**Statement of Requirements (SOR)**

***DiGiSENSE RESTFul API***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Version No** | **Approved Date** | **Approver Name** | **E-Mail Id** | **Remarks** |
| 0.1 |  | Anoop RAJAGOPAL | [RAJAGOPAL.ANOOP@mahindra.com](mailto:RAJAGOPAL.ANOOP@mahindra.com) |  |

This document is to be considered as a whole, the parts of which shall not be separated.

© Mahindra & Mahindra Ltd.  
No duplication permitted without the consent of the issuing department.  
No circulation permitted without the consent of Mahindra & Mahindra.

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver. No** | **Date** | **Author** | **Description** | **Remarks** |
| 0.1 | 13-10-2016 | Gireesan Namboothiri P. | Initial Document |  |
| 0.4 | 02-02-2017 | Gireesan Namboothiri P. | Added Trip status, driver data edit, trip expenses. Prioritized API for first delivery. |  |
| 0.5 | 13-02-2017 | Gireesan Namboothiri P. | Cleaned up documentation. Added links based on publically hosted API. Re-prioritized based on March 10 App Delivery. Added time format |  |
| 0.6 | 15-02-2017 | Gireesan Namboothiri P. | Added expense types, expanded API operations. Changed priority of 1.18 to match dashboard report. Added YAML file in word |  |
| 0.7 | 28-02-2017 | Gireesan Namboothiri P. | Modified requirements for Driver ID based on Base64 data. Driver Unmap if the value is not available.  Incidental expenses, response string updated. |  |
| 0.8 | 15-03-2017 | Gireesan Namboothiri P. | Added priorities for next cycle. Trip Management, Reports, GeoFence, Push Notifications. Added additional inputs from Farm (based on Sidhant’s inputs) |  |
| 0.9 | 22-03-2017 | Gireesan Namboothiri P | Corrected Reports MTBD summary and added link to the swagger. Swagger file version 2.0.0. Alerts updated with default 48 hours. |  |
| 0.10 | 12-04-2017 | Gireesan Namboothiri P | Added priority to Farm Specific APIs |  |
| 0.11 | 05-02-2017 | Gireesan Namboothiri P. | Added Requirement roadmap for all the platforms. Removed Anoop Rajagopal’s details. |  |

**In case of questions about this document and its contents, do not hesitate to contact us**

|  |  |  |
| --- | --- | --- |
| **Name** | **Contact Number** | **E-mail Id** |
| Gireesan Namboothiri P. | 044-22167347  9845487983 | namboothiri.gireesan@mahindra.com |
|  |  |  |

**Requirement Color Definitions**

|  |  |
| --- | --- |
| **Color** | **Description** |
|  |  |
|  |  |
|  |  |

**Table of Contents**

1. Introduction 6

1. Purpose of the document 6

2. Requirements Terminology 6

3. Confidentiality 6

4. Coverage of Platform/Vehicles 7

5. Glossary 7

6. Terms Definition 7

2. Documents 7

2.1. Reference Documents 7

2.2. Applicable Documents 7

3. High Level Requirements 7

4. High Level Diagram 8

4.1. General System Architecture 8

5. System Requirements 9

1. Functional Requirements 9

3. Non-Functional Requirements 21

Annexure: I 23

Annexure: II 23

# Introduction

## Purpose of the document

This document captures the requirements for DiGiSENSE RESTFul API. All the platform functionalities to be extended as RESTFul for flexibility and adaptability.

## Requirements Terminology

The following terminology is used to define the applicability of each requirement in this document.

- The word **Must** in the text means legislative or regulatory requirements (e.g. Health and Safety).

- The word **Shall** in the text means a mandatory requirement.

- The word **Should** in the text means a recommendation or advice on implementing a requirement. Such recommendations or advice are expected to be followed, unless justified reasons are stated for not doing so.

- The word **Will** in the text means an intention or a provision, in connection with a requirement.

**Requirements are noted as follows:**

**Req\_Id Req\_Text**

**Req\_Id** : is the unique requirement identifier of Mid A-IVI EV.

**Req\_Text** : is a description of the requirement.

**Remark:**

- In case of text which doesn’t express a requirement, the word **May** shall be used to denote a permissible practice or action

## Confidentiality

This document is M & M property and contains proprietary and confidential information. This copy is provided on the express condition that its content shall not be reproduced, disclosed to third parties or used, in whole or in part, without the prior written permission of M & M.

