System Analysis and Project Management

Project Work

IntelliCAR

The Ultimate IoT Car Care

Authored By

Divya Jayaprakash (dxj160830)
Gautami Murugan(gxm161330)
Ravi Vanam(rxv160430)
Ashwin Kannan (axk162031)
Preeti Natarajan(pxn162030)

Expert Guidance:
Dr. Srinivasan Raghunathan
MIS 6308.0W1 - SAPM - Summer 2017

TABLE OF CONTENTS

1. Executive Summary:

In the modern world, the connection between technology and people exists pretty much all the time. Technology being an absolute need, it has an answer for mankind's problems. The aim of technology is aiming for comfort of use in whatever form it is. The purpose of technology is directed towards easiness in life.

In this project, we are presenting a new mobile application, "IntelliCar", a mobile application that comprises of features that provides more humanistic and intelligent services to a car driver than just automation. **The mobile application includes features like Sharing Location Notification once the car is parked, Remote engine ignition, automatic garage door opener, Cardless fuel payment, Lights/AC control all through mobile phone of the car owner which is integrated with the car processor**. The processor will receive signals from the car owner's mobile application and act accordingly based on the user's command.

They hold a great deal of promise with safety, in saving time and expenditure thus making life easier for people. The faster the world is, the lesser the time we have for our self which are to be spent on smaller things. Technology plays a humongous role in helping save people's time with things that require sufficient bit of human efforts by automating it. Intelligence in cars also brings more scope for advancement in the future, it saves people's time and therefore becomes efficient and effective.

Finding a parked car in a huge parking lot, garage door opener, turning the lights of the car on when needed from outside are some of the things that needs artificial intelligence. This approach with the help of technology will significantly reduce time and efforts consumed on these activities. The biggest benefit of the feature is that the technology caters comfort and safety. The fact that the importance of technology is undeniable, improvements from time to time through new devices and applications are made just to make our lives easier. This project is aimed at providing scalable time effective solutions along with safety and comfort to the car owners.

2. Problem Statement:

The existing system lacks some of the useful features, which if incorporated can benefit the car owners. These useful features can add value to the existing system.

1. Remote Engine Ignition:

After a cold night, it can be hard to leave the warm embrace of your home and venture into a frigid car. The engine will almost be dead and chokes for about 10 minutes. It is much more comfortable and useful if there is a chance to turn on the car's ignition from a distance a few minutes before someone enters a frigid car.

2. Receiving location as a google map notification after the car is parked:

It can be difficult to remember the location of a car in a parking garage or giant parking lot. It helps if there is an aid which helps remembering where the car is parked.

3. Card less fuel payment using IntelliCAR mobile app, at any fuel station across the country:

Transforming from the cashless payments to the era of card less payments, this method can be made easier for the car owners and there is also a need to make sure fraud is not occurring at the pumps.

4. Voice indication of the road signs all through the drive:

Road signs carry important information necessary for driving – current traffic situation, right of way, prohibit or permit certain directions and navigation. There is indeed an added safety measure to understand the neighborhood, and so enable a support system to assist the driver with all road signs through voice indication.

5. Phone Beep if trunk or door open for a long time:

Open door or trunk can be dangerous on the road and, if the door/trunk is not properly closed after parking in the garage or huge parking. A convenient way to double check if the door is open for a long time is important and would be a feature for convenience.

3. Project Name: IntelliCAR - an ultimate IOT car care with mobile device

4. Objectives

- Remote engine ignition
- Receiving location as a notification after the car is parked
- Card less fuel payment, at any fuel station across the country
- Voice indication of the road signs all through the drive
- Garage door opener
- Remote switching on/off of lights/AC
- A car icon which shows if the doors, trunk are opened and lights on
- Phone Beep if trunk or door open for a long time

5. Expected Value and Scope:

The following is the idea for a new application. “IntelliCar” is a mobile application that comprises of features that provides more humanistic and intelligent services to a car driver than just automation.

It provides the following facilities to the car owner:

1. **Remote Engine Ignition:**
   It lets the car turn on the ignition from a distance, so that it can linger a few more minutes inside while the car warms up. It is a convenient setup to have, especially throughout the winter.

2. **Receiving location as a notification after the car is parked:**
   The idea is that if the owner parks the car in a huge parking lot somewhere and not have to remember where he parked, this application comes in handy to identify the parked location. It thereby saves a lot of time in searching the parked vehicle.

3. **Card less fuel payment, at any fuel station across the country:**
   To make the payment at a fuel station easier for the driver, the card less fuel payment. The possibility of eliminating cards is a big convenience for anyone and it also reduces the chances of fuel theft.

4. **Voice indication of the road signs all through the drive:**
   Since road sign play and important role in road safety, automatic recognition of road signs is an implicit necessity. Lack of concentration or ignorance can cause of lot damage; this feature makes sure it rules out the possible threats to daily life that can be formed due to lack of concentration or ignorance.

5. **Phone Beep if trunk or door open for a long time:**

This feature helps to double check if the door is open for a long time is important and would be a feature for convenience

Apart from these features, the application offers other features like:

- Remote Garage door opener
- Remote AC/Lights control
- Car Icon which shows if the doors, trunk are opened and lights on
- These features add to the convenience and safety of every car owner and makes life easier for them

6. Data Collection:

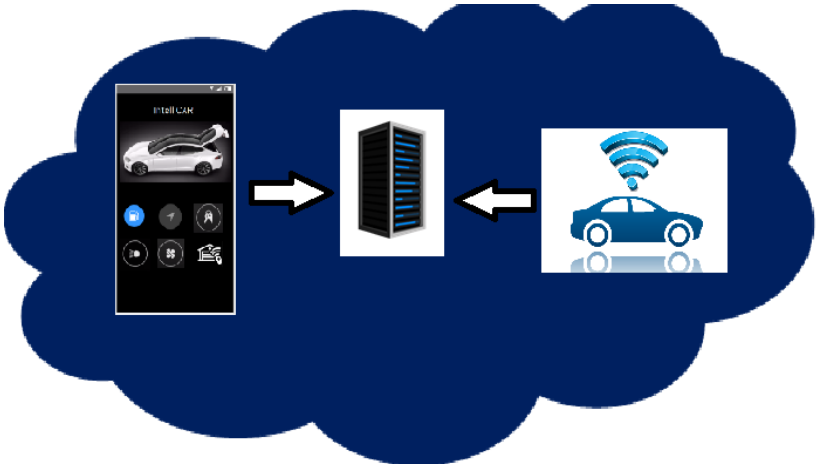
Since the concept is technical, we had to check the feasibility of wireless network with extensive range. The next point was to check the load balance. Several million cars across country are going to pour in requests and need to be served at a time.

We thoroughly researched on the internet and spoke with electrical engineers (subject matter experts) to confirm the feasibility. We eventually gathered the below information.

Nodes relay information across the network. Our various mobile phone and car master processor could operate as those nodes. ZigBee is used for personal area network, which is not our scope. Since we need a higher range of local area network, we choose to communicate between the mobile phone, server and the car master processor via wifi.

7. Proposed Solution:

Our ultimate conclusion after data collection was to design the IntelliCAR application purely based on Internet of Things (IOT). IntelliCAR Application runs on the remote server. Every mobile device and its connected car communicates to the server through wifi. Requests are handled by the server.



In the era of smart devices, there are ready to use protocols to make devices poll and talk continuously. With Internet, distance between the object is not considered as the parameter at all. Only the speed of communication matters.

8. Feasibility:

Technical Feasibility: Internet of Things (IOT) and mobile application development

Economic Feasibility: Application development, Server maintenance and Customer Service Cost is feasible

Organizational and cultural feasibility:
If the product is launched, it will definitely be welcomed by the car users

9. FURPS Requirements:

Functional Requirements: IntelliCAR mobile application, Mobile Device, Wifi, connected car

