists of a built-in payment system.

Step 4: Finding associations-using relationships that are verbs

- Application **Tracks** Automobile
- Application **Tracks** Two-wheeler
- Application **Tracks** four-wheeler
- Circuit **Stimulates** Ignition
- Circuit **Monitors** GSM Module
- Sensor **Triggers** Circuit
- GSM Module **Pings** Driver's Acquaintance
- Driver **Drives** Automobile
- User **Utilizes** Application
- Driver **Utilizes** Application
- Service Provider **Utilizes** Application

- Driver's Acquaintance **Utilizes** Application
- Service Provider **Assists** Driver
- Driver **ContactedBy** Driver's Acquaintance
- Service Provider **Services** Automobile
- Level Detector **Signals** Circuit
- Airbag System **Signals** Circuit
- GSM Module **Locates** Location
- Level Detector **Reports** GSM Module
- Airbag System **Reports** GSM Module
- Driver's Acquaintance **Receives** Location
- Sensor **Activates/Deactivates** Ignition
- Automobile **ConsistsOf** Ignition **(Composition)**
- Circuit **Notifies** Driver's Acquaintance
- Level Detector **Informs** Ignition
- Airbag System **Informs** Ignition

Step 5: Refining associations by eliminating spurious associations:

ASSOCIATION	DESCRIPTION
Application **Tracks** Automobile	Application tracks the state of the automobile at each and every instant of time.
Circuit **Stimulates** Ignition	Circuit stimulates the ignition depending on the state of the sensors and detectors.
Sensor **Triggers** Circuit	Sensors trigger the circuit depending on whether they are activated or deactivated.

GSM Module **Pings** Driver's Acquaintance	In case of an accident GSM Module pings the driver's acquaintances.
User **Utilizes** Application	User makes use of the functionalities provided by the system with the help of the application.
Service Provider **Assists** Driver	When driver's vehicle is in need of service, the service provider assists him.
Driver **ContactedBy** Driver's Acquaintance	In case of an accident the driver's acquaintances get notified as they are related to the driver.
Airbag System **Signals** Circuit	Airbag System Signals Circuit depending upon its state.
Level Detector **Signals** Circuit	Level Detector Signals Circuit depending upon its state.
Level Detector **Reports** GSM Module	Level Detector Reports GSM Module when the parameters of the vehicle cross the threshold.
Airbag System **Reports** GSM Module	Airbag System Reports GSM Module when the parameters of the vehicle cross the threshold.
Automobile **ConsistsOf** Ignition	Ignition is a part of the automobile and it can't independently exist without it.
GSM Module **Locates** Location	GSM Module Locates Location at every instant of time and notifies the driver's acquaintance in case of an accident.

Reasons for eliminating spurious associations:

ASSOCIATION	REASON
Application **Tracks** Two-wheeler Application **Tracks** four-wheeler	Removed as Two-wheeler and Four-wheeler classes can be generalized to an Automobile class.
Circuit **Monitors** GSM Module	Redundant as the circuit indirectly achieves this functionality by signalling the accident detector, which in turn reports to the GSM Module.
Driver **Drives** Automobile	Redundant as it doesn't specify any functionality of the system.
Service Provider **Services** Automobile	Redundant as it doesn't specify any functionality of the system.
Driver's Acquaintance **Receives** Location	Redundant as the driver's acquaintance indirectly receives the location of the vehicle when the GSM Module pings him/her after detecting the location.
Driver **Utilizes** Application Service Provider **Utilizes** Application Driver's Acquaintance **Utilizes** Application	Removed as Driver, Service Provider and Driver's Acquaintance classes can be generalized to a User class.
Sensor **Activates/Deactivates** Ignition	Redundant as the sensor indirectly activates/deactivates the ignition by triggering the circuit, which in turn stimulates the ignition.

Circuit **Notifies** Driver's Acquaintance	Redundant as the driver's acquaintance receives a notification when the accident detector reports to the GSM Module in case of an accident, which in turn pings the driver's acquaintance.
Level Detector **Informs** Ignition Airbag System **Informs** Ignition	Level Detector/ Airbag System classes are generalized to an accident detector class which signals the circuit to stimulate an ignition.

Step 6: Identifying the attributes of the associations.

ASSOCIATION	ATTRIBUTES
Application **Tracks** Automobile	1 **TO** 1..*
Circuit **Stimulates** Ignition	1 **TO** 1
Sensor **Triggers** Circuit	1..* **TO** 1
GSM Module **Pings** Driver's Acquaintance	1 **TO** 1..*
User **Utilizes** Application	1 **TO** 1..*
Service Provider **Assists** Driver	1 **TO** 1..*
Driver **ContactedBy** Driver's Acquaintance	1..* **TO** 1..*
Level Detector **Signals** Circuit	1 **TO** 1

Airbag System **Signals** Circuit	1 **TO** 1
Level Detector **Reports** GSM Module	1 **TO** 1
Airbag System **Reports** GSM Module	1 **TO** 1
Automobile **ConsistsOf** Ignition	1 **TO** 1 It's a **composition** relationship.
GSM Module **Locates** Location	1 **TO** 1

Step 7: Identifying the attributes of the classes.

CLASSES	ATTRIBUTES
Automobile	Manufacturing Company Fuel tank capacity Mileage Fuel type Colour Model Engine Number
Two-wheeler	Threshold Angle Chassis Number
Four-wheeler	Number of Airbag modules Number of Crash Sensors Number of cylinders
Driver's Acquaintance	Relation Blood Group

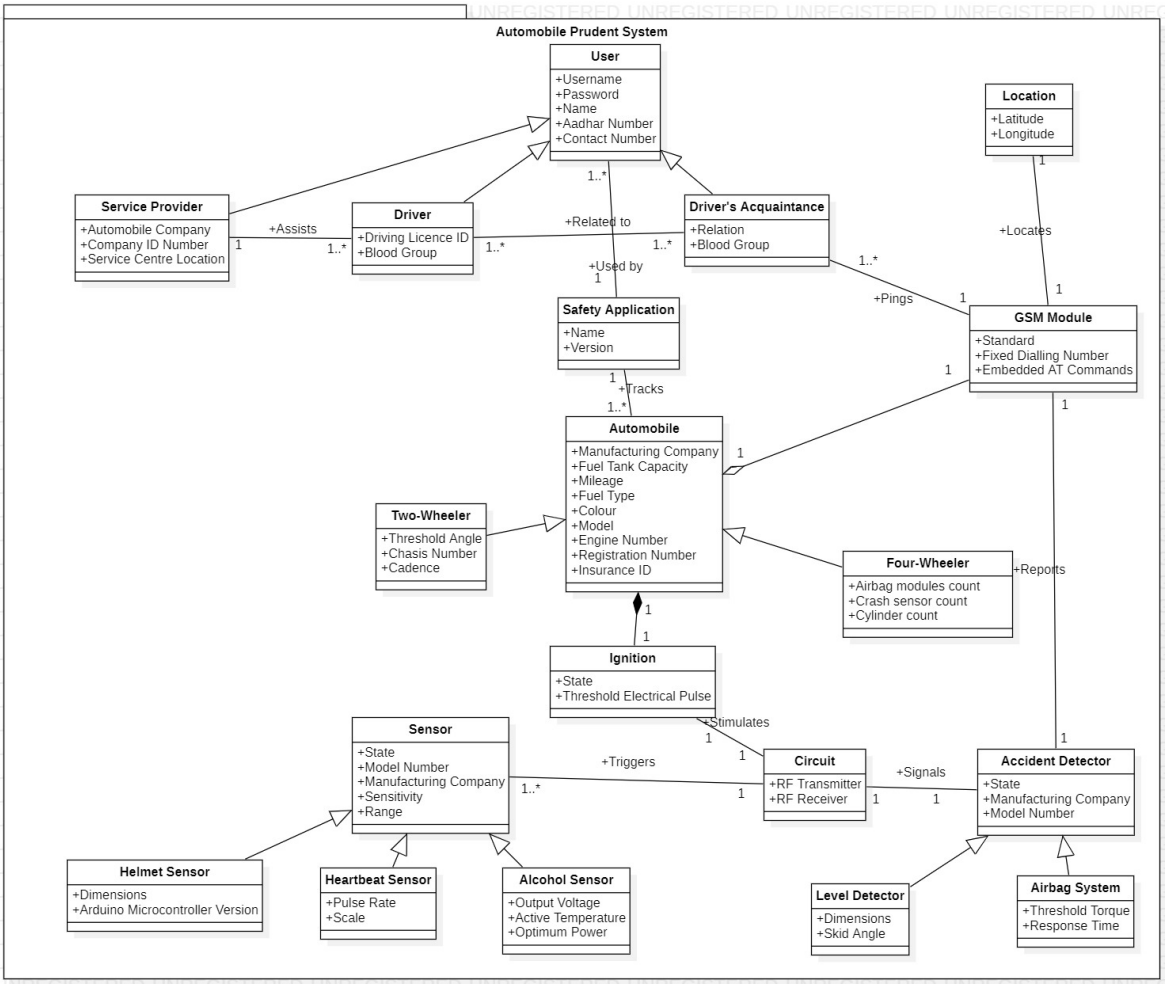
Helmet Sensor	Dimensions Arduino Microcontroller Version
Mobile safety application	Name Version
Circuit	RF Transmitter RF Receiver
Level detector	Dimension Skid Angle
Driver	Driving License ID Vehicle Registration No Insurance ID Blood Group
Ignition	State Threshold electrical pulse
User	Username Password Name Aadhar Number Contact Number
Airbag system	Threshold Torque Response Time
Alcohol Sensor	Output Voltage Active Temperature Optimum Power
Heartbeat Sensor	Pulse Rate Scale
Service Provider	Automobile Company Company Identification Number Server Centre Location
GSM Module	Standard

	Fixed Dialling Number Embedded AT commands
Location	Latitude Longitude
Accident Detector	State Manufacturing Company Model No.
Sensor	State Model No. Manufacturing Company Sensitivity Range

Step 8: Organizing and Simplifying classes using inheritance.

- **User** – Generalized Version of Driver, Service Provider and Driver's Acquaintance
- **Sensor** - Generalized Version of Helmet Sensor, Heartbeat Sensor and Alcohol Sensor
- **Accident Detector** - Generalized Version of Level Detector and Airbag System.

Step 9: Partial Class Model



CLASS MODELLING

CLASS RESPONSIBILITY COLLABORATION CARDS

SAFETY APPLICATION	
RESPONSIBILITY	**COLLABORATION**
Login(),Update_Details(), Change_password(), Make_payment()	User, Automobile

AUTOMOBILE	
RESPONSIBILITY	**COLLABORATION**
Start(), Stop()	Safety Application, Ignition, GSM Module

DRIVER	
RESPONSIBILITY	**COLLABORATION**
Fix_service_date(), Check_service_status(), Send_emergency_notification()	ServiceProvider, Driver's Acquaintance

DRIVER'S ACQUAINTANCE	
RESPONSIBILITY	**COLLABORATION**
Detect_driver_location(), Send_notification()	Driver, GSM Module

SERVICE PROVIDER	
RESPONSIBILITY	**COLLABORATION**
Update_service_status(), Generate_bill()	**Driver**

USER	
RESPONSIBILITY	**COLLABORATION**
Plan_trip()	**Safety application**

IGNITION	
RESPONSIBILITY	**COLLABORATION**
Find_state(), Check_condition()	**Automobile, Circuit**

CIRCUIT	
RESPONSIBILITY	**COLLABORATION**
Convert_to_state()	**Sensor, Accident detector, Ignition**

SENSOR	
RESPONSIBILITY	**COLLABORATION**
Trigger_circuit(), Activate(), Deactivate()	**Circuit**

LEVEL DETECTOR	
RESPONSIBILITY	**COLLABORATION**
Detect_state(), Signal_circuit(), Report_GSM_Module()	**Circuit, GSM Module**

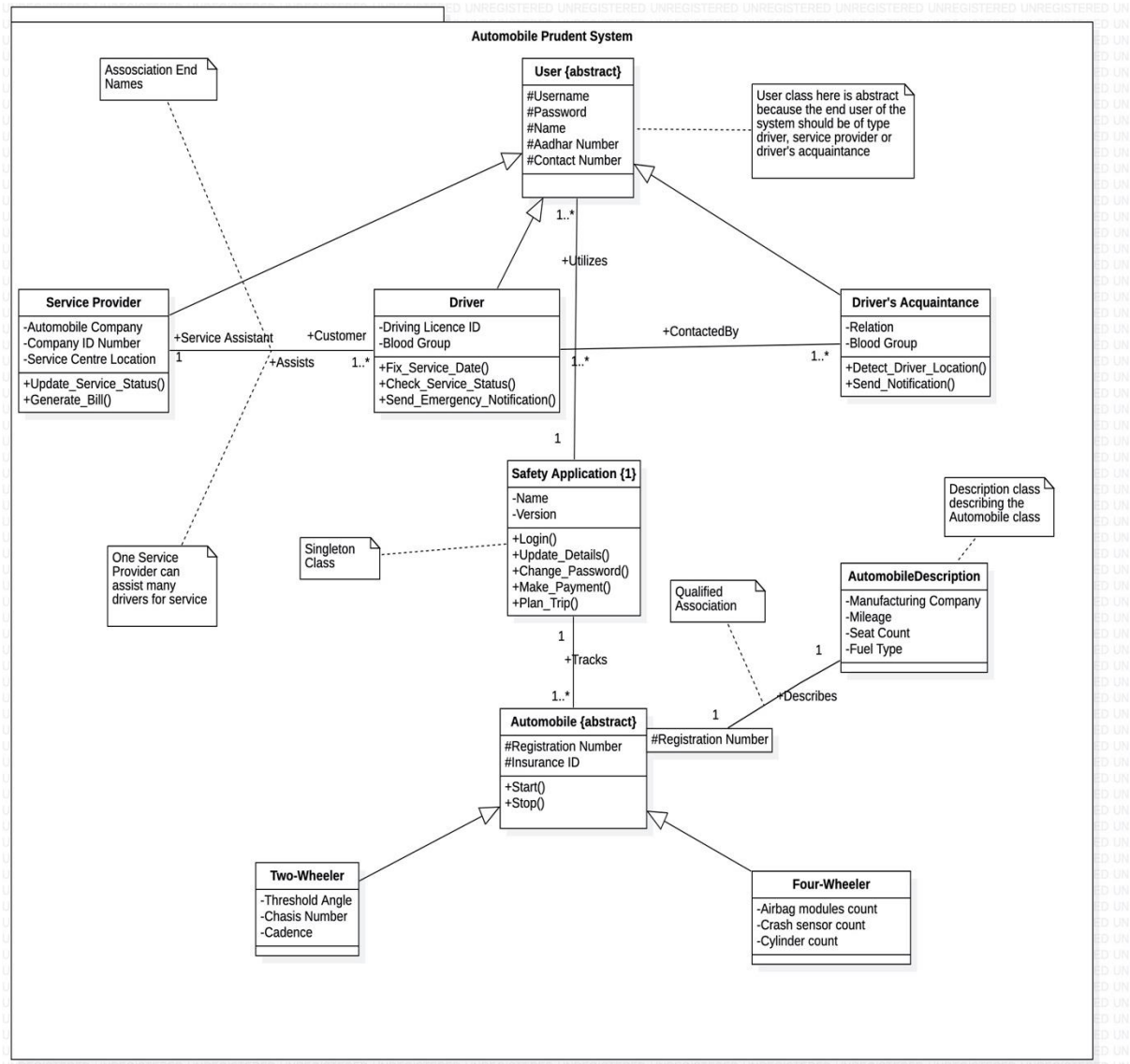
AIRBAG SYSTEM	
RESPONSIBILITY	**COLLABORATION**
Detect_state(), Signal_circuit(), Report_GSM_Module()	Circuit, GSM Module