## Coverage of Platform/Vehicles

## Glossary

| ***Abbreviation*** | ***Description*** |
| --- | --- |
| *GPS* | *Global Positioning System* |

## Terms Definition

# Documents

## Reference Documents

| **Ref.** | **Reference** | **Version** | **Title / Content** |
| --- | --- | --- | --- |
|  | Mobile Application Manual\_v1 4.pdf | 1.4 | Standardization of API, interfaces, styling and coding standards |

## Applicable Documents

| **Ref.** | **Reference** | **Version** | **Title / Content** |
| --- | --- | --- | --- |
|  |  |  |  |

# High Level Requirements

In the highest level, the system shall implement RESTFul API with Richardson Maturity model level 2. Level 3 is desirable. All the API requirements are captured with Swagger Tools. [Swagger Editor](http://editor.swagger.io/) to be used to visualize and edit the requirements. Current version of the YAML file is kept at [firebase hosting](https://swaggerapis.firebaseapp.com/)

1. Login shall enable Oauth 2.0 with social login support
2. On a successful login, the response will have a [Json Web Token (JWT)](https://jwt.io/).
3. JWT will be used for subsequent requests which provides **authorization** and **authentication**
4. On a failed login, differentiate between login failed (invalid credentials) and user no longer available (user deleted)
5. The API does not differentiate between a specific sector user (eg: FARM/Commercial/MTBD). Instead it should recognize all vehicles mapped under the user name and give responses accordingly.
6. All the requirements and response strings are explicitly captured using Swagger.
7. All the API implementation to be hosted in UAT for testing and validated, security audited APIs to be hosted to the production server
8. Every feature customization for a platform need to be captured using API.
9. On every feature change from the platform, API change has to be initiated, validated with MRV, implemented and tested as per the steps above.
10. Every API to be tested with automation using tools. Eg: Jmeter, POSTMAN
11. Push notifications to be used for **alerts and live route updates**
12. All the date-time format to follow ***2017-01-13 08:18:51*** format

# High Level Diagram

## General System Architecture

Everything in Green comes under the scope of the API development. The priority of implementation will be specified to make it aligned to the business requirements

Oauth 2.0

Social Networks

Oauth 2.0

Mahindra

Auth

Token

Authorization

Role based access: JWT

Token

Mobile

Application

Service API (REST)