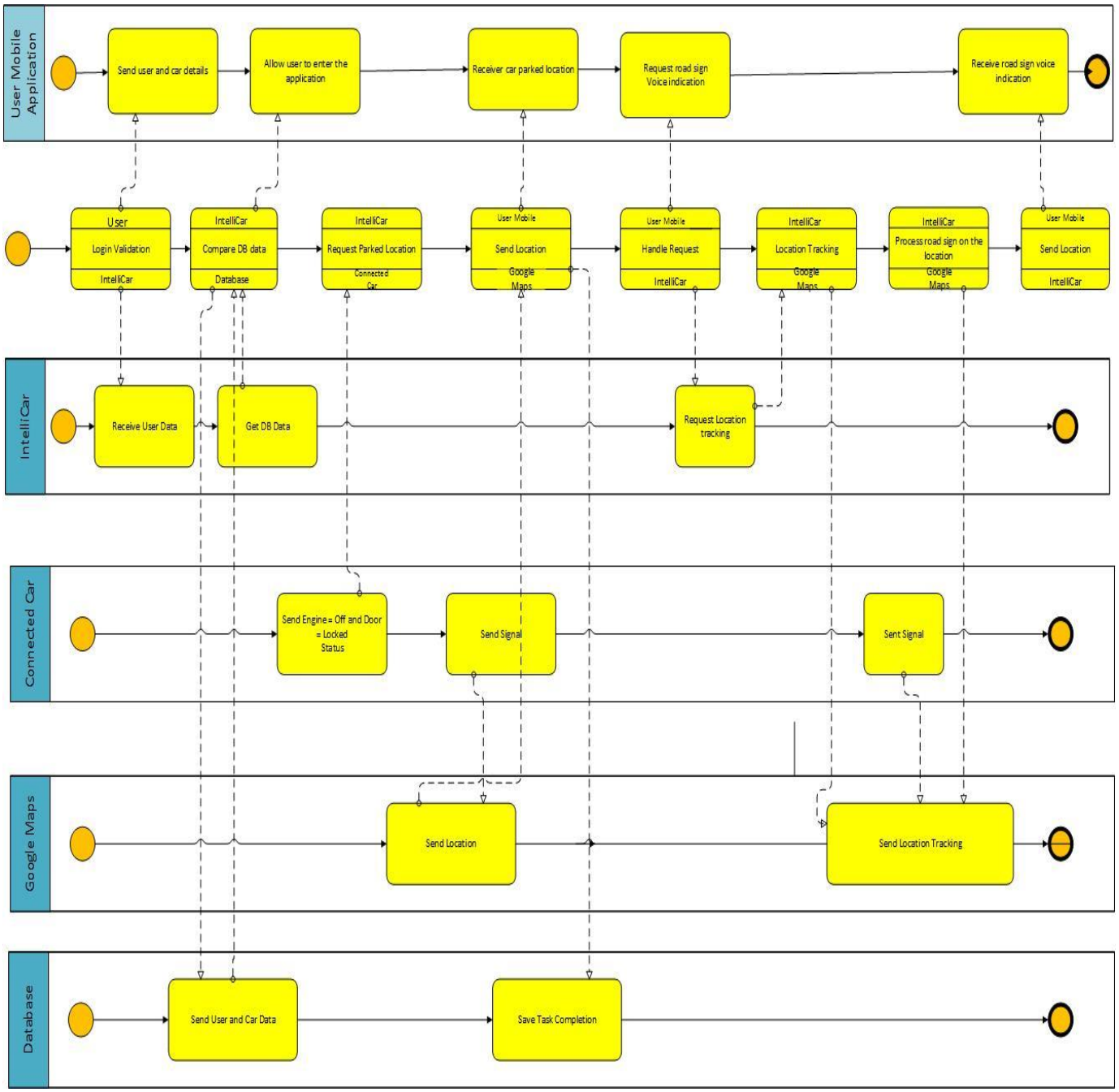
Usability Requirements: Skilled at mobile app usage, user registration

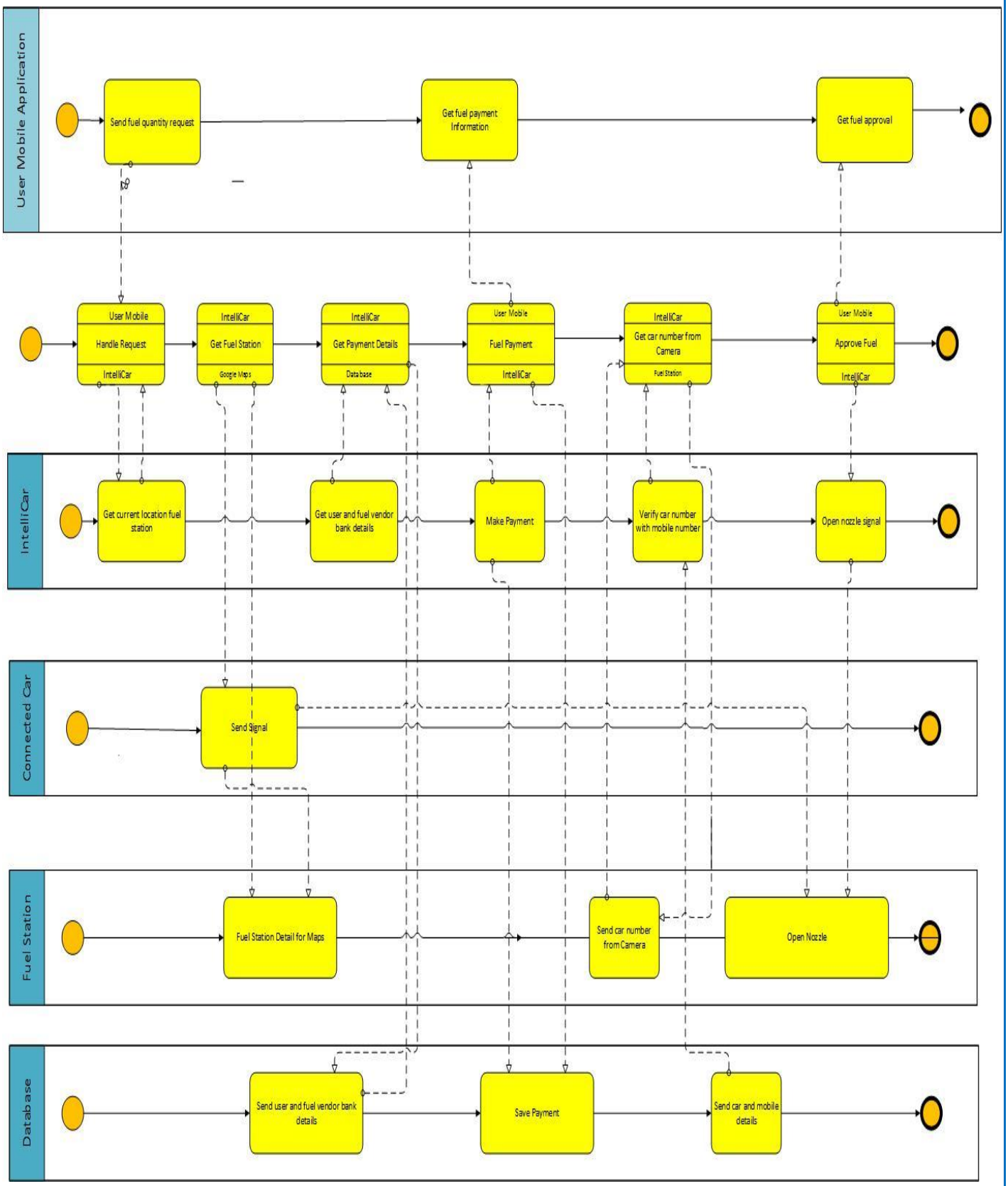
Reliability requirement: Continuous data connectivity in the mobile, Fail safe server & database