GSM MODULE	
RESPONSIBILITY	**COLLABORATION**
Send_location()	Level detector, Airbag System, Automobile, Location, DriverAcquaintance

LOCATION	
RESPONSIBILITY	**COLLABORATION**
Get_Location()	GSM Module

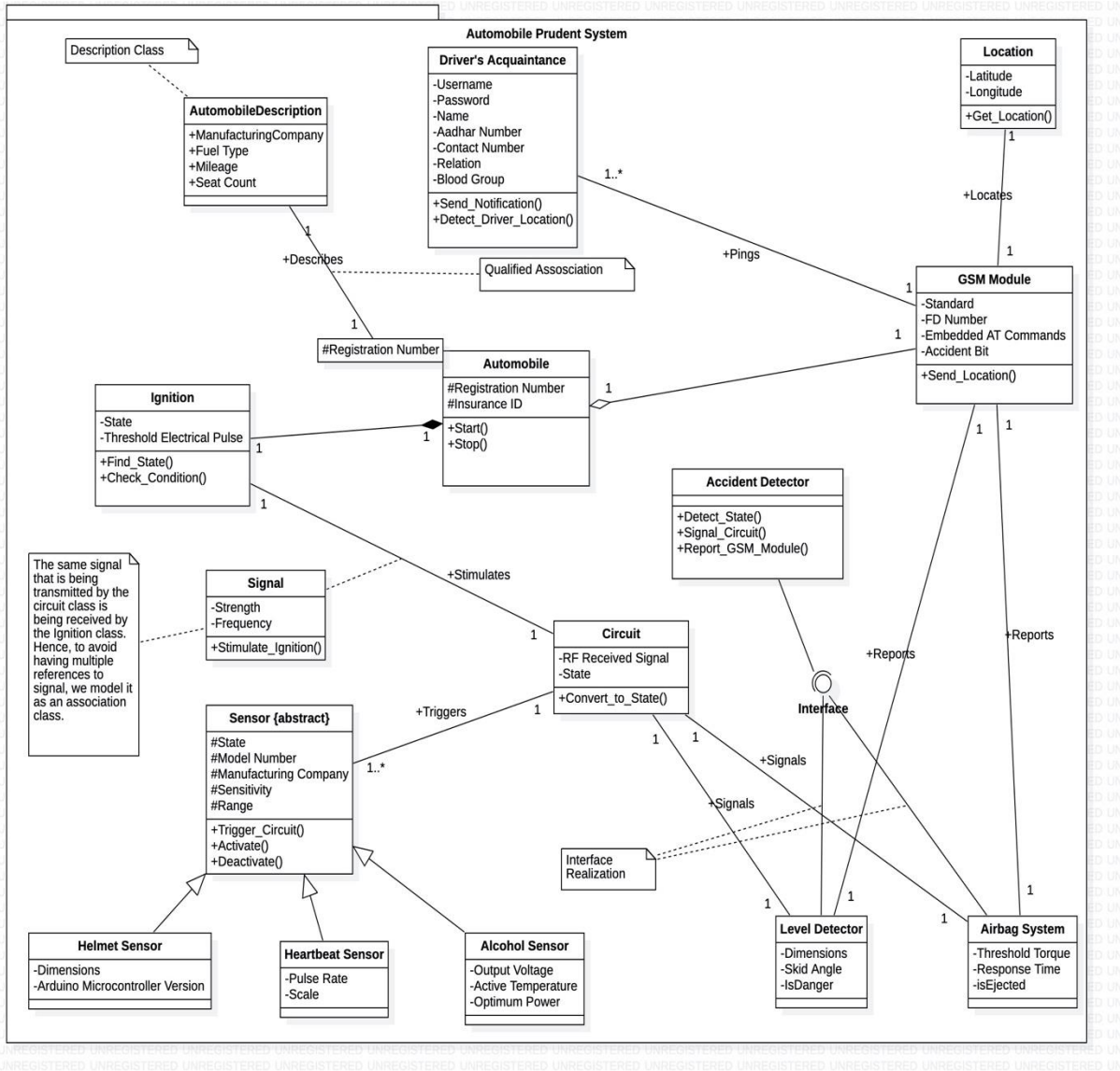
SIGNAL	
RESPONSIBILITY	**COLLABORATION**
Stimulate_ignition()	Ignition, Circuit

CLASS DIAGRAM – 1 → VERSION 1



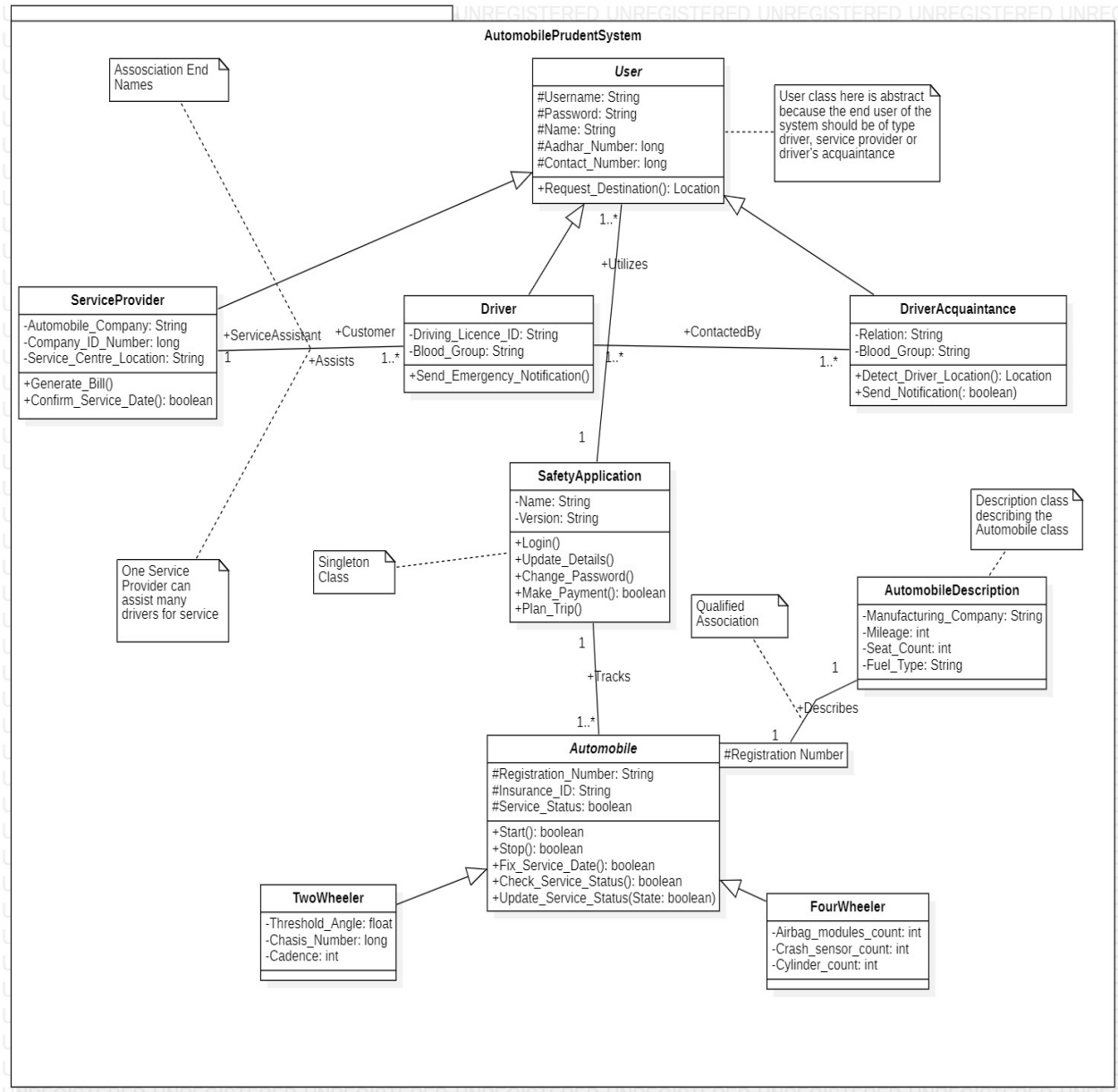
METHOD	DESCRIPTION
Plan_Trip	It returns the distance to the destination location from the current location when the user enters it.
Update_Service_Status	Service Provider updates the status of the service into the application.
Generate_bill	Service Provider generates the final bill upon completion of the service
Fix_Service_Date	Driver chooses the required service date.
Check_Service_Status	Driver checks the current status of his automobile's service.
Send_emergency_notification	Driver requests help from emergency workers in case of an accident.
Detect_driver_location	Used to detect the current location of the vehicle.
Send_notification	Driver's acquaintance requests help from emergency workers in case the driver undergoes an accident.
Login	User logs into the application using this functionality.
Update_details	User updates his account details.
Change_password	User updates his password upon entering his valid current password.
Make_payment	Driver uses this functionality to pay for his service.
Start	Used by the driver to start the automobile.
Stop	Used by the driver to stop the automobile.

CLASS DIAGRAM – 2 → VERSION 1

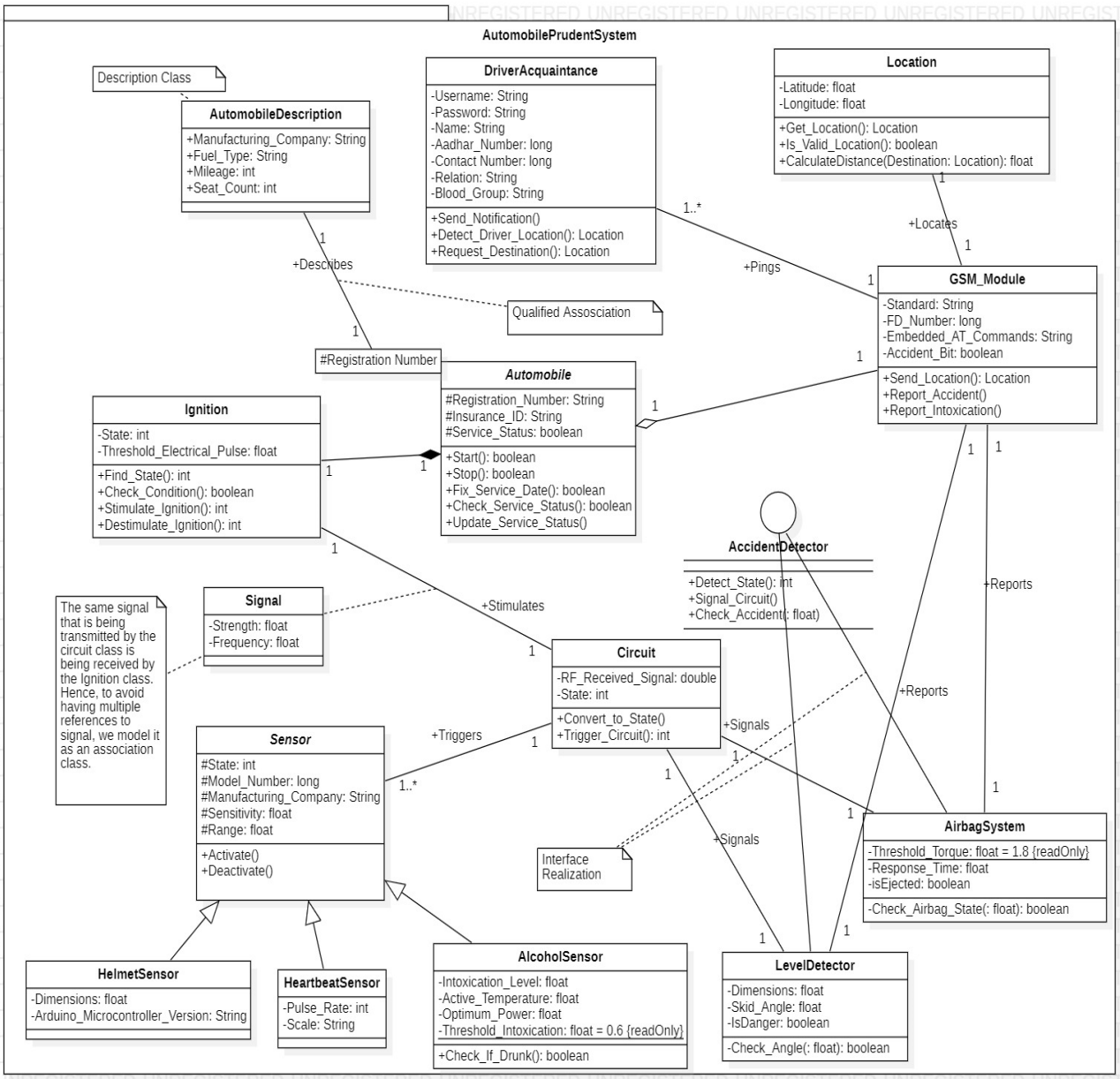


METHOD	DESCRIPTION
Find_state	If Check_condition() returns a true flag, state of the ignition is set to “active”.
Check_condition	If the strength of the received signal crosses the threshold electrical pulse, it returns a true flag.
Send_location	GSM Module sends the driver’s location to his acquaintances in the event of an accident.
Trigger_circuit	If the state of the sensor is active, the circuit gets triggered.
Activate	It activates the sensor if the helmet/seatbelt is worn by the driver, or the level of intoxication is less than the threshold.
Deactivate	It de-activates the sensor if the helmet/seatbelt is not worn by the driver, or the level of intoxication crosses the threshold.
Convert_to_state	Converts the received RF signal into a 44oolean value that indicates the state of the circuit.
Stimulate_ignition	If the state of the circuit is true, it sends an electrical pulse indicating the same to the ignition circuit.
Detect_state	It activates the accident detector if the skid angle crosses the threshold or if the airbag system gets ejected.
Signal_circuit	If the state of the accident detector is true, it sends a signal indicating the same to the circuit.
Report_GSM_Module	If the state of the accident detector is true, it reports the GSM Module to track the driver’s location and in-turn inform his acquaintance.

CLASS DIAGRAM – 1 → VERSION 2



CLASS DIAGRAM – 2 → VERSION 2

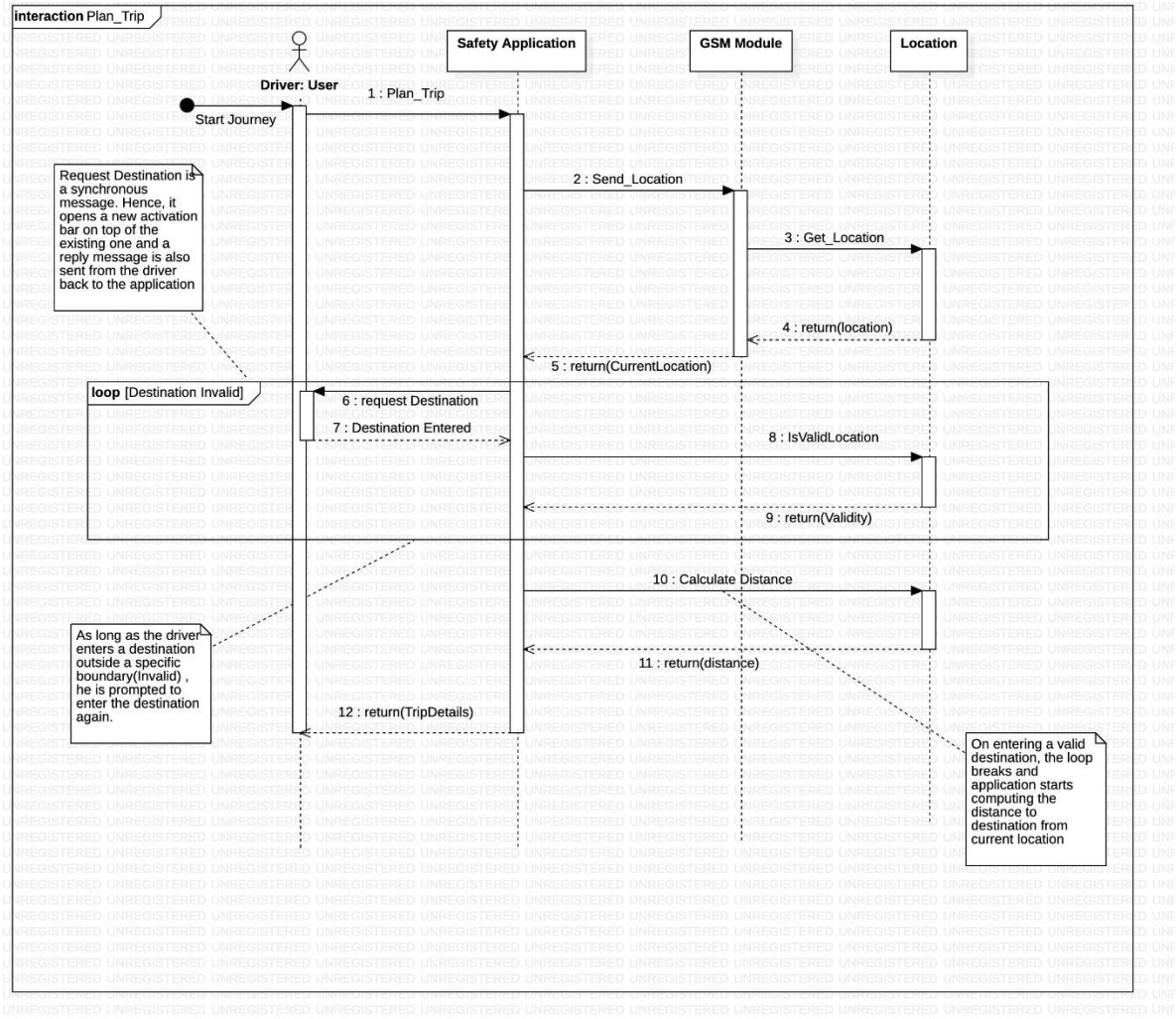


SEQUENCE MODELLING

SD1 **Associated Usecase** – Plan_Trip --> **VERSION 1**
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.



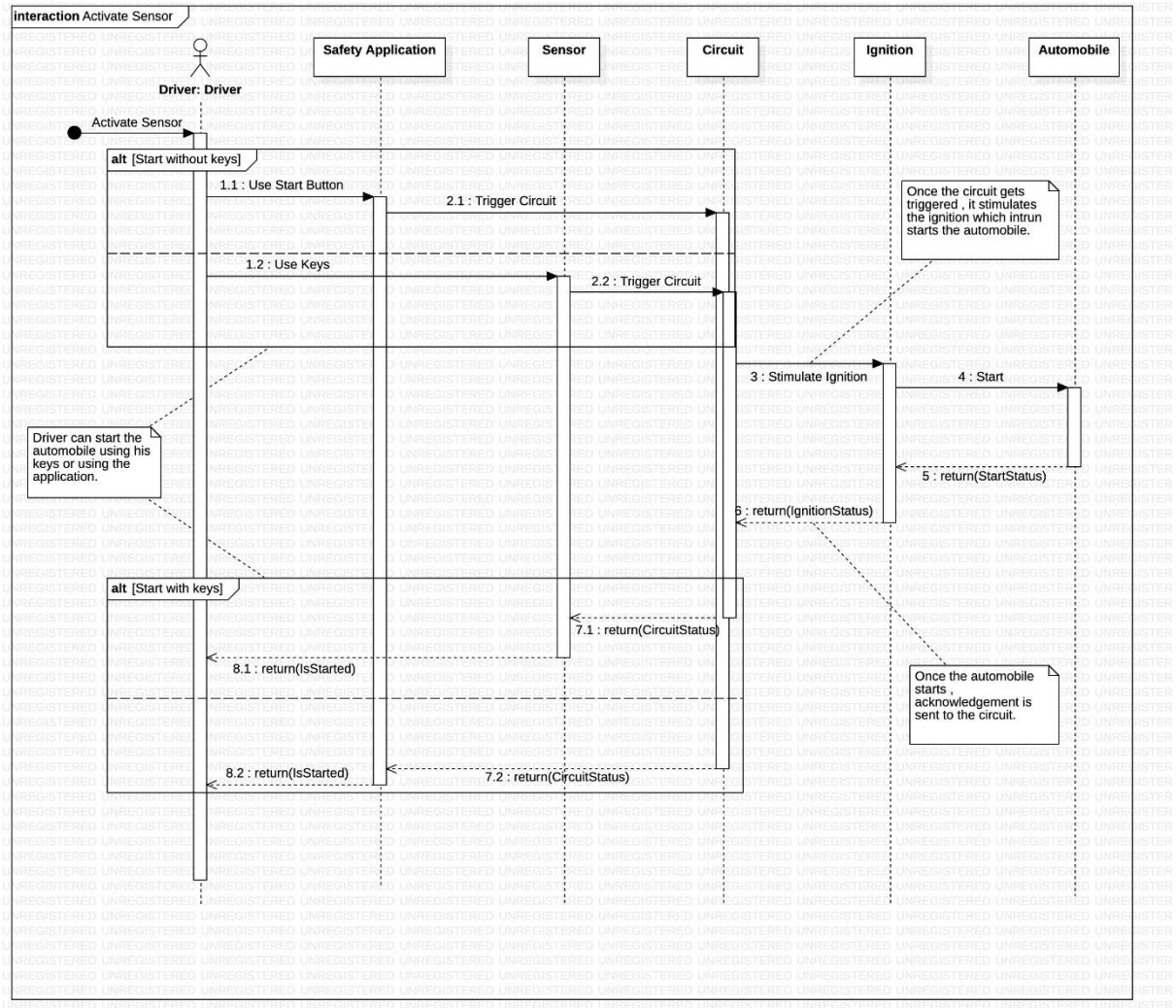
SD2

Associated Usecase – Activate Sensor --> **VERSION 1**

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

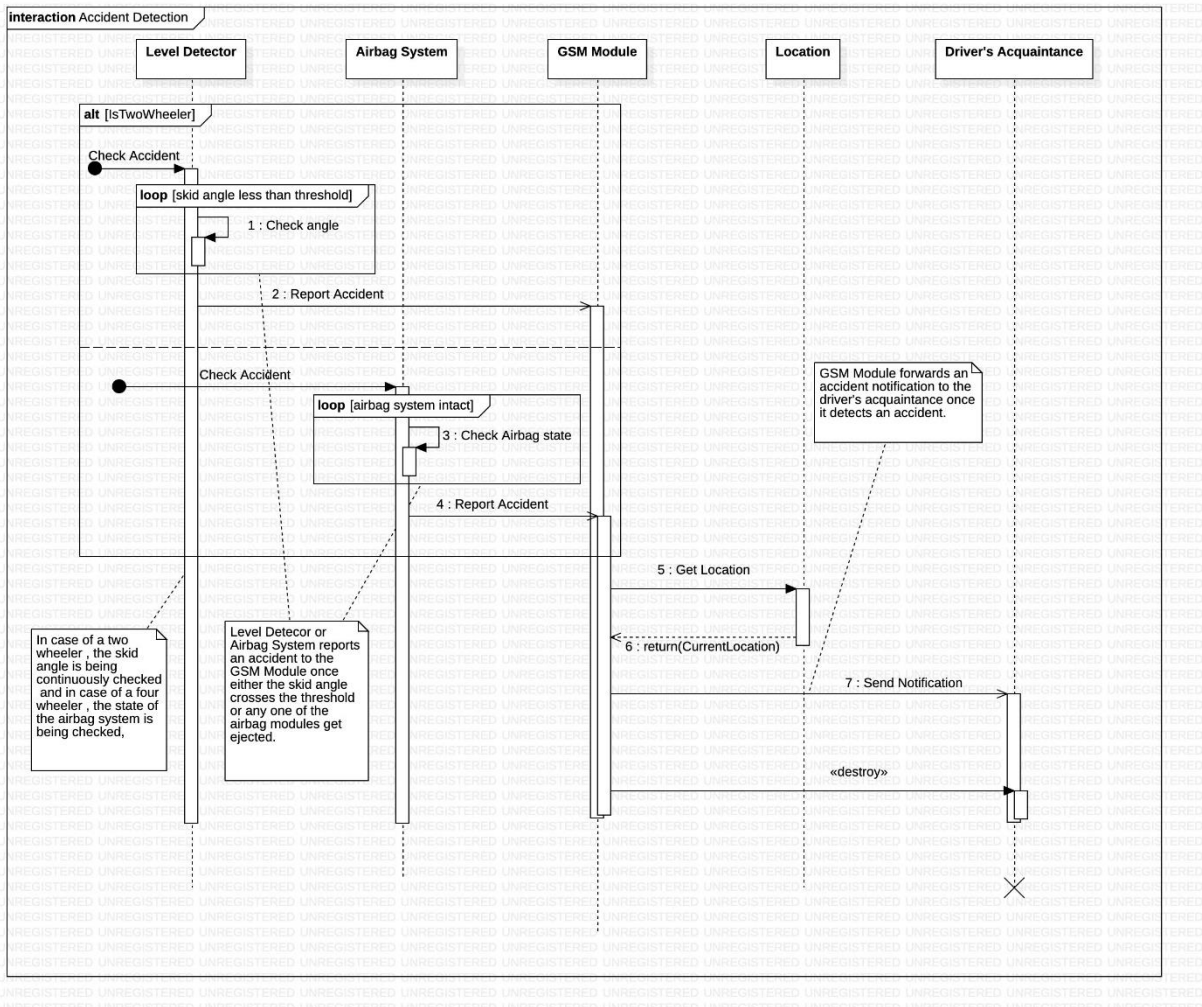
Associated Usecase – Encounter Accident, Detect Location --> **VERSION 1**