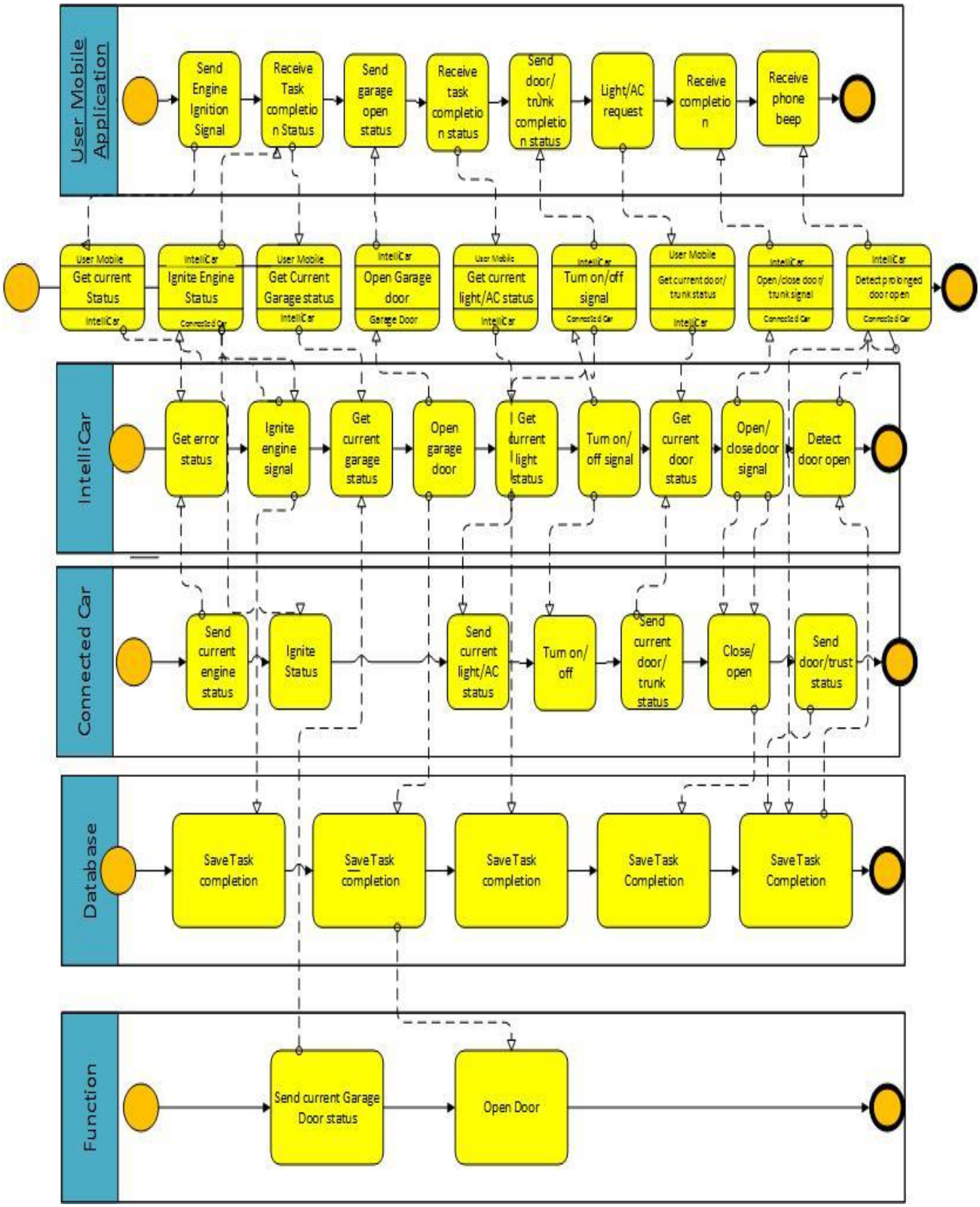
Performance requirements: Network bandwidth, Database capacity, Server load balancing

Security requirements: Internet Security, secured login, secured payment gateway

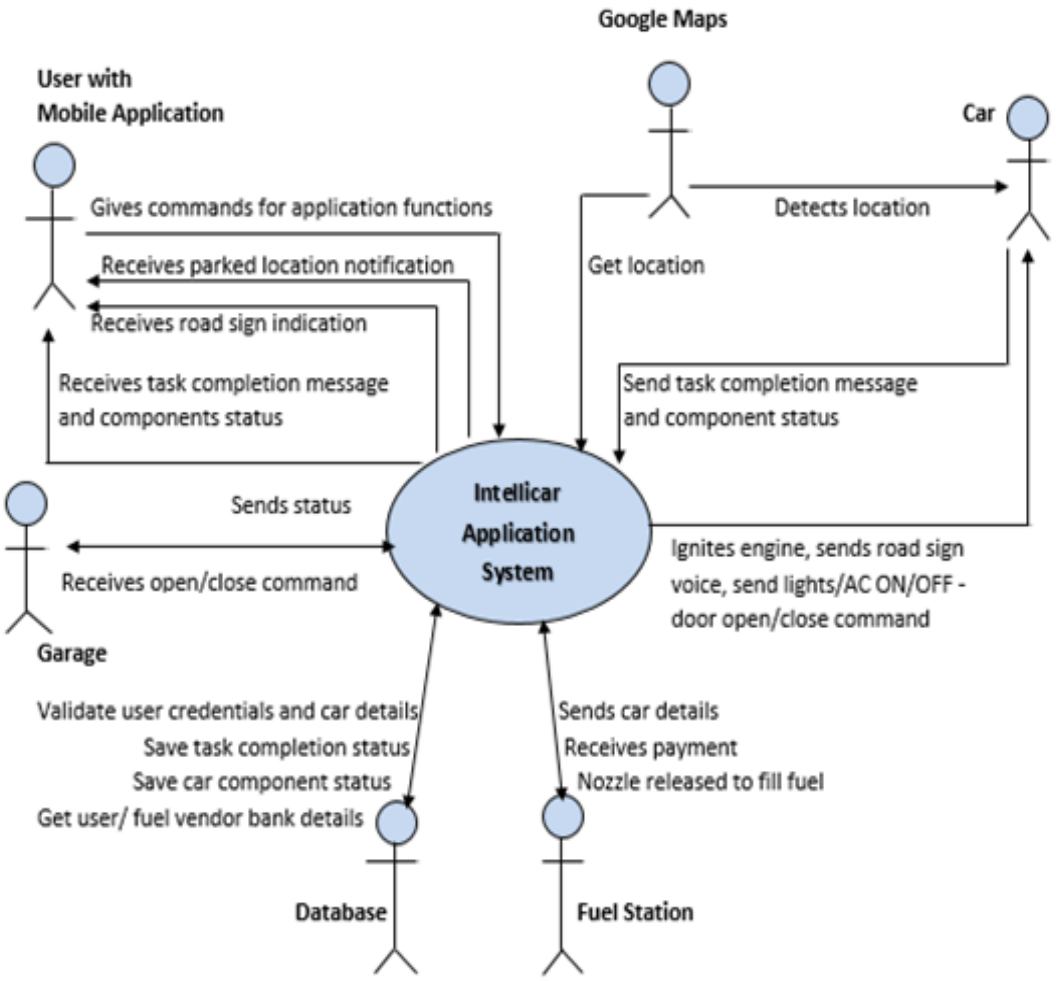
10. Business Process Model Notation (BPMN): (choreography with pools)




User Mobile Application
Send fuel quantity request
Get fuel payment Information
Get fuel approval
User Mobile
Handle Request
IntelliCar
IntelliCar
Get Fuel Station
Google Maps
IntelliCar
Get Payment Details
Database
User Mobile
Fuel Payment
IntelliCar
IntelliCar
Get car number from Camera
Fuel Station
User Mobile
Approve Fuel
IntelliCar
IntelliCar
Get current location fuel station
Get user and fuel vendor bank details
Make Payment
Verify car number with mobile number
Open nozzle signal
Connected Car
Send Signal
Fuel Station
Fuel Station Detail for Maps
Send car number from Camera
Open Nozzle
Database
Send user and fuel vendor bank details
Save Payment
Send car and mobile details