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

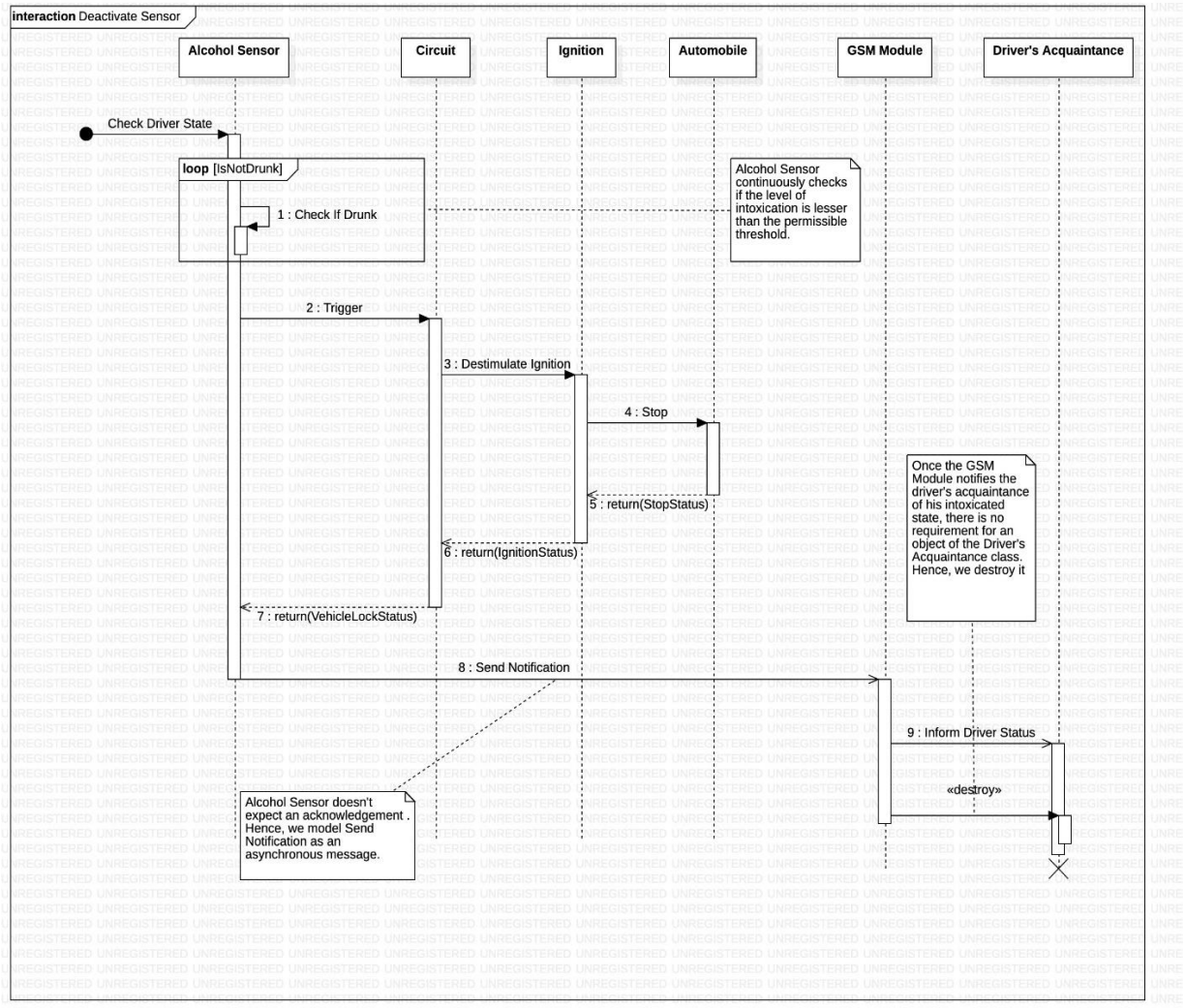
Associated Usecase – Deactivate Sensor, Enable alcohol sensor --> **VERSION 1**

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.



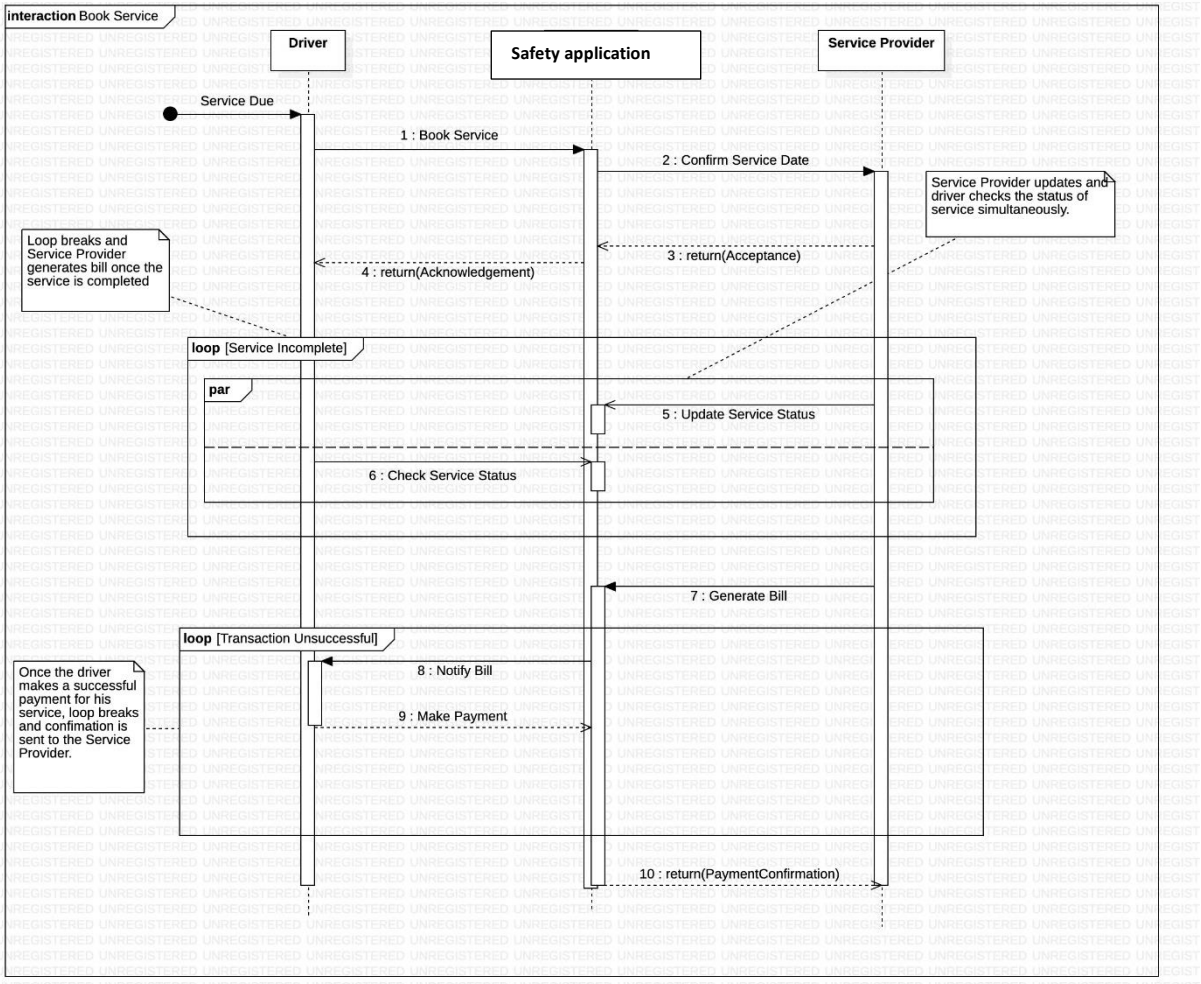
SD5

Associated Usecase – Book Service --> **VERSION 1**

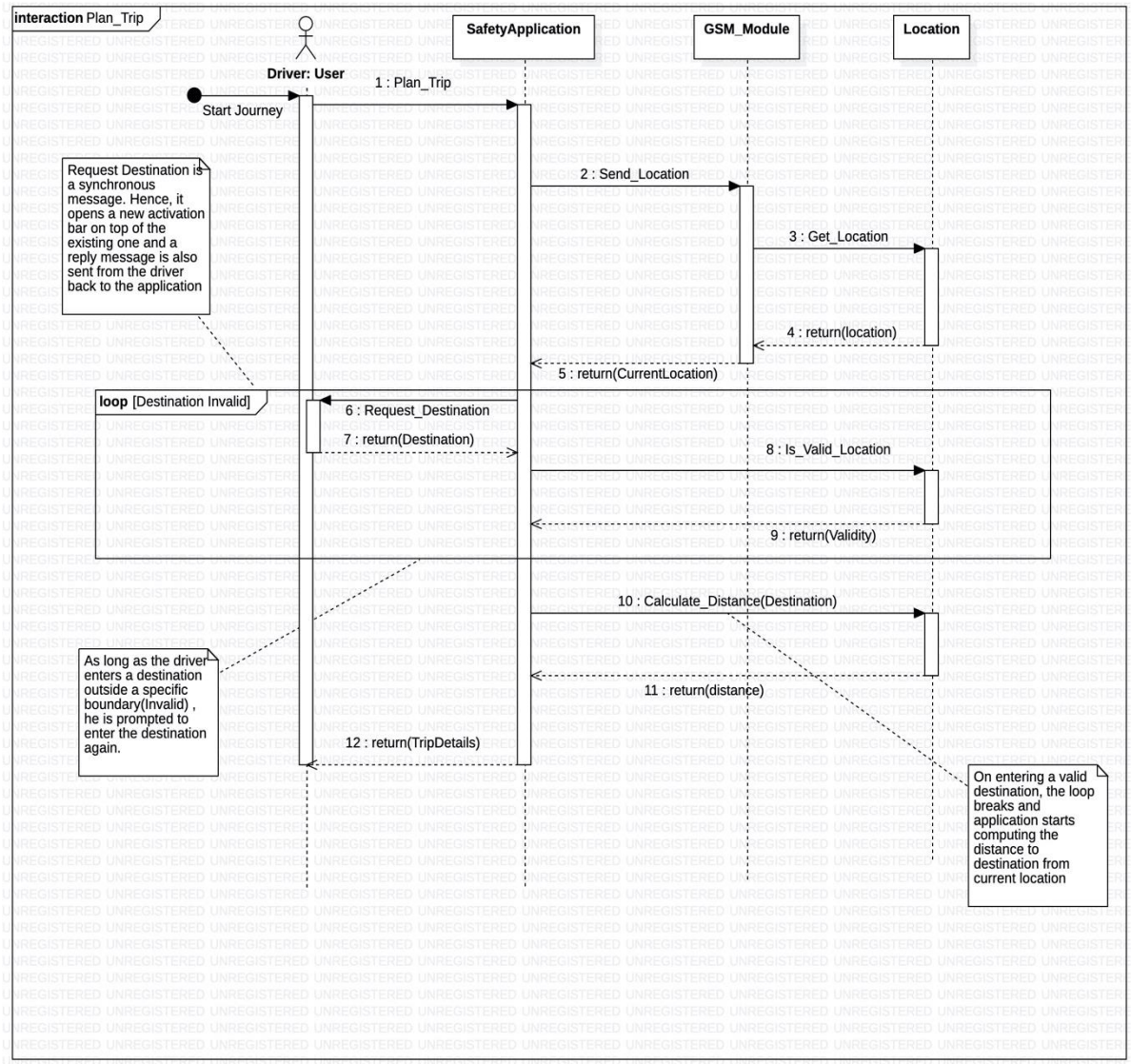
Collaborating classes – Driver, Safety Application, Service Provider

Steps:

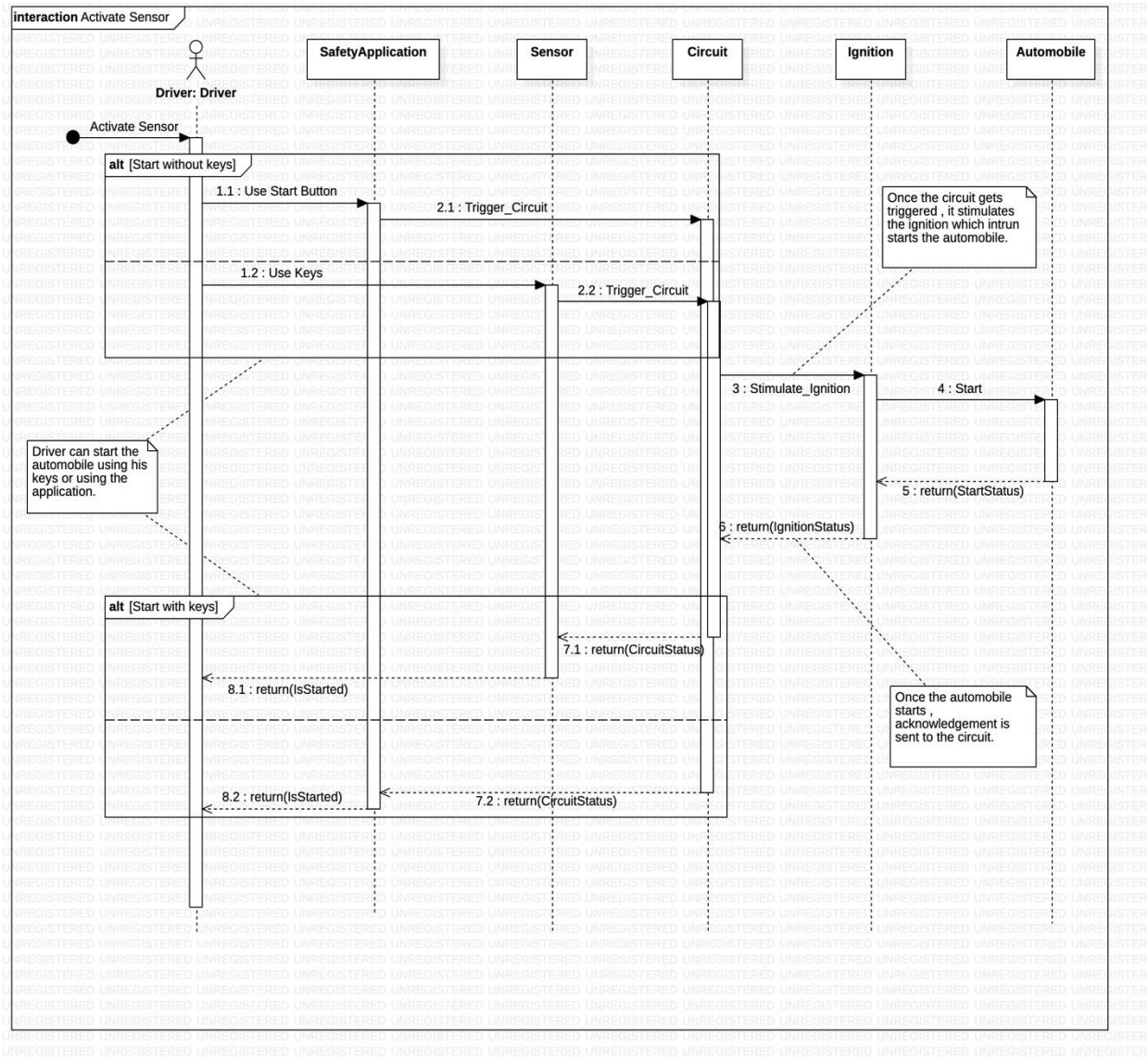
- Driver books service for his vehicle when its on due.
- The safety application class contacts the service provider class to confirm the required service date.
- Driver receives an acknowledgement from the application class, if the service provider returns an acceptance to the safety application class.
- 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.