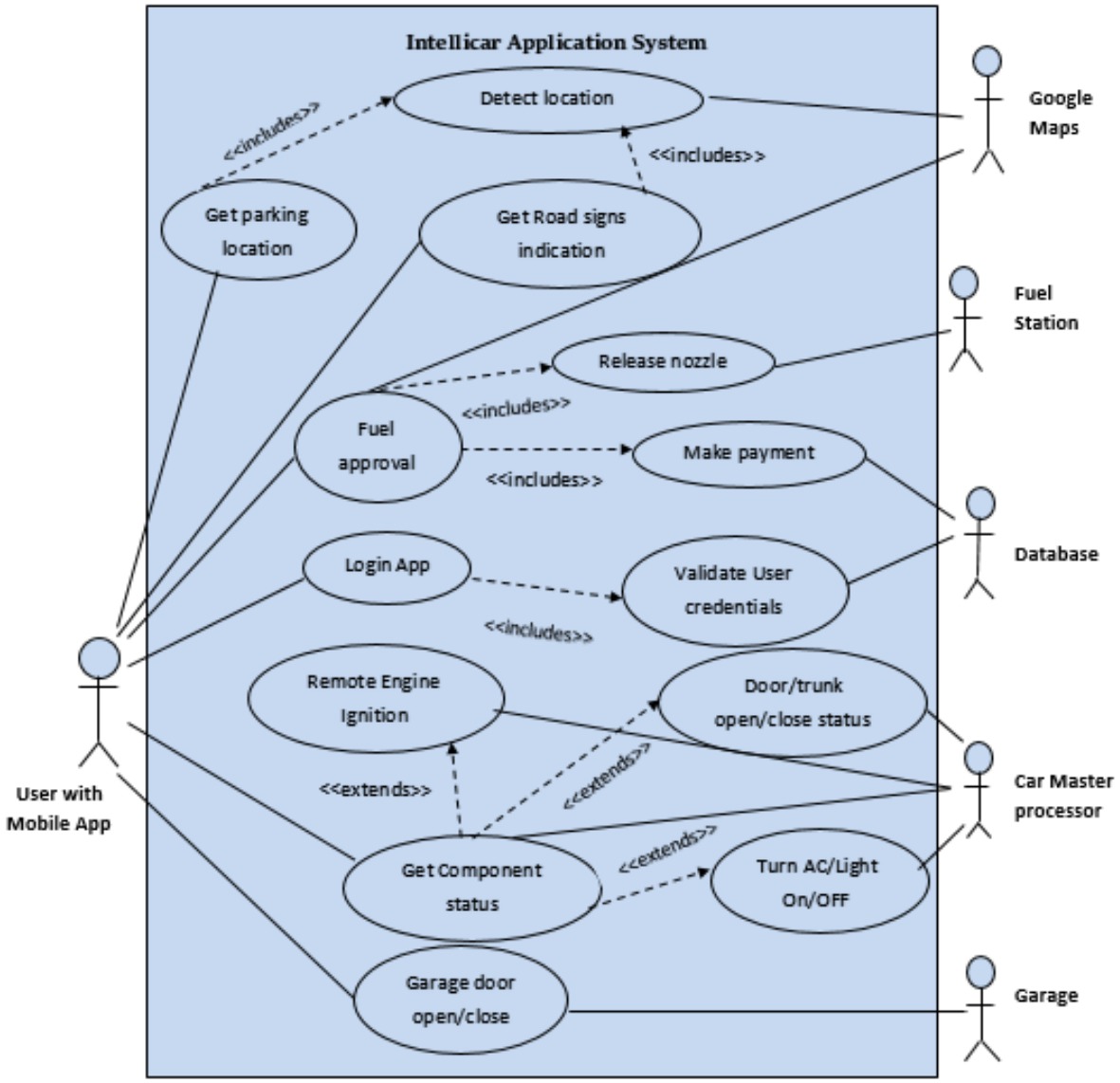

User Mobile Application
Send Engine Ignition Signal
Receive Task completio n Status
Send garage open status
Receive task completio n status
Send door/ trunk completio n status
Light/AC request
Receive completio n
Receive phone beep
User Mobile
Get current Status
IntelliCar
Ignite Engine Status
Connected Car
User Mobile
Get Current Garage status
IntelliCar
Open Garage door
Garage Door
User Mobile
Get current light/AC status
IntelliCar
Turn on/off signal
Connected Car
User Mobile
Get current door/ trunk status
IntelliCar
Open/close door/ trunk signal
Connected Car
Detect p rolonged door open
Connected Car
IntelliCar
Get error status
Ignite engine signal
Get current garage status
Open garage door
I Get current l light status
Turn on/ off signal
Get current door status
Open/ close door signal
Detect door open
Connected Car
Send current engine status
Ignite Status
Send current light/AC status
Turn on/ off
Send current door/ trunk status
Close/ open
Send door/trust status
Database
SaveTask completion
SaveTask completion
SaveTask completion
SaveTask Completion
SaveTask Completion
Function
Send current Garage Door status
Open Door

11. Context Diagram:



12. Process Model

Use-Case Diagram:



Use-Case Descriptions:

Use Case Name: Sign Up/Login IntelliCar Application

ID: 1

Primary Actor: User

Use Case Type: Detail, essential

Brief Description: Loads functionalities of application

Trigger: User logs into the mobile application

Type: External

Relationship:

Association: Mobile application

Includes: Validates user credentials

Normal flow of events:

1. The user launches the app and registers his information.

S-1 subflow is followed for entering user information.

2. The user provides first name and last name along with email ID.
3. The user provides the mobile number and IMEI number.
4. The user is prompted to enter the username and password.
5. The user enters the credentials and is directed to the home page of the app.
6. The app displays the various functionalities it can perform.

Subflow:

S-1. User information

1. User enters his basic information and his credit card details for future reference.

External flow:

5.A. If Login authentication fails, user re-enters his details.

Use Case Name: Remote Engine Ignition

ID: 2

Primary Actor: User

Use Case Type: Detail, essential

Brief Description: Ignites the car engine

Trigger: User wants to start the car engine

Type: External

Relationship:

Association: Mobile application, Intellicar Processor, Car

Normal flow of events:

1. User chooses to click the Ignite Car Engine Ignition button.
2. Intellicar application gets the status of the car engine.
3. Intellicar triggers the engine ignition.
4. The car engine gets ignited.
5. Task completion status notified to user.

External flow:

4.A. If task is not completed, re-initiate the task.

Use Case Name: Parking Location Share

ID: 3

Primary Actor: Car processor/ Google maps

Use Case Type: Detail, essential

Brief Description: User receives parking location of the car

Trigger: Car is parked at a location

Type: External

Relationship:

Association: Mobile, Intellicar Processor, Car

Includes: Google maps detects car location

Normal flow of events:

1. Car Engine stops and doors are locked.
2. Gets the confirmation of locking.
3. The Intellicar processor will interact with Google Map.
4. Intellicar receives location from maps.
5. The parking location is shared as a message to user.

Use Case Name: Road Signs Guide

ID: 4

Primary Actor: Car processor/ Google maps

Use Case Type: Detail, essential

Brief Description: User receives Road Signs information

Trigger: User is driving the car

Type: External

Relationship:

Association: Mobile, Intellicar Processor, Car

Includes: Google maps detects car location

Normal flow of events:

1. User drives the car.
2. The Intellicar processor interacts with Google Maps and deducts location and uses its road sign information.
3. Each upcoming road sign is recognized before 500 feet.
4. User receives the road signs information while driving.

Use Case Name: Fuel Approval and Payment

ID: 5

Primary Actor: User/ Car processor

Use Case Type: Detail, essential

Brief Description: Card less fuel payment by user

Trigger: Car reaches fuel station

Type: External

Relationship:

Association: Mobile, Intellicar Processor, Car, Fuel Station, Bank, Camera

Includes: Fuel station authentication, bank authentication to make payment.

Normal flow of events:

1. User reaches the fuel station.
2. Intellicar deducts current location with google maps.
3. User parks his car before a particular fuel tank.
4. The camera scans the car number and sends to database.
5. Intellicar verifies the car number.
6. The user enters the app and makes payment.

S-1 subflow is performed while making payment.

7. The user pays amount for required fuel.
8. Intellicar sends status and nozzle releases.
9. User fills the fuel.

Subflow:

S-1: User makes payment

1. The user enters the payment function in the Mobile Application.
2. User selects the method of payment and pays for fuel.
3. The nozzle is released at station to fill fuel.

Exceptional Flow:

2.A. If the car number is not present the process will not proceed further.

3.A. If car is accepted the transaction proceeds else the transaction is cancelled.

4.A. User authentication by bank database.

4.B. If bank transaction fails, the entire process is re-initiated.

Use Case Name: Open/Close Garage Door

ID: 6

Primary Actor: User/Car Processor

Use Case Type: Detail, essential

Brief Description: Garage door is open/close

Trigger: User wants to open/close the garage door

Type: External

Relationship:

Association: Intellicar Processor, Garage

Normal flow of events:

1. User clicks the open/close garage door button.
2. Gets current status of garage door.
3. Processor triggers garage door open/close.
4. The garage door opens/closes.
5. Display confirmation user.

Use Case Name: Switch ON/OFF Car lights/AC

ID: 7

Primary Actor: User/ Car Processor

Use Case Type: Detail, essential

Brief Description: The car lights are switched ON/OFF

Trigger: User wants to switch ON/OFF car lights/AC

Type: External

Relationship:

Association: Intellicar Processor, Car Lights, Air Conditioner

Normal flow of events:

1. User clicks the switch ON/OFF lights button or AC button.
2. The Intellicar App interacts with car processor.
3. Intellicar App triggers the processor.
4. The car lights/AC are switched ON/OFF.

Exceptional Flow:

4.A. If the temperature is optimum according to the user then the user doesn't take any action