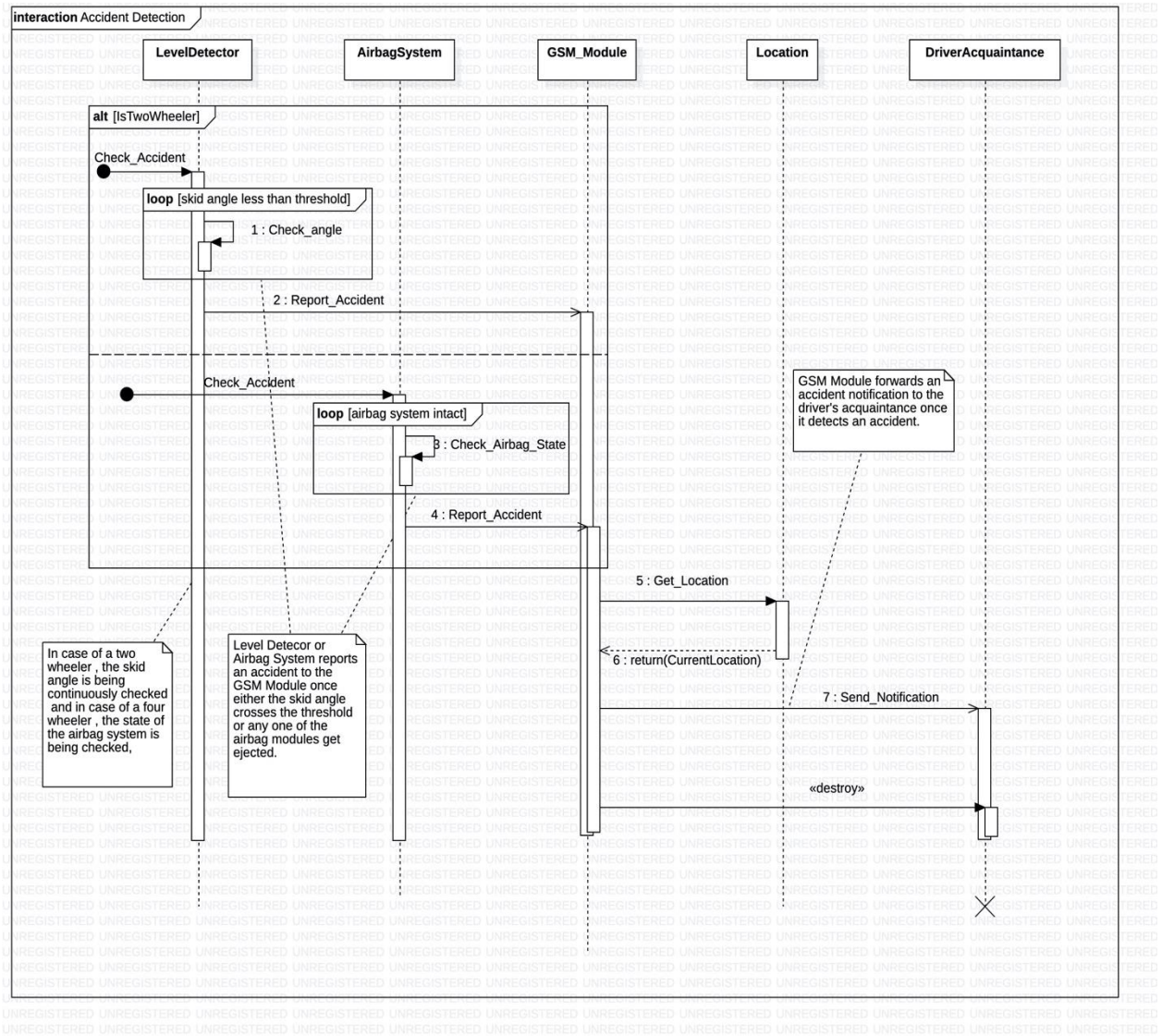
SD1 --> **VERSION 2**



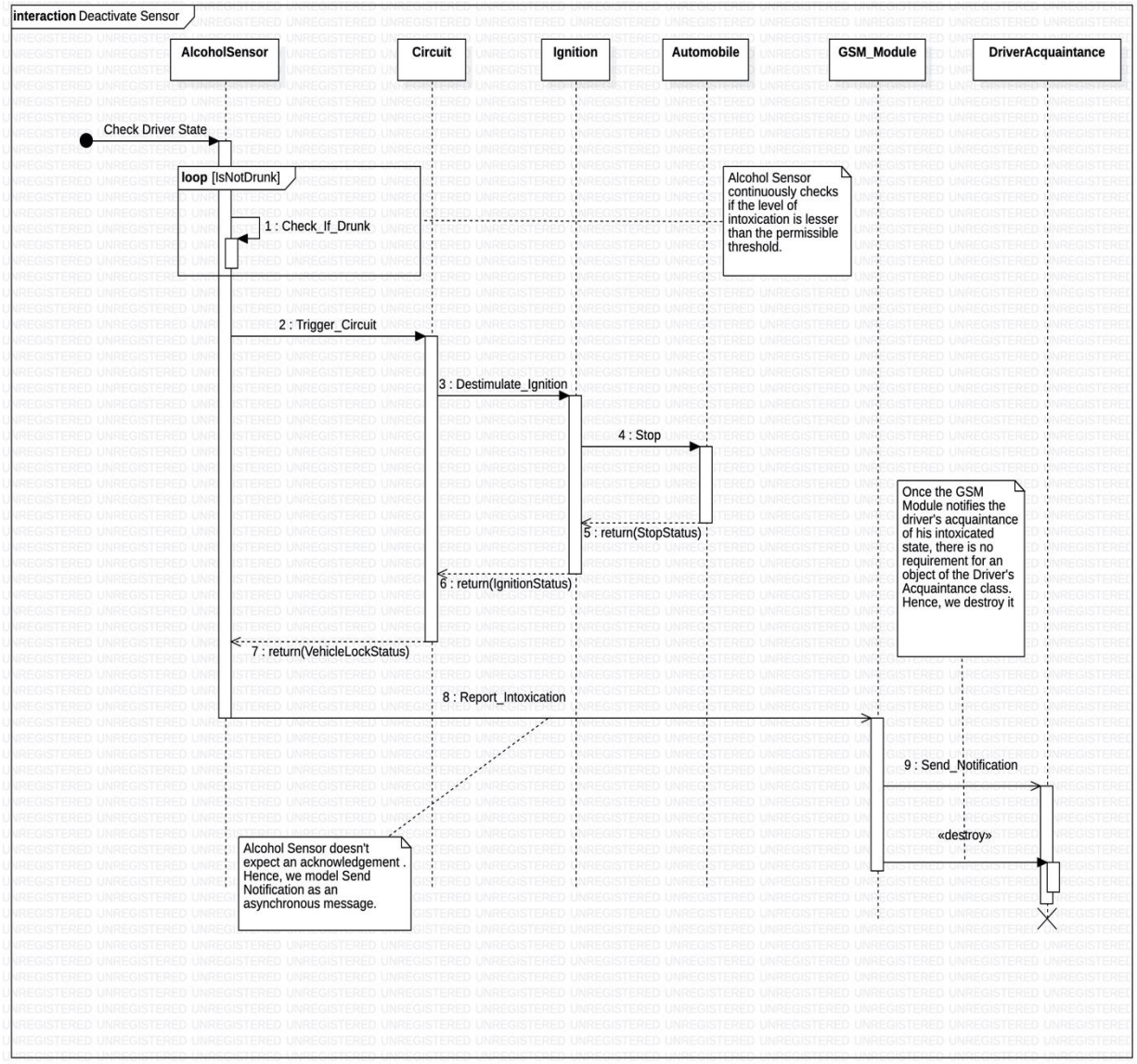
SD2 --> **VERSION 2**



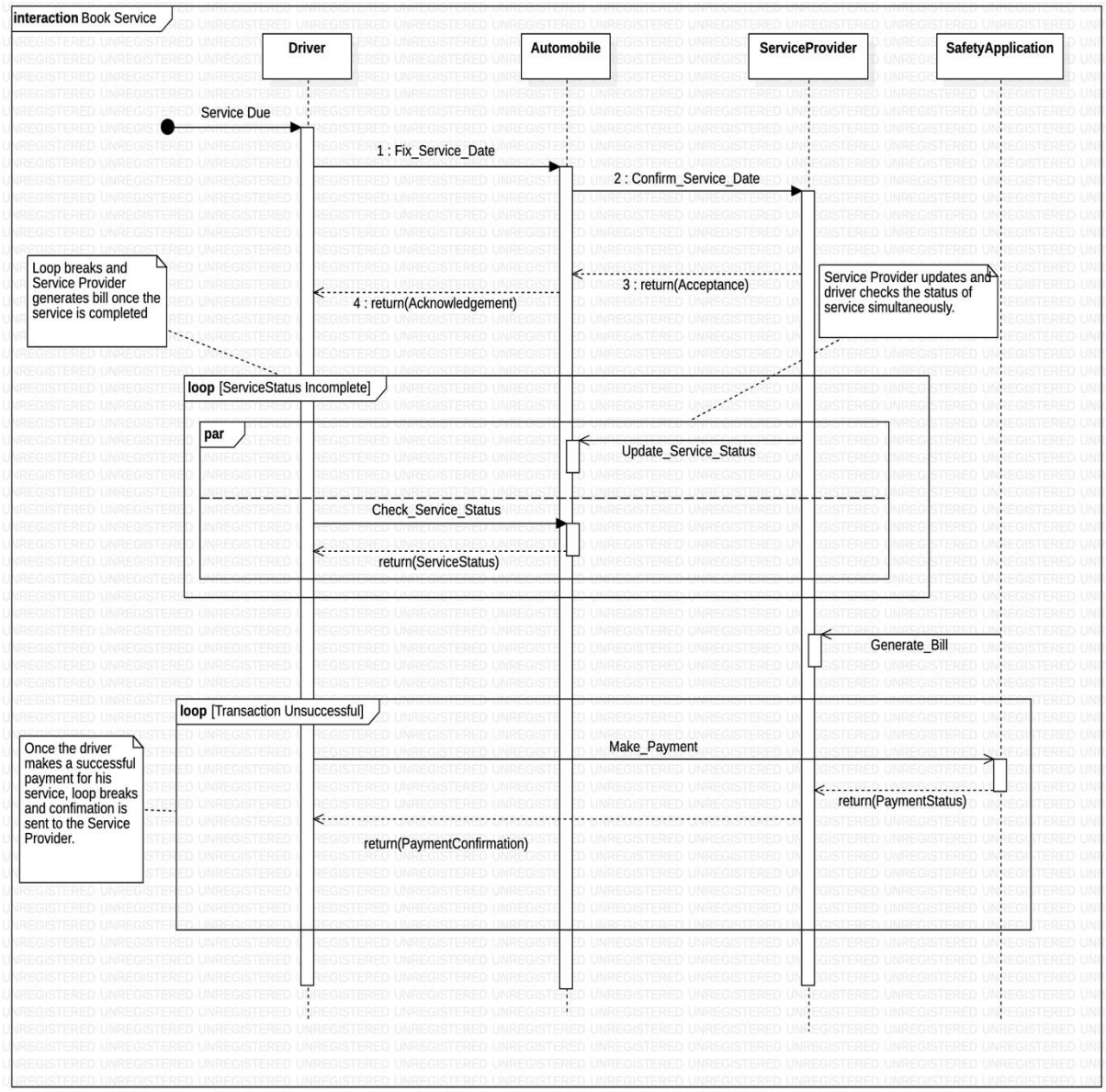
SD3 --> **VERSION 2**



SD4 --> **VERSION 2**



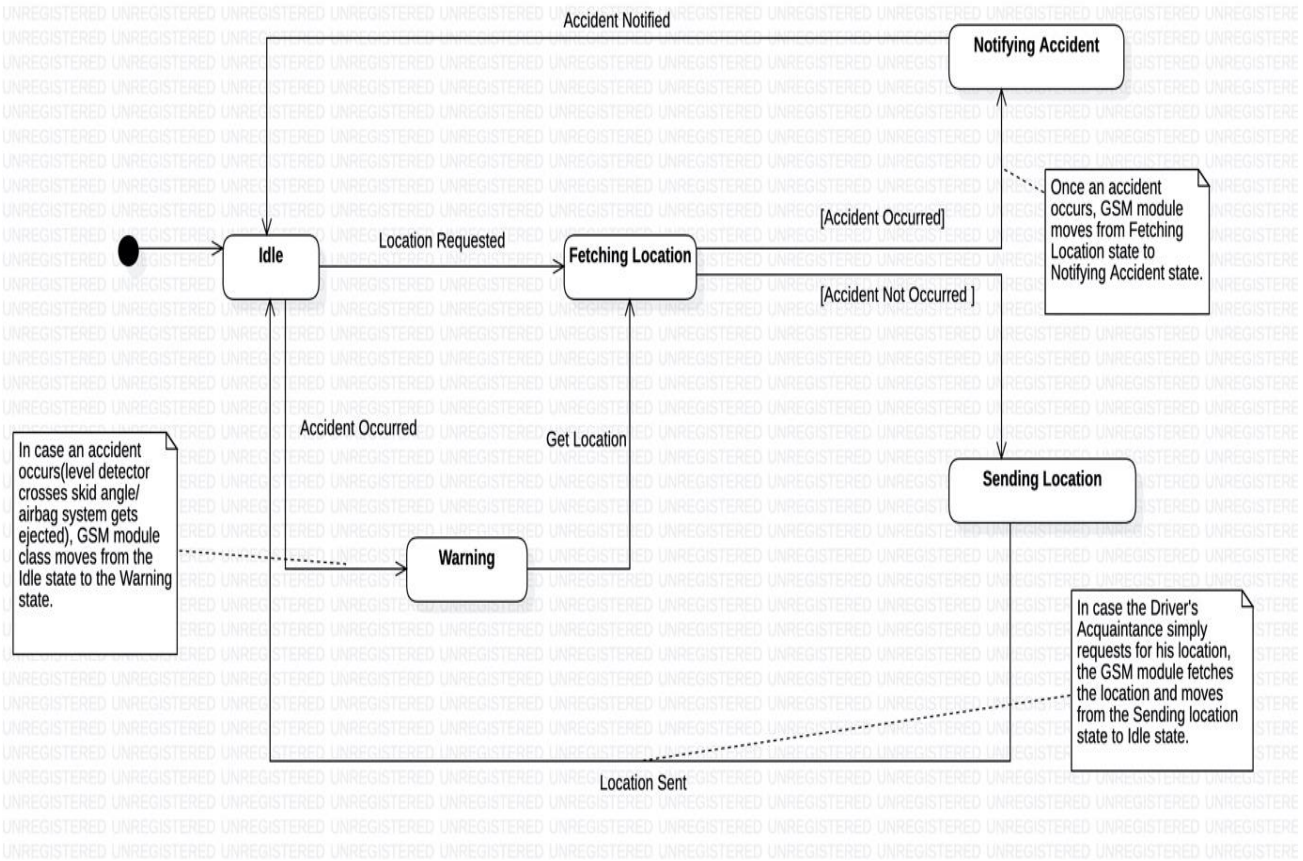
SD5 --> **VERSION 2**



STATE MODELLING

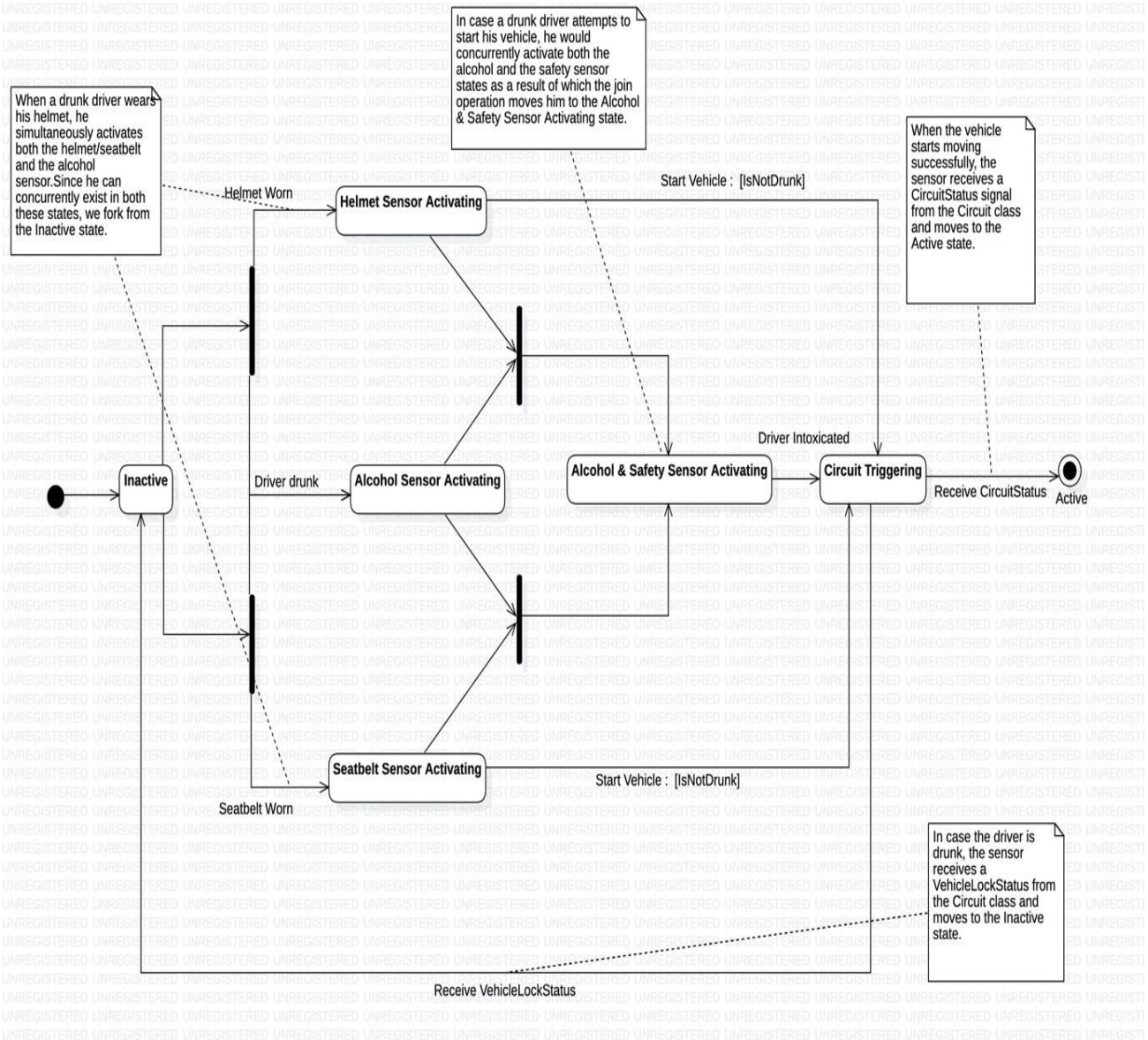
STATE DIAGRAM 1:

- **Associated Class:** GSM Module
- **Scenario:**
  - Driver undergoes an accident.
  - GSM Module moves from **idle** to **warning** state.
  - It fetches the location and moves to the **fetching location** state, post which it makes a transition to the **notifying accident** state.
  - It is in this state where it notifies acquaintances about the accident.
  - Once the GSM Module receives an acknowledgement that the notification has been read, it moves back to the **idle** state.



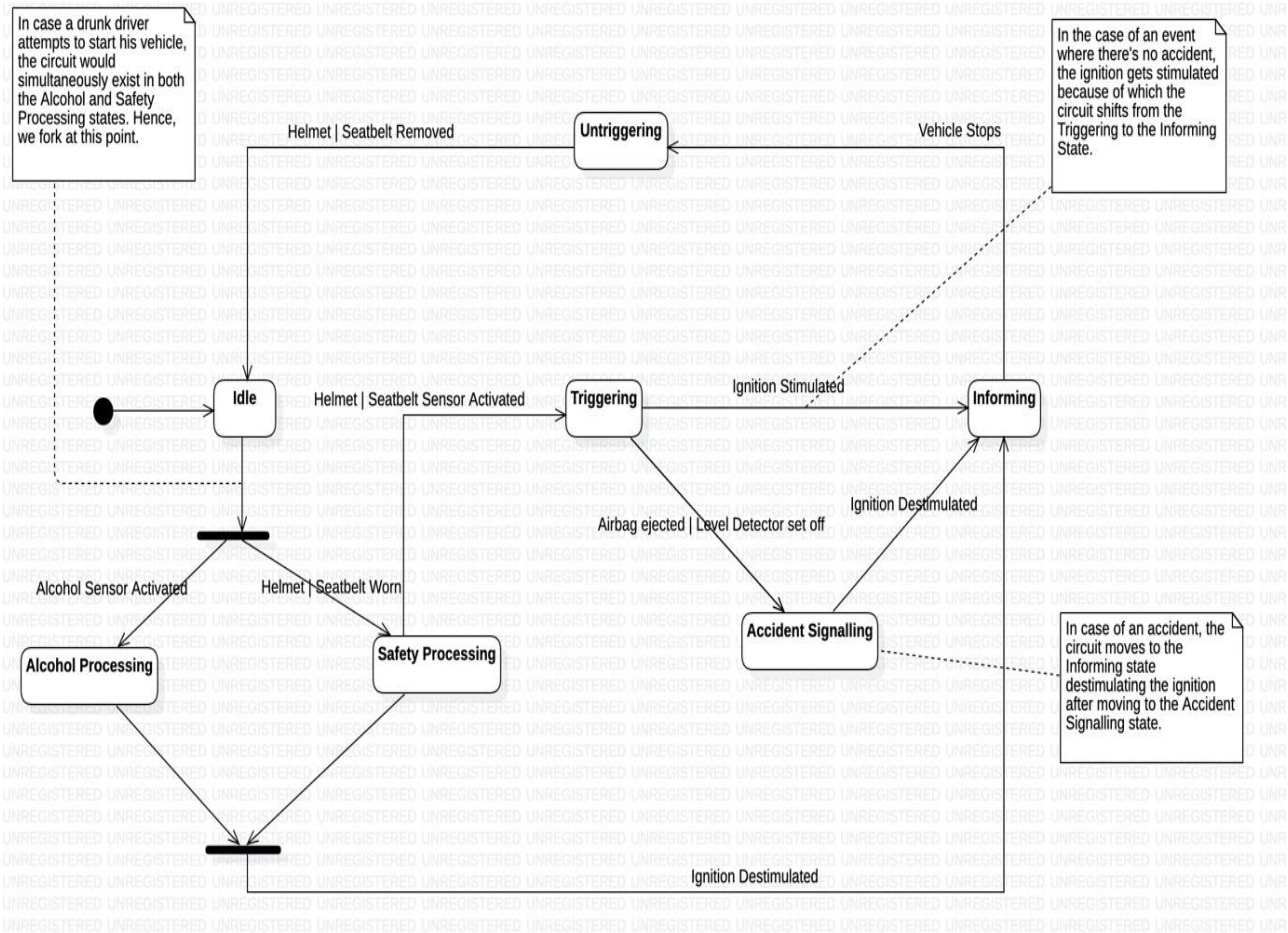
STATE DIAGRAM 2:

- **Associated Class:** Sensor (Helmet Sensor, Seatbelt Sensor, Alcohol Sensor)
- **Scenario:**
  - Driver gets intoxicated due to consumption of alcohol.
  - He wears his seatbelt and attempts to start his car.
  - In this case both the helmet and alcohol sensors get activated because of which sensor forks to these states to concurrently exist in them from the **inactive** state.
  - Due to the join operation, the sensor further shifts to the **alcohol and safety sensor activating** state, post which it transits to the **circuit triggering** state since he is intoxicated.
  - Upon receiving the VehicleLockStatus from the circuit class, sensor moves to the **inactive** state.
  - The vehicle is currently locked and the driver is unable to start it.



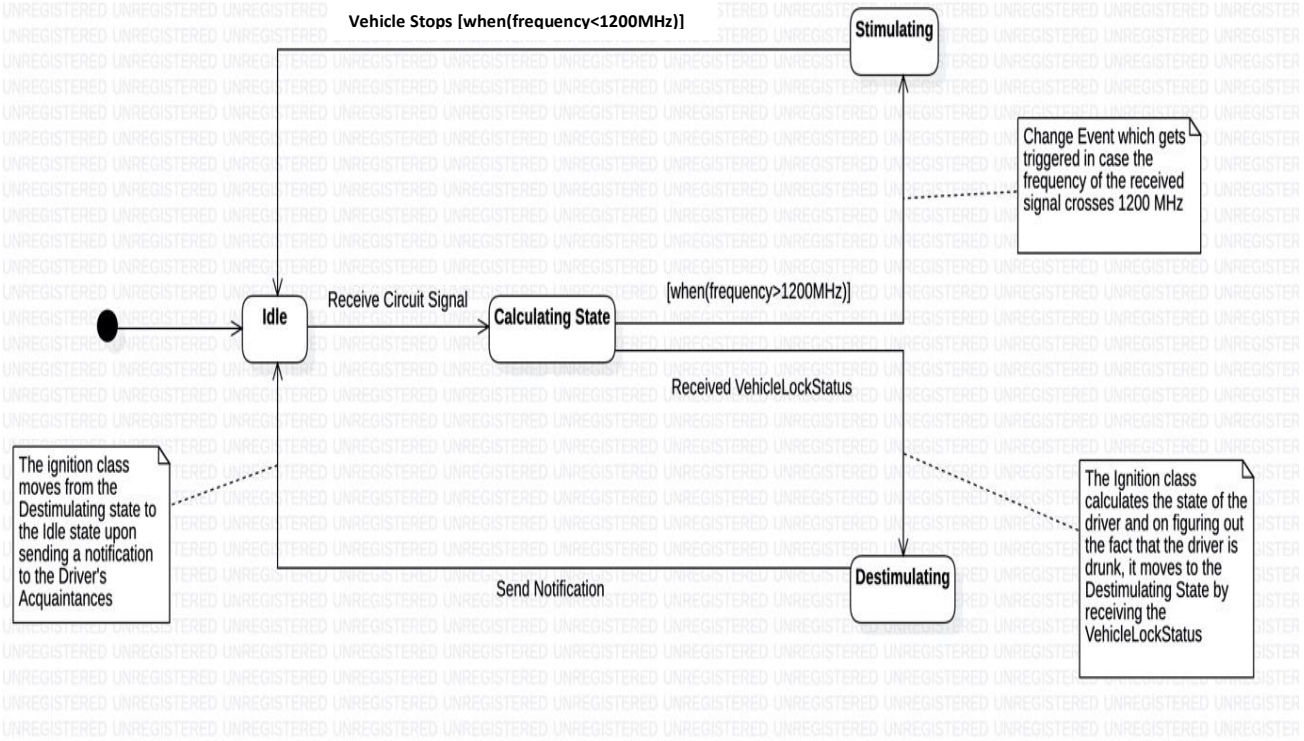
STATE DIAGRAM 3:

- **Associated Class:** Circuit
- **Scenario:**
  - Driver wears his helmet.
  - Circuit moves from **idle** to **safety processing** state.
  - Upon the activation of the helmet sensor, it further shifts to the **triggering** state.
  - The driver then under goes an accident.
  - Now the circuit moves to the **informing** state, destimulating the ignition after moving to the **Accident Signalling** state.
  - As the vehicle stops, it moves to the **Untriggering** state.
  - Upon the deactivation of the helmet sensor by the removal of the helmet, it moves back to the **idle** state.