Use Case Name: Open/Close Car Trunk/Door

ID: 8

Primary Actor: User/ Car Processor

Use Case Type: Detail, essential

Brief Description: The car trunk/door are open/closed

Trigger: User wants to open/close car trunk/door

Type: External

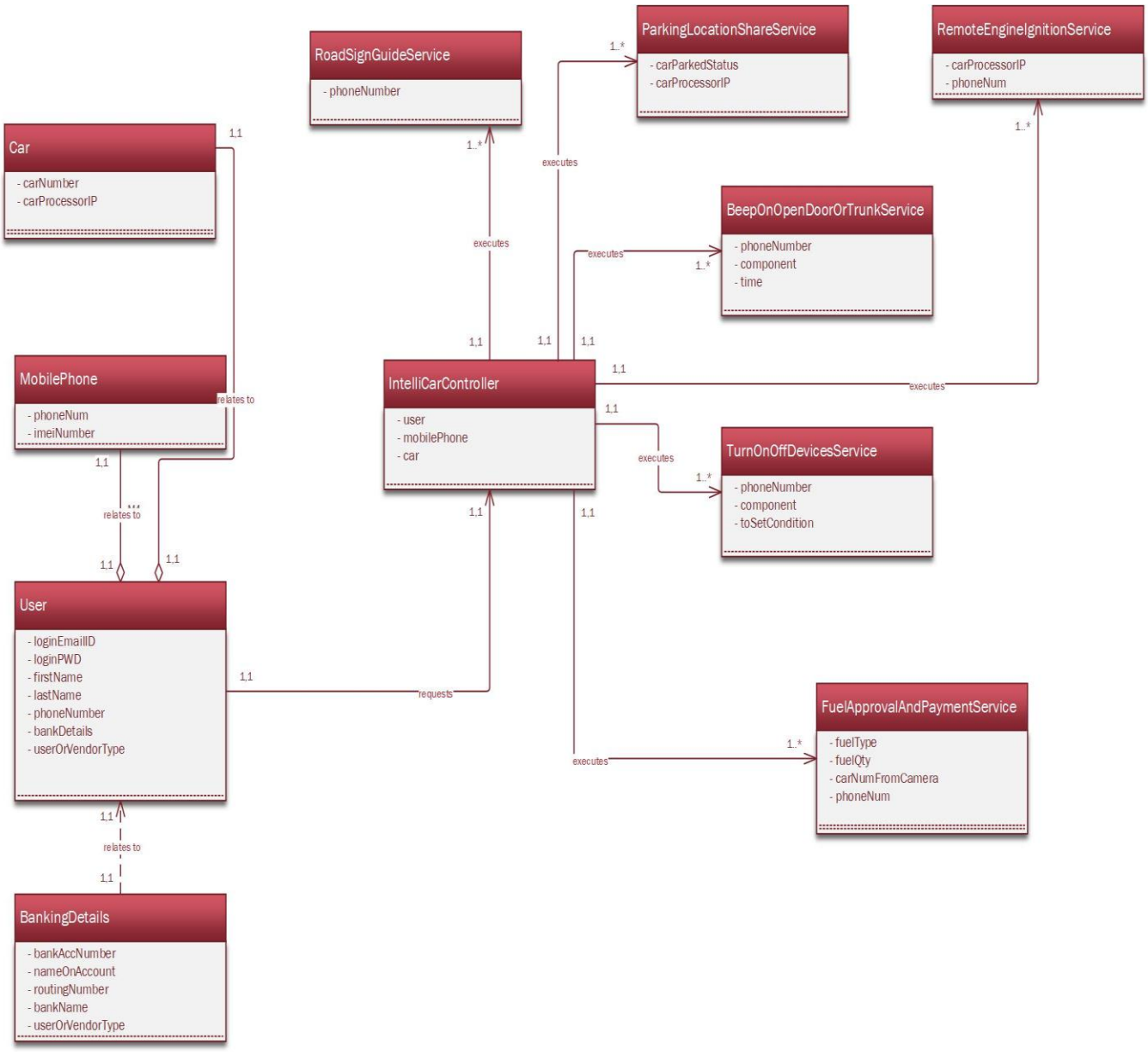
Relationship:

Association: Intellicar Processor, Car Trunk/Door

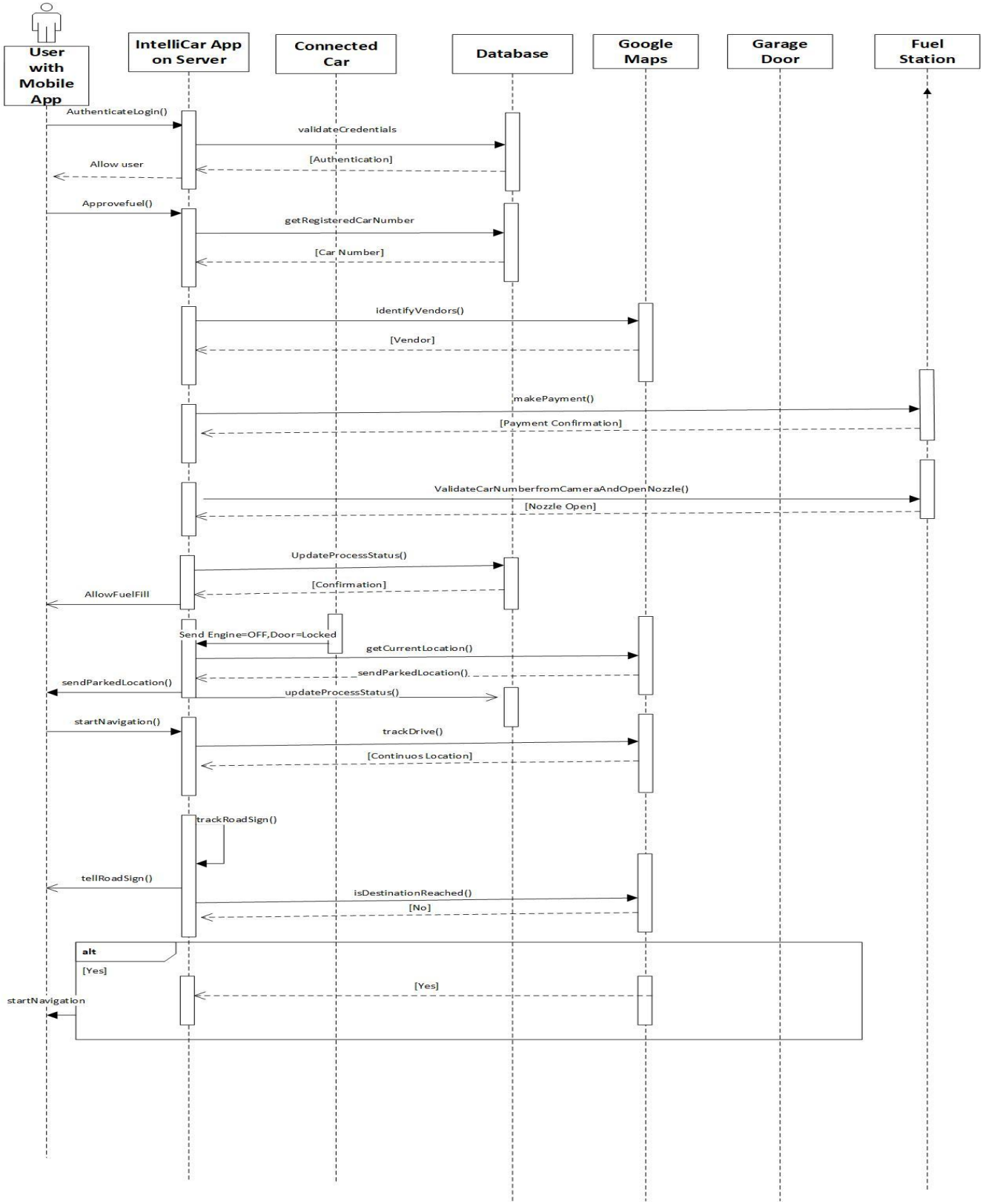
Normal flow of events:

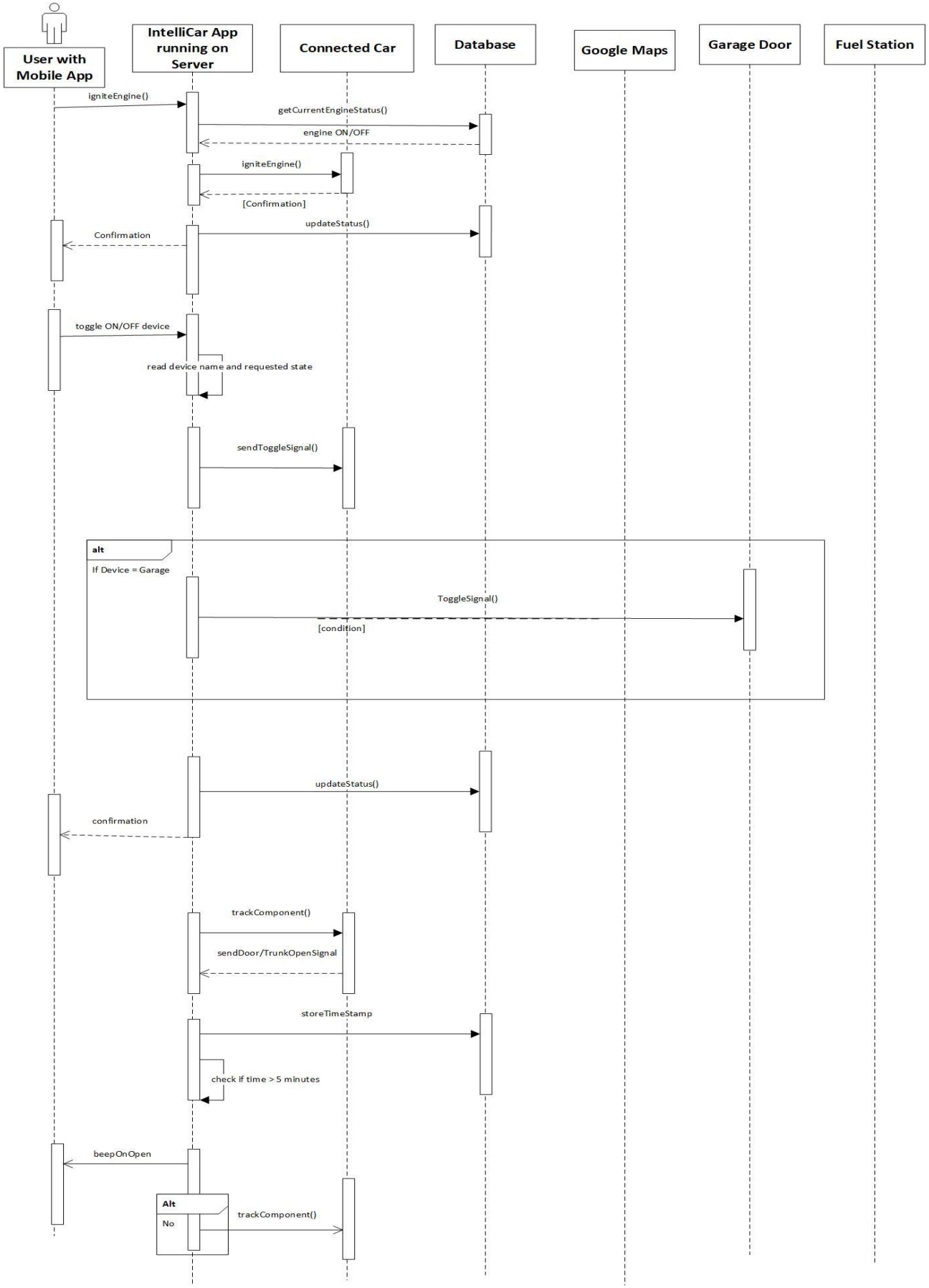
1. User clicks the trunk/door open/close button.
2. The Intellicar App interacts with car processor.
3. Intellicar App triggers the processor.
4. The car trunk/door is open/close.

13. Data Model - Class Diagram:



14. Object Behavior Model - Sequence Diagram:




User with Mobile App
IntelliCar App running on Server
Connected Car
Database
Google Maps
Garage Door
Fuel Station
igniteEngine()
getCurrentEngineStatus()
engine ON/OFF
igniteEngine()
[Confirmation]
updateStatus()
Confirmation
toggle ON/OFF device
read device name and requested state
sendToggleSignal()
alt
If Device = Garage
ToggleSignal()
[condition]
updateStatus()
confirmation
trackComponent()
sendDoor/TrunkOpenSignal
storeTimeStamp
check if time > 5 minutes
beepOnOpen
Alt
No
trackComponent()

15. Documentation of All Data - Data Dictionary

User= UserID + **Name** + Address + Email ID + Password + [Car User/ Fuel Vendor]

Name= First Name + Last Name

Car= CarNumber + Car Master Processor IP + Make

Mobile Phone= MobileNumber + IMEI Number

Bank Details= BankAccNum + Name of the Account + Routing Number + Bank Name

User Registration= RegistrationID + UserID + CarNumber + BankAccNum + MobileNumber

Beep On Open= CarNumber + Is Door open + Is Trunk Open + Time Stamp on Open + Beep Needed

Intellicar Requests= RequestID + Request Type + Registration ID

Parking Location Share= RequestID + Car Parked Status + Time Stamp + Task Status + Parked Location Address

Remote Engine Ignition= RequestID + Current Engine Status + Task Status

Fuel Approval and Payment= RequestID + Fuel Type + Fuel Quantity + Car Number from Camera + Fuel Vendor + Task Status

Road Sign Guide= RequestID + Car Navigation + Task Status

Toggle Device= RequestID + Device Name + Current Device Status + Requested State + Task Status

16. Functional Specifications of the Proposed System

- Parking Location Share feature along with the Google Maps app allows a user to receive the current location of the car once it is parked. This helps the users in locating the car parking spot easily once they are back.

- RoadSignGuideService interacts with Google Maps app to assist a user with a voice indication of the upcoming traffic and road sign boards while driving. This keeps the users alert and makes their driving experience hassle-free.

- The BeepOnOpenDoorOrTrunk feature sends a notification to the user whenever the car door or the trunk is left open for a long time. This feature is useful for users who are absent-minded.

- TurnOnOffDevice service allows a user to control the switching on/off of lights, air conditioner/heater, locking and unlocking of door/windows and also closing/opening of garage door from a distant location.

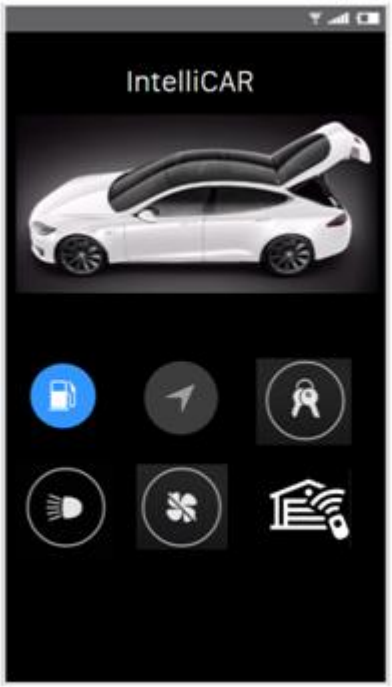
- Cardless fuel payment feature lets a user pay through the IntelliCar app when a user goes to a gas station with IntelliCar facilities.

- The car icon in the IntelliCar app shows the user if the door/trunk is open or if the lights are left on.

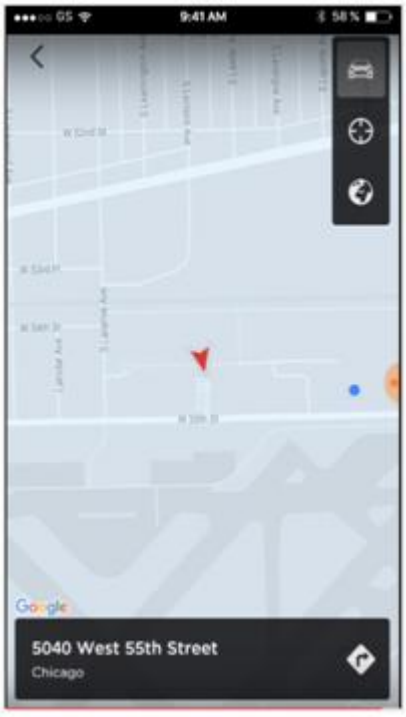
17. Interface Design

Login Screen:

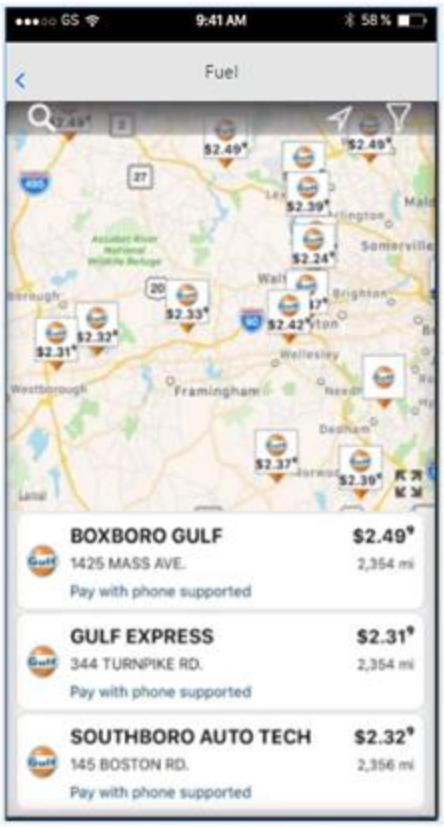
Main Screen:



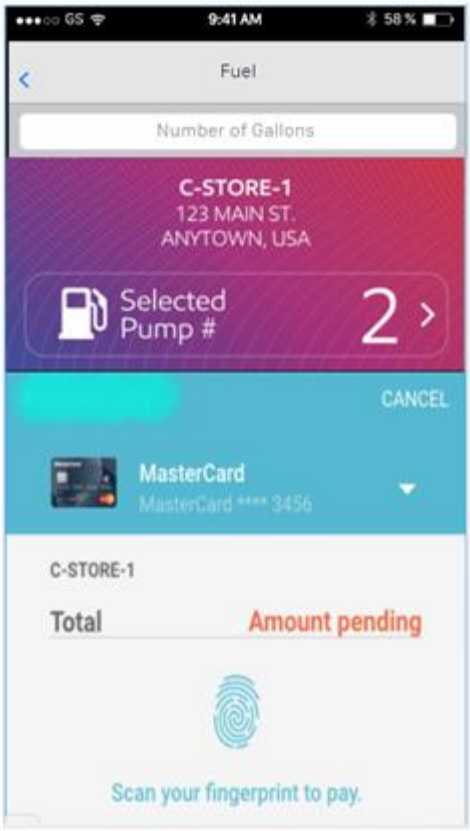
Location of the Vehicle:



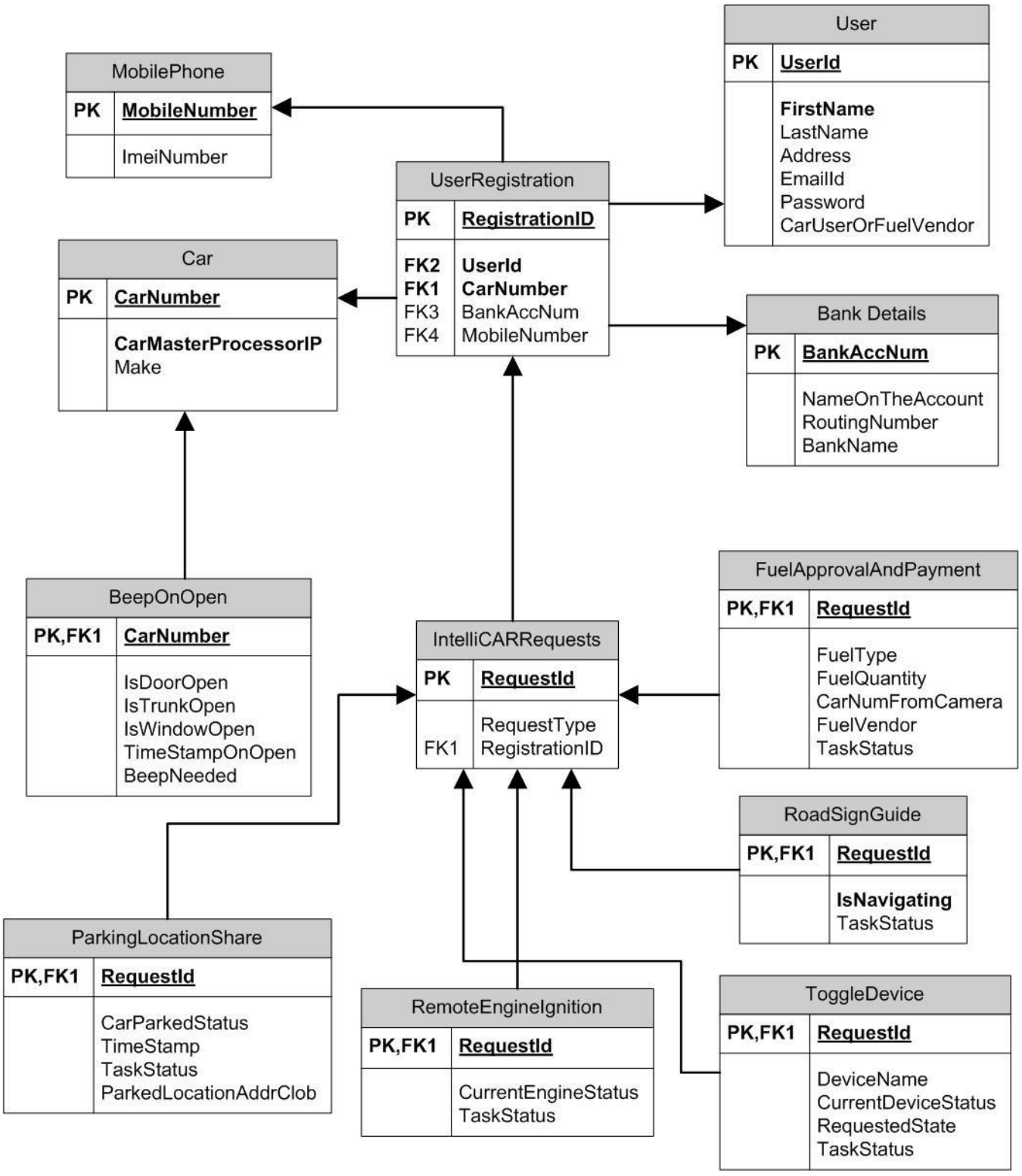
Fuel Station Selection:



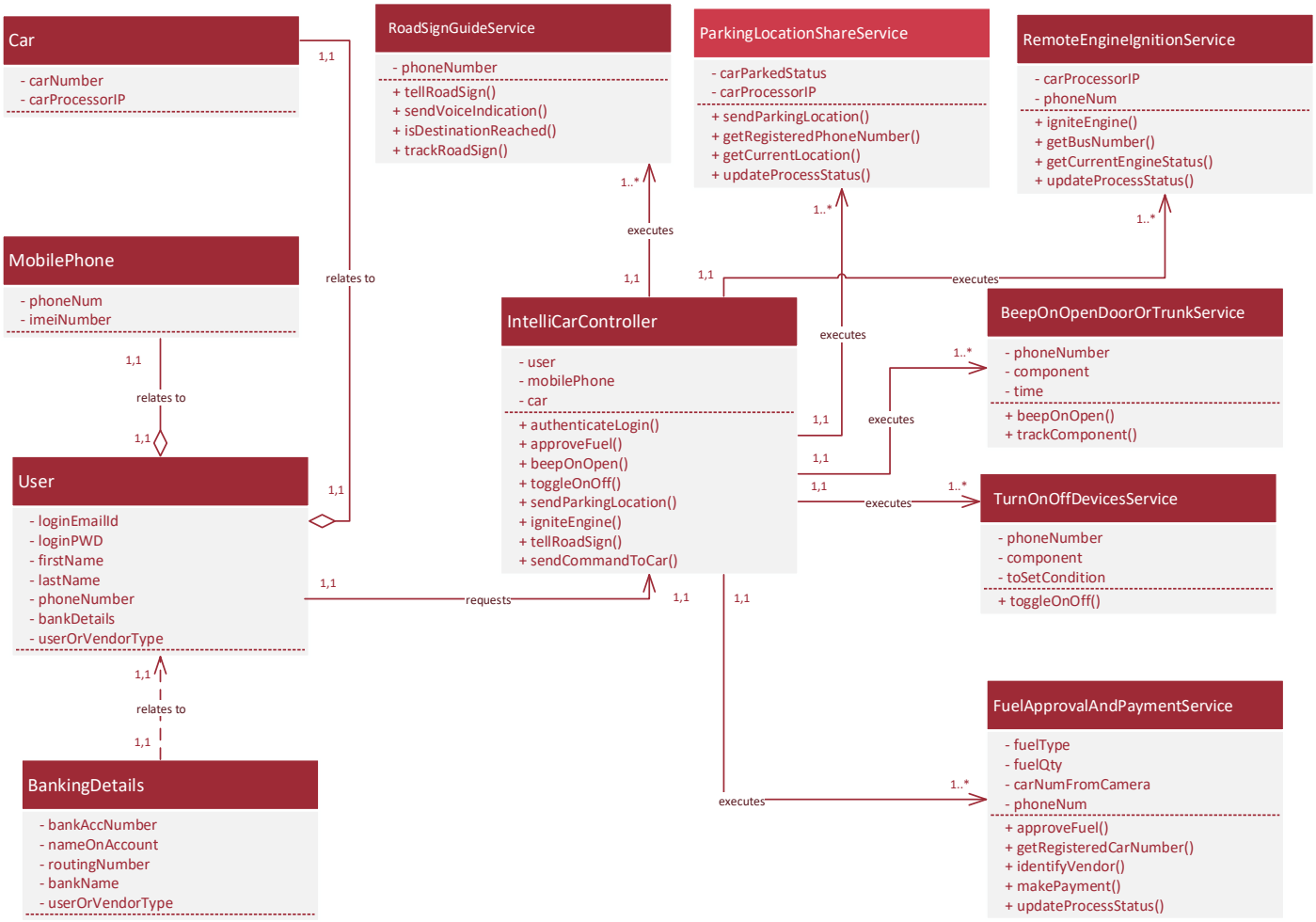
Fuel Payment:



18. Database Design



19. Complete Class Diagram



20. Software Design

Method Name: igniteEngine()

Class Name: RemoteEngineIgnitionService

ID: EngineFeature

Clients (Consumers): Car Driver

Associated Use Cases: User Authentication, Send Command To Car

Description of Responsibilities: If engine is idling, send ignition signal.

Update the user about the operation performed.