STATE DIAGRAM 4:

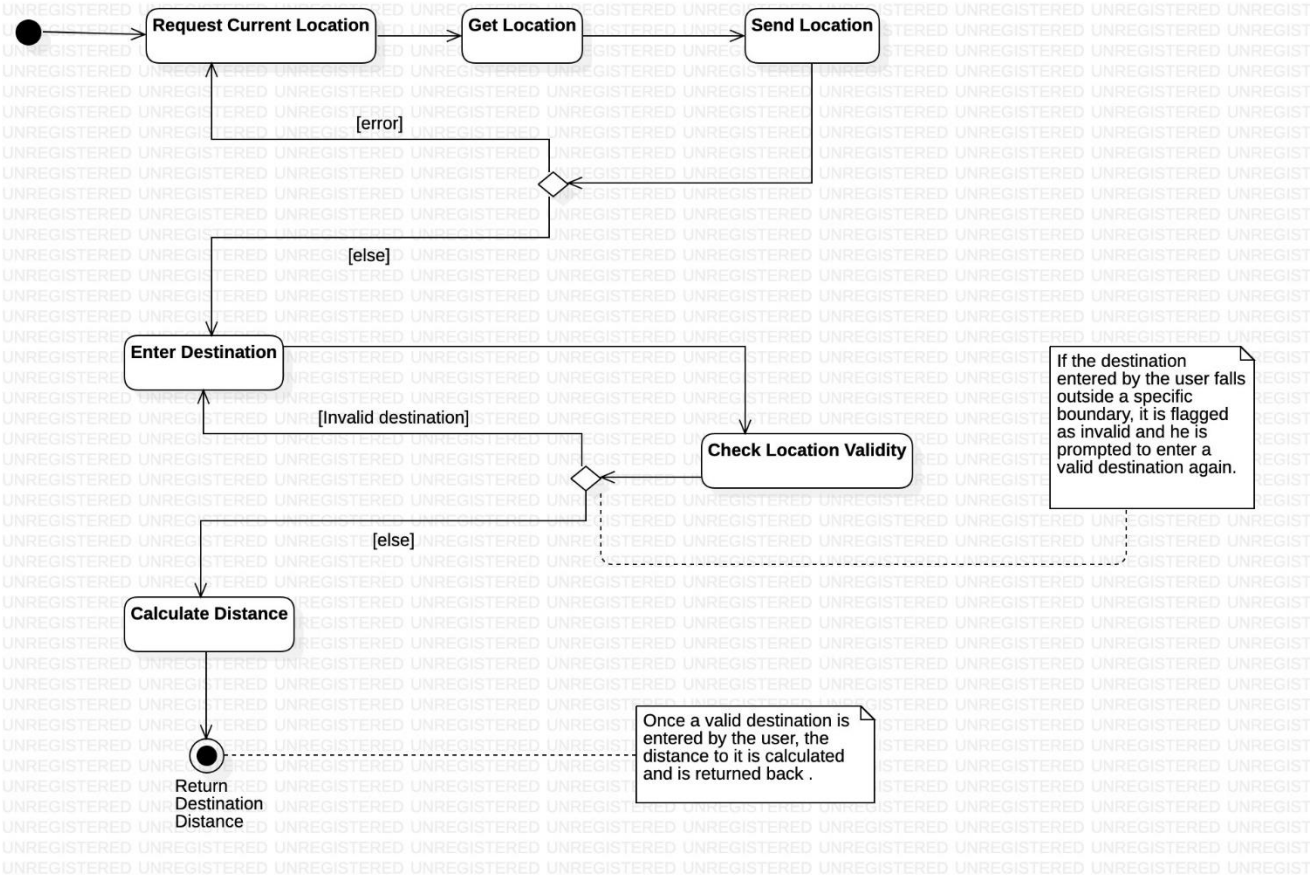
- **Associated Class:** Ignition
- **Scenario:**
  - Driver attempts to start his vehicle.
  - Upon successfully receiving the Circuit signal from the circuit class, the ignition moves from the **idle** to the **Calculating state**.
  - Since the driver isn't drunk and the situation is normal, the ignition receives a signal of frequency 1400MHz from the circuit.
  - As this is greater than the threshold of 1200MHz, a change event gets triggered due to which it moves to the **Stimulating** state.
  - Driver drives successfully and reaches the destination.
  - Upon turning off his vehicle using the keys, the ignition receives a signal of frequency 900MHz from the circuit class (signifying a normal stop).
  - This makes it move back to the **idle** state.



ACTIVITY MODELLING

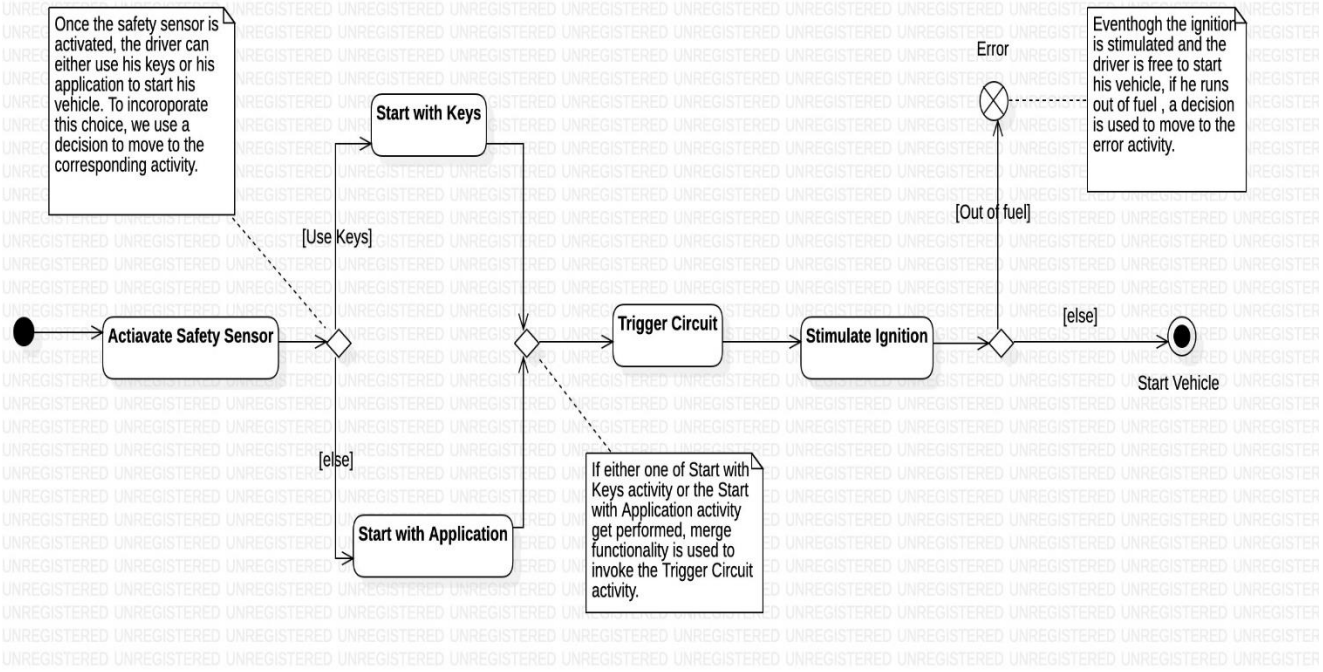
Activity Diagram 1:
Associated Usecase: Plan Trip
Classes Involved: GSM Module, Location, Driver, Safety Application
Scenario:

- To plan the trip the safety application requests the current location of the vehicle from the GSM Module.
- GSM Module in-turn retrieves the location by contacting the location class.
- Current Location is returned back to the safety application.
- 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.



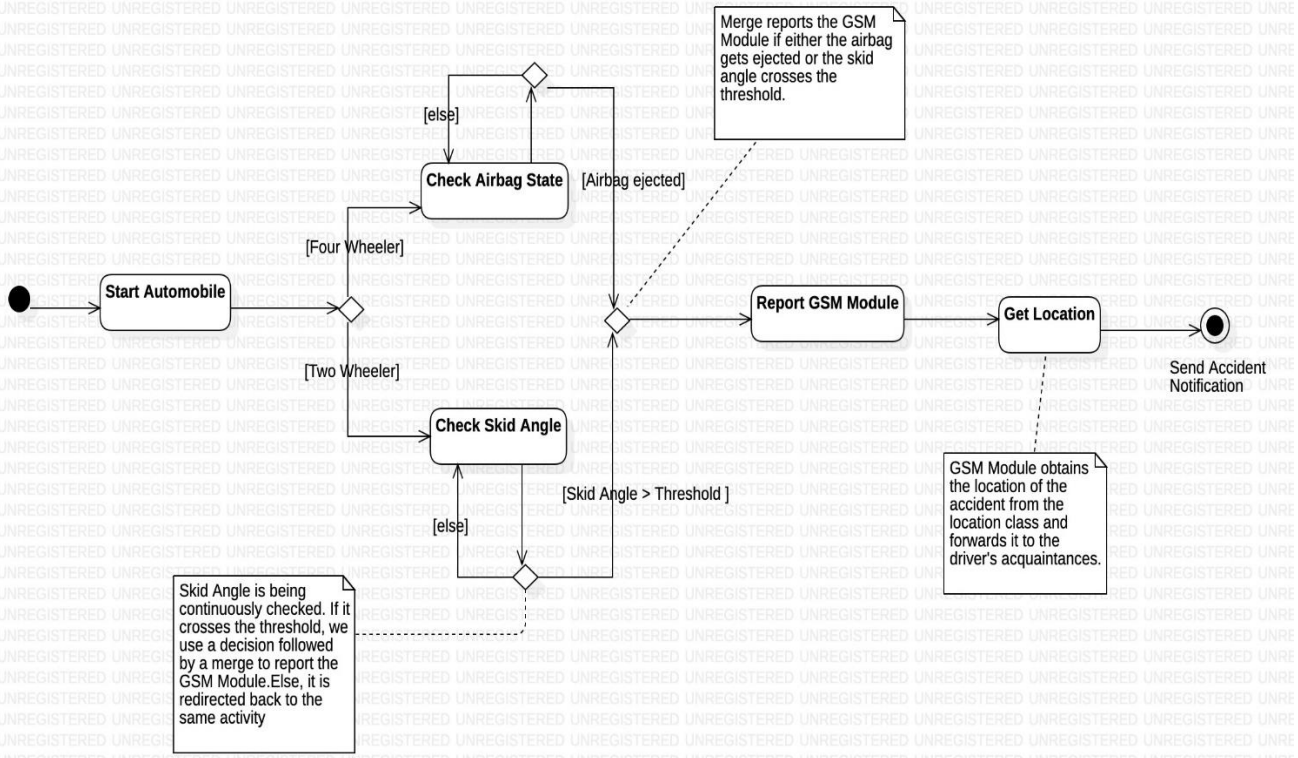
Activity Diagram 2:
Associated Usecase: Start Automobile
Classes Involved: Sensor, Circuit, Ignition, Safety Application
Scenario:

- Driver has two options to start his automobile, i.e. With keys / Using the application.
- For this purpose a decision diamond is being used.
- If either one of **Start with Keys** activity or the **Start with Application** activity get performed, merge functionality is used to invoke the **Trigger Circuit** activity.
- Once the circuit gets triggered, it stimulates ignition which in-turn starts the automobile.
- Once the stimulate ignition activity gets performed, depending on the fuel status of the vehicle the system either performs the **start vehicle** activity or moves to the **error** activity.



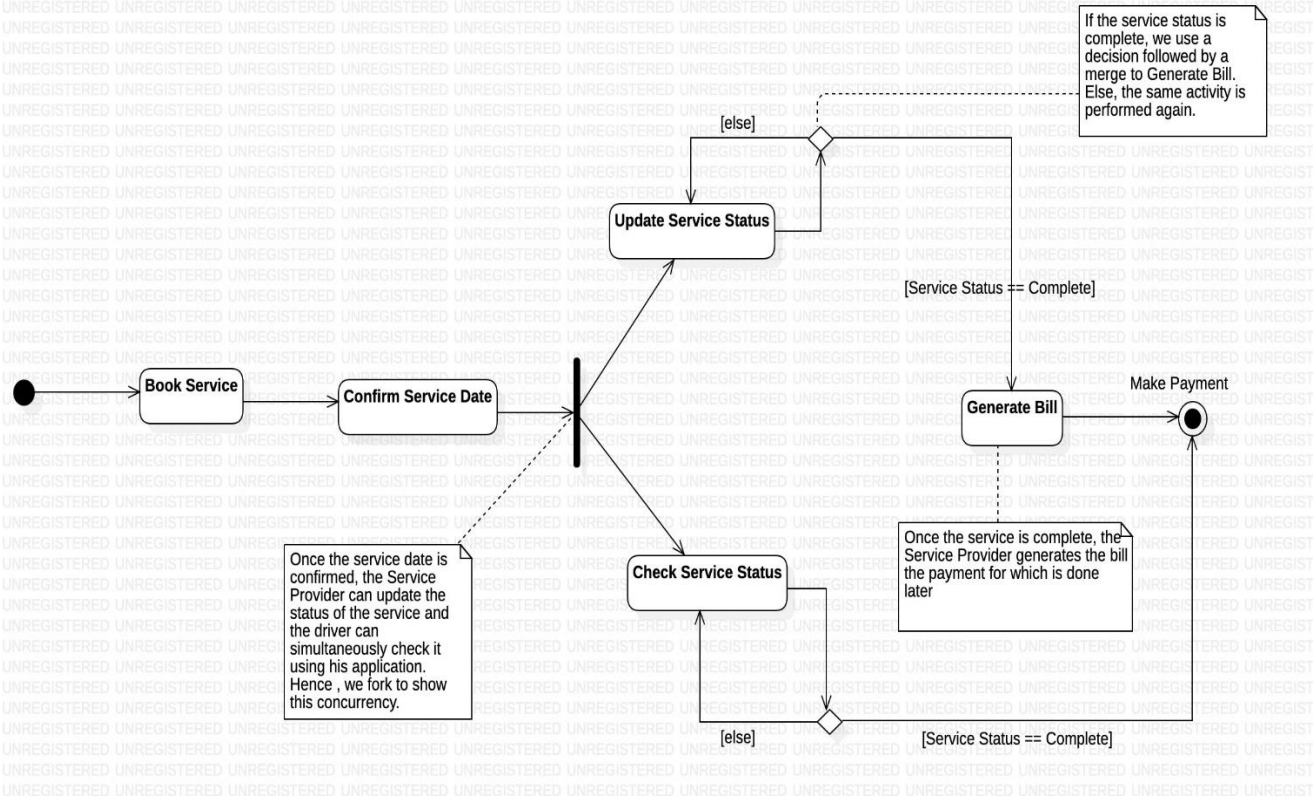
Activity Diagram 3:
Associated Usecase: Send Accident Notification
Classes Involved: Sensor, GSM Module, Level Detector, Airbag System, Automobile
Scenario:

- Once the automobile is started and is on the move, the state of the airbag/skid angle is checked depending on its type. (decision)
- If the Airbag gets ejected or if the skid angle crosses the threshold, a merge is being used to report an accident to the GSM Module.
- If either of the guard condition aren't satisfied, the corresponding activity gets performed repeatedly.
- 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.



Activity Diagram 4:
Associated Usecase: Book Service
Classes Involved: Service Provider, Driver, Safety Application
Scenario:

- Driver books service for his vehicle when its on due.
- The Service Provider confirms the service date with the Driver through the safety application.
- Once the service date is confirmed, the Service Provider can update the status of the service and the driver can simultaneously check it using his application. Hence, we fork to show this concurrency.
- If the service status is complete, we use a decision to perform the **Generate Bill** activity. Else, the same activity is performed again.
- Once the bill is generated, the Driver uses the Safety Application to make the required payment by performing the **Make Payment** activity.



CODE GENERATION

Automobile.java

```
package AutomobilePrudentSystem;

import java.util.*;

public abstract class Automobile {

   public Automobile() {
         Service_Status = false;
   }

   protected String Registration_Number;

   protected String Insurance_ID;

   protected boolean Service_Status;

   public boolean Start() {
      // TODO implement here
      return false;
   }

   public boolean Stop() {
      // TODO implement here
      return false;
   }

   public boolean Fix_Service_Date(String date) {
         ServiceProvider obj1 = new ServiceProvider();
         return COnfirm_Service_Date(date);
   }

   public boolean Check_Service_Status() {
      return Service_Status;
   }


   public void Update_Service_Status(boolean state) {
                  this.Service_Status = state;
   }

}
```

AutomobileDescription.java

```
package AutomobilePrudentSystem;


import java.util.*;

public class AutomobileDescription {

    public AutomobileDescription() {
    }

    public String Manufacturing_Company;

    public String Fuel_Type;

    public int Mileage;

    public int Seat_Count;

}
```

Driver.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class Driver extends User {

    public Driver() {
    }

    private String Driving_Licence_ID;

    private String Blood_Group;

    public ServiceProvider ServiceAssistant;

    public void Send_Emergency_Notification() {
        // TODO implement here
    }

}
```

DriverAcquaintance.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class DriverAcquaintance extends User {

    public DriverAcquaintance() {
    }

    private String Relation;

    private String Blood_Group;

    public Location Detect_Driver_Location() {
        Location obj1 = neew Location();
        return obj1;
    }

    public void Send_Notification(boolean notify) {
                    if(notify == true){
                    System.out.println("Driver has met with an accident. Arrange for emergency");
                    }
                    else{
                            System.out.println("Driver Drunk");
                    }
    }

}
```

FourWheeler.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class FourWheeler extends Automobile {

    public FourWheeler() {
    }

    private int Airbag_modules_count;

    private int Crash_sensor_count;

    private int Cylinder_count;

}
```

TwoWheeler.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class TwoWheeler extends Automobile {

    public TwoWheeler() {
    }

    private float Threshold_Angle;

    private long Chasis_Number;

    private int Cadence;

}
```

SafetyApplication.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class SafetyApplication {

    public SafetyApplication() {
    }
    private String Name;

    private String Version;

    public void Login() {
        // TODO implement here
    }

    public void Update_Details() {
        // TODO implement here
    }

    public void Change_Password() {
        // TODO implement here
    }

    public boolean Make_Payment() {
        // TODO implement here
        return false;
    }
```

```
    public void Plan_Trip() {
                GSM_Module obj1 = new GSM_Module();
                Location CurrentLoc = obj1.Send_Location();

                Driver obj2 = new Driver();
                Location Destination = obj2.Request_Destination();

                while(Destination.Is_Valid_Location() == false){
                        Destination = obj2.Request_Destination();
                }

                float distance = CurrentLoc.Calculate_Distance(Destination);
                System.out.println("Your desired destination is " +distance+" kms away from here");
    }

}
```

ServiceProvider.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class ServiceProvider extends User {

    public ServiceProvider() {
    }

    private String Automobile_Company;

    private long Company_ID_Number;

    private String Service_Centre_Location;

    public Set<Driver> Customer;

    public void Generate_Bill() {
        // TODO implement here
    }

    public boolean Confirm_Service_Date() {
        // TODO implement here
        return false;
    }

}
```

User.java

```
package AutomobilePrudentSystem;

import java.util.*;

public abstract class User {
   static Scanner input = new Scanner(System.in);
    public User() {
    }

    protected String Username;

    protected String Password;

    protected String Name;

    protected long Aadhar_Number;

    protected long Contact_Number;

    public Location Request_Destination() {
            Location Destination = new Location(input.nextFloat(),input.nextFloat());
            return Destination;
    }

}
```

AccidentDetector.java

```
package AutomobilePrudentSystem;

import java.util.*;

public interface AccidentDetector {

    public int Detect_State();

    public void Signal_Circuit();

    public void Check_Accident();

}
```

LevelDetector.java

```
package AutomobilePrudentSystem;


import java.util.*;

public class LevelDetector implements AccidentDetector {


   public LevelDetector() {
         IsDanger = false;
   }
   private float Dimensions;

   private float Skid_Angle;

   private boolean IsDanger;

   private boolean Check_Angle(float Threshold_Angle) {
      if(Skid_Angle < Threshold_Angle)
                  return true;
         return false;
   }

   public int Detect_State() {
      if(IsDanger == true)return 1;
      return 0;
   }

   public void Signal_Circuit() {
      // TODO implement here
   }

   public void Check_Accident(float Threshold_Angle) {
                  while(Check_Angle(Threshold_Angle) == true);
                  IsDanger = true;
                  GSM_Module obj1 = new GSM_Module();
                  obj1.Report_Accident();
   }

}
```

AirbagSystem.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class AirbagSystem implements AccidentDetector {

   public AirbagSystem() {
         isEjected = false;
   }

   private static float Threshold_Torque = 1.8;

   private float Response_Time;

   private boolean isEjected;

   private boolean Check_Airbag_State(torque) {
         if(Torque > Threshold_Torque) isEjected = true;
         else isEjected = false;
      return isEjected;
   }

   public int Detect_State() {
      if(isEjected == true)return 1;
      return 0;
   }

   public void Signal_Circuit() {
      // TODO implement here
   }

   public void Check_Accident(float torque) {
      while(Check_Airbag_State(torque) == true);
            GSM_Module obj1 = new GSM_Module();
                  obj1.Report_Accident();

   }

}
```

AlcoholSensor.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class AlcoholSensor extends Sensor {

   public AlcoholSensor() {
   }

   private float Intoxication_Level;

   private float Active_Temperature;

   private float Optimum_Power;

   private static float Threshold_Intoxication = 0.6;

   public boolean Check_If_Drunk() {
      if(Intoxication_Level < Threshold_Intoxication)
                  return false;
      else{
                  Circuit obj1 = new Circuit();
                  if(obj1.Trigger_Circuit() != 0){
                        GSM_Module obj2 = new GSM_Module();
                        Obj2.Report_Intoxication();
                  }
                  return true;
      }
   }

}
```

Circuit.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class Circuit {

   public Circuit() {
         State = 0;
   }
   private double RF_Received_Signal;
   private int State;
   public void Convert_to_State() {
      // TODO implement here
   }
```

```
    public int Trigger_Circuit() {
            Ignition obj1 = new Ignition();
            if(obj1.Simulate_Ignition() != 0 ||
                            obj1.Destimulate_Ignition() != 1){
                    State = 1;
             }
            else
                    State = 0;
       return State;
    }

}
```

GSM_Module.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class GSM_Module {

    public GSM_Module() {
          Accident_Bit = false;
    }

    private String Standard;

    private long FD_Number;

    private String Embedded_AT_Commands;

    private boolean Accident_Bit;

    public Location Send_Location() {
       Location CurrentLoc = new Location();
            return CurrentLoc.Get_Location();
    }

    public void Report_Accident() {
                    DriverAcquaintance obj2 = new DriverAcquaintance();
                    Accident_Bit = true;
                    Obj2.Send_Notification(Accident_Bit);
    }

    public void Report_Intoxication() {
                    DriverAcquaintance obj1 = new DriverAcquaintance();
                    Accident_Bit = false;
                    Obj1.Send_Notification(Accident_Bit);
    }

}
```

Sensor.java

```
package AutomobilePrudentSystem;

import java.util.*;

public abstract class Sensor {
   public Sensor() {
          State = 0;
   }
   protected int State;

   protected long Model_Number;

   protected String Manufacturing_Company;

   protected float Sensitivity;

   protected float Range;


   public void Activate() {
      State = 1;
   }

   public void Deactivate() {
      State = 0;
   }

}
```

HeartbeatSensor.java

```
package AutomobilePrudentSystem;

import java.util.*;

public class HeartbeatSensor extends Sensor {
   public HeartbeatSensor() {
   }

   private int Pulse_Rate;
   private String Scale;

}
```

HelmetSensor.java

```
package AutomobilePrudentSystem;
import java.util.*;
public class HelmetSensor extends Sensor {
   public HelmetSensor() {
   }
   private float Dimensions;
   private String Arduino_Microcontroller_Version;
}
```

Ignition.java

```
package AutomobilePrudentSystem;
import java.util.*;
public class Ignition {
   public Ignition() {
         State = 0;
   }
   private int State;

   private float Threshold_Electrical_Pulse;
   public int Find_State() {
      return State;
   }

   public boolean Check_Condition() {
      return false;
   }

   public int Stimulate_Ignition() {
          Automobile obj1 = new Automobile();
      if(obj1.Start() == true)       State = 1;
      else          State = 0;
      return State;
   }

   public int Destimulate_Ignition() {
      Automobile obj1 = new Automobile();
      if(obj1.Stop() == true)
                  State = 0;
      else
                  State = 1;
      return State;
   }

}
```

Location.java

```
package AutomobilePrudentSystem;

import java.util.*;
import java.lang.*;
public class Location {

   public Location() {}
   public Location(float lat, float lon) {
                    this.Latitude = lat;
                    this.Longitude = lon;
   }

   private float Latitude, Longitude;

   public Location Get_Location() {
      return this;
   }

   public boolean Is_Valid_Location() {
      if(Latitude < -90 || Latitude > 90 ||  Longitude < -180 || Longitude > 180)
                return false;
           return true;
   }
   public float CalculateDistance(Location Destination) {
      float lon1 = Math.toRadians(this.Longitude);
      float lon2 = Math.toRadians(Destination.Longitude);
      float lat1 = Math.toRadians(this.Latitude);
      float lat2 = Math.toRadians(Destination.Latitude);

      float dlon = lon2 - lon1, dlat = lat2 - lat1;
      double a = Math.pow(Math.sin(dlat / 2), 2) + Math.cos(lat1) * Math.cos(lat2)
            * Math.pow(Math.sin(dlon / 2),2);

      float c = 2 * Math.asin(Math.sqrt(a));

      // Radius of earth in kilometers.
      float r = 6371;

      return(c * r);
   }
}
```

Signal.java

```
package AutomobilePrudentSystem;
import java.util.*;

public class Signal {
  public Signal() {
   }
   private float Strength;
   private float Frequency;
}
```

REFERENCES:

- Dramatic Increase in Road Accidents https://www.prb.org/roadtrafficaccidentsincreasedramaticallyworldwide/
- M. K. A. Mohd Rasli, N. K. Madzhi and J. Johari, "Smart helmet with sensors for accident prevention," 2013 International Conference on Electrical, Electronics and System Engineering (ICEESE), Kuala Lumpur, 2013, pp. 21- 26, doi: 10.1109/ICEESE.2013.6895036