Arguments Received: String carProcessorIP, long phoneNumber

Type of Value Returned: String “Complete” to acknowledge that the location is shared; otherwise “Incomplete”

Pre-Conditions: User sign in is required

Post-Conditions: Engine ignition triggered

Pseudocode: (Structured English)

Get the carProcessorIP from DB using phone number

IF carProcessorIP received from the mobile = carProcessorIP registered with the mobile

THEN

Get the processor to engine bus’s number from DB

WHILE engine status = IDLE & process = INCOMPLETE

DO Send ignition signal

IF process = COMPLETE

THEN Update the process status = COMPLETE in DB

END IF

END IF

Method Name: sendParkingLocation()

Class Name: ParkingLocationShareService

ID: ParkingLocationFeature

Clients (Consumers): Car Driver

Associated Use Cases: User Authentication, Send Notification To Mobile Application

Description of Responsibilities: If the car is parked, send current location google map notification to the mobile application

Arguments Received: Boolean carParkedStatus, carProcessorIP

Type of Value Returned: String “Complete” to acknowledge that the location is shared; otherwise “Incomplete”

Pre-Conditions: Car is parked

Post-Conditions: Google current location shared

Pseudocode: (Structured English)

IF carParkedStatus = true

THEN

Get the registered phone number from DB using carProcessorIP

Send google current location notification

Update the process status = COMPLETE in DB

END IF

Method Name: approveFuel()

Class Name: FuelApprovalAndPaymentService

ID: FuelFeature

Clients (Consumers): Car Driver and Gas Vendor

Associated Use Cases: User Authentication, Approve Fuel Litres for the Car, Make Payment

Description of Responsibilities:

1. Automatically, acquire the bank details of the user and the fuel vendor (using google location).
2. Acquire fuel needs, make payment and approve fuel nozzle opening

Arguments Received: Long phoneNumber, String carNumFromCamera, String fuelType, int fuelqty

Type of Value Returned: Boolean approval true/false

Pre-Conditions: User payment details validated, Car is at the fuel station, fuel requirements submitted

Post-Conditions: Google current location shared

Pseudocode: (Structured English)

Get the car number for the phone number from DB

IF car number from camera = car num from DB

THEN

Get the user fuel needs

Identify the fuel vendor based on location

Transact amount from user account to vendor

IF payment = success

THEN Send Fuel Nozzle open signal along with fuel quantity

END IF

END IF

Method Name: tellRoadSign()

Class Name: RoadSignGuideService

ID: RoadSignVoiceFeature

Clients (Consumers): Car Driver

Associated Use Cases: User Authentication, Continuous Road Sign Tracking, Voice Indication

Description of Responsibilities:

1. Track user drive location using google maps
2. Track road signs saved
3. Send voice indication

Arguments Received: Long phoneNumber

Type of Value Returned: void

Pre-Conditions: Road sign information embedded into all google locations

Post-Conditions: Voice indication of road signs

Pseudocode: (Structured English)

IF navigate = true

THEN

WHILE destination = NOTREACHED

Do

WHILE hasNextRoadSign = true

Do Send voice indication

END IF

Method Name: toggleOnOff()

Class Name: TurnOnOffDevicesService

ID: DeviceControlFeature

Clients (Consumers): Car Driver

Associated Use Cases: User Authentication, component Tracking

Description of Responsibilities:

1. Get the on/off | open/close status of the components like car doors, garage door, car lights, AC
2. Get the user input (on/off)
3. Send toggle signal

Arguments Received: Long phoneNumber, String component, Boolean condition

Type of Value Returned: String “Complete” to acknowledge that component toggle is done

Pre-Conditions: Continuous Component Tracking

Post-Conditions: ON/OFF | OPEN/CLOSE done

Pseudocode: (Structured English)

Get the name of the component to be operated from user

Get the condition to be set (on/off | open/close) from user

IF currentCondition NotEqualTo requiredCondition

THEN Send toggle signal to the component

END IF

Method Name: beepOnOpen()

Class Name: BeepOnOpenDoorOrTrunkService

ID: BeepFeature

Clients (Consumers): Car Driver

Associated Use Cases: User Authentication, Continuous Door and Trunk Tracking, Track Time

Description of Responsibilities:

1. Get the new door/trunk open status
2. Calculate the time it is being kept open
3. Send beep signal to phone

Arguments Received: Long phoneNumber, String component, DateTime time

Type of Value Returned: Boolean beep = true

Pre-Conditions: Continuous door/trunk Tracking

Post-Conditions: Beep on 5 minutes of open state

Pseudocode: (Structured English)

Get component open signal

Set status = open and time = system time

CASE

IF front right door: component = front right door

IF front left door: component = front left door

IF back right door: component = back right door

IF back left door: component = back left door

IF trunk door: component = trunk door

END CASE

IF systemTime = time + 5 minutes

THEN send beep signal and component

Project Management Deliverables:

21) Project Activities and Schedule:

Weekly Schedule	Tasks
May 30, 2017 to June 5, 2017	Group Formation and Brainstorming Ideas
June 6, 2017 to June 12, 2017	Finalized on “IntelliCar” and functionalities of project #Milestone 1
June 13, 2017 to June 19, 2017	Documentation of Problem Statement and Objectives
June 20, 2017 to June 26, 2017	Designed BPMN model and Context Diagram
June 27, 2017 to July 2, 2017	Identified process and designed use case diagram and use case description #Milestone 2
July 3, 2017 to July 9, 2017	Developed class diagram and Sequence diagram
July 10, 2017 to July 16, 2017	Designed Database and Interface #Milestone 3
July 17, 2017 to July 23, 2017	Software Design and Algorithm
July 24, 2017 to July 30, 2017	Implementation of prototype of the proposed system #Milestone 4
July 31, 2017 to August 6, 2017	Report documentation, proof-read, and submission #Milestone 5

22) Planned vs Execution Timeline

Planned Project Days Allocation	Actual Project Days Allocated
May 30 – August 6, 2017	**May 30 – August 6, 2017**

23) Allocation of Activities

Ashwin	1. Executive Summary 2. Problem Statement 3. Business process model using BPMN for the key business processes
Preeti	1. Context Diagram for the proposed system 2. Process Model 3. Data Model
Gautami	1. Object Behavior Model 2. Documentation of all data 3. Functional Specification Document for the proposed system
Divya	1. Complete Class Diagram 2. Software Design 3. Implementation
Ravi	1. Database Design 2. Interface Design

24) Minutes of Meeting

Meeting 1:

Meeting for	Group 3 - May 30, 2017 4:00 – 5:00 PM
Meeting Agenda	Brainstorming Project Ideas
Attendees	All Members
Project Idea Research	All Members
Deadline	June 5, 2017

Meeting 2:

Meeting for	Group 3 -June 6, 2017 4:00 – 5:00 PM
Meeting Agenda	Finalizing Project Topic – Discussing feasibility
Attendees	All Members
Project Idea Research	All Members
Deadline	June 13, 2017

Meeting 3:

Meeting for	Group 3 -June 14, 2017 4:00 – 5:00 PM
Meeting Agenda	Defining Problem Statement and Objective
Attendees	All Members
Project Idea Research	All Members
Deadline	June 19, 2017

Meeting 4:

Meeting for	Group 3 -June 20, 2017 4:00 – 5:00 PM
Meeting Agenda	Designed BPMN model and Context Diagram for the proposed system
Attendees	All Members
BPMN model and Context Diagram	Ashwin, Gautami, Divya
Deadline	June 26, 2017

Meeting 5:

Meeting for	Group 3 -June 27, 2017 4:00 – 5:00 PM
Meeting Agenda	Identified process and designed use case diagram and use case description
Attendees	All Members
Use Case diagram	Gautami, Divya, Ravi, Ashwin
Deadline	July 3, 2017

Meeting 6:

Meeting for	Group 3 -July 3, 2017 4:00 – 5:00 PM
Meeting Agenda	Developed class diagram and Sequence diagram for the proposed system
Attendees	All Members
Class diagram and Sequence diagram	Preeti, Gautami
Deadline	July 10, 2017

Meeting 7:

Meeting for	Group 3 -July 11, 2017 4:00 – 5:00 PM
Meeting Agenda	Designed Database and Interface
Attendees	All Members
Database and Interface	Ravi, Preeti
Deadline	July 17, 2017

Meeting 8:

Meeting for	Group 3 -July 18, 2017 4:00 – 5:00 PM
Meeting Agenda	Software Design and Algorithm
Attendees	All Members
Software Design and Algorithm	Divya
Deadline	July 24, 2017

Meeting 9:

Meeting for	Group 3 -July 25, 2017 4:00 – 5:00 PM
Meeting Agenda	Implementation of prototype of the proposed system
Attendees	All Members
Implementation	All Members
Deadline	July 31, 2017

Meeting 10:

Meeting for	Group 3 -August 1, 2017 4:00 – 5:00 PM
Meeting Agenda	Report documentation, proof-read
Attendees	All Members
Documenting Deliverables	All Members
Deadline	August 6, 2017