Token

Token

Data (JSON): HTTPS

# System Requirements



Legend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Available | To be developed | Not Needed | Platform/Dependency | in progress |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **Subfeatures** | **MTBD** | **FARM** | **CV** | **Personal** |
| Login | Login |  |  |  |  |
|  | Forgot Password |  |  |  |  |
|  | Reset Password |  |  |  |  |
|  | Change Password |  |  |  |  |
| Profile Management | View Profile |  |  |  |  |
|  | Update Profile |  |  |  |  |
| Vehicles | List Vehicles |  |  |  |  |
|  | List Vehicles under Category |  |  |  |  |
|  | VehicleSummary |  |  |  |  |
|  | Live Parameters |  |  |  |  |
| Alerts | For all vehicles, Last 48 Hours |  |  |  |  |
|  | For given vehicle, date range |  |  |  |  |
|  | Service alert: Cost Estimation based on WYH |  |  |  |  |
| LocationHistory | Get Location history for given date range,given vehicle |  |  |  |  |
| Geofence | Create Geofence |  |  |  |  |
|  | Read Geofence details |  |  |  |  |
|  | Update Geofence (add/modify/delete points) |  |  |  |  |
|  | Assign geofence to one or more vehicles |  |  |  |  |
|  | Delete named geofence |  |  |  |  |
|  | Get Geofences for a given vehicle. Lists currently active geofences (based on the date) |  |  |  |  |
|  | Get vehicles mapped to a given geofence with dates |  |  |  |  |
| Route Management | Create Route |  |  |  |  |
|  | Read Route details |  |  |  |  |
|  | Update Route (add/modify/delete points) |  |  |  |  |
|  | Assign Route to one or more vehicles |  |  |  |  |
|  | Delete named Route |  |  |  |  |
|  | Get Routes for a given vehicle. Lists currently active Routes (based on the date) |  |  |  |  |
|  | Get vehicles mapped to a given Route with dates |  |  |  |  |
| Trip Management | Get active trip for a vehicle. Returns the status, current waypoints for the route, pickup/delivery details |  |  |  |  |
|  | Status update: start trip. Update the status of trip as started |  |  |  |  |
|  | Status update: complete the trip. Update the status of trip as þ |  |  |  |  |
|  | Status update: þ a delivery. For a given trip for the vehicle, update a delivery/pick-up as done. |  |  |  |  |
|  | Live update on the notification in case of trip update from the server. |  |  |  |  |
| Driver Management | Create Driver |  |  |  |  |
|  | View Driver |  |  |  |  |
|  | Update Driver |  |  |  |  |
|  | Delete Driver |  |  |  |  |
|  | Vehicle to Driver Mapping |  |  |  |  |
|  | Vehicle to Driver Unmapping |  |  |  |  |
| Breakdown | Invoe breakdown call from mobile |  |  |  |  |
| Dealer | Locate Dealers |  |  |  |  |
|  | Service Booking |  |  |  |  |
|  | Spares Listing- Availability at given dealer |  |  |  |  |
|  | Spares Ordering to given dealer |  |  |  |  |
| Trip Expense Management | Create Expense for a given vehicle |  |  |  |  |
|  | View Expenses for a given vehicle for given date range |  |  |  |  |
| Weather and Tips Feed | View Weather information for given vehicle location |  |  |  |  |
|  | View Farming tips for given vehicle location |  |  |  |  |
| Discussion Forum | Create Discussion Topic |  |  |  |  |
|  | Reply to a topic |  |  |  |  |
| Usage Statistics | Feature usage from app |  |  |  |  |
| Reports | Report Summary. Provides landing information on available reports for a given platform |  |  |  |  |
|  | MTBD:VehicleHealth |  |  |  |  |
|  | MTBD:VehicleStatus |  |  |  |  |
|  | MTBD:Speed Report |  |  |  |  |
|  | MTBD:Usage Time |  |  |  |  |
|  | MTBD:Engine RPM |  |  |  |  |
|  | MTBD:Vehicle Movement Summary |  |  |  |  |
|  | MTBD:Vehicle Movement Report |  |  |  |  |
|  | MTBD:Vehicle Alerts |  |  |  |  |
|  | MTBD:Vehicle Alert-Violations |  |  |  |  |
|  | MTBD:Alert Summary |  |  |  |  |
|  | MTBD:Delivery: Vehicle Usage |  |  |  |  |
|  | MTBD:Delivery: Vehicle Usage Summary |  |  |  |  |
|  | MTBD:Vehicle Health: Vehicle Health Summary |  |  |  |  |
|  | Farm:Vehicle Usage |  |  |  |  |
|  | Farm: Vehicle Details |  |  |  |  |
|  | Farm: Usage hours report |  |  |  |  |
|  | Farm: Vehicle Utilization Report |  |  |  |  |
|  | Farm: Engine Hours Report |  |  |  |  |
| Vehicle Finder/3rd Party Location Sharing | allow other persons, nominated by registered owner/App user, to also see location of vehicle on request |  |  |  |  |
| Activate Vehicle Functions | Remote Door Lock |  |  |  |  |
|  | Remote Door Unloack |  |  |  |  |
|  | Door Lock Status |  |  |  |  |
| General | Push Notifications |  |  |  |  |

## Detailed Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **[Req\_ID]** | **Feature** | **[Requirement\_Text]** |  |
| [1.01](http://editor.swagger.io/#/?tags=req101) | [Login](https://swaggerapis.firebaseapp.com/#!/Login/post_login) | Login using Mahindra DiGiSENSE Credentials. Shall return an authentication and authorization token on success | 1.01 |
| 1.02 | Login | Login Using Social Login. Shall use Standard Oauth 2.0 handshake. On success, return an authentication and authorization token |  |
| 1.03 | Login | Login using Mahindra Oauth Services for DiGiSENSE. . Shall use Standard Oauth 2.0 handshake. On success, return an authentication and authorization token |  |
| 1.04 | Login | Social Login should support Google and Facebook integration |  |
| 1.05 | Login | Linking of DiGiSense id with Social login credentials to be supported |  |
| 1.06 | [Login](https://swaggerapis.firebaseapp.com/#!/Login/post_forgotPassword) | ForgotPassword: Should use the standard forgot password for DiGiSENSE | 1.02 |
| 1.07 | [Login](https://swaggerapis.firebaseapp.com/#!/Login/post_resetPassword) | resetPassword: Should use the standard reset password for DiGiSENSE | 1.03 |
| 1.08 | [Login](https://swaggerapis.firebaseapp.com/#!/Login/post_changePassword) | ChangePassword: Should use the standard change password for DiGiSENSE | 1.04 |
| 1.07 | Login | ForgotPassword: Should use the standard forgot password for respective login types (DiGiSENSE/Facebook/Google/Mahindra Oauth 2.0) |  |
| 2.01 | ServiceAvailable | List all the services available to the current logged in user. Eg: Listing of vehicles, listing of alerts, listing of reports, viewing reports, and track vehicle |  |
| 2.02 | ServiceAvailable | Available service for the user is directly linked to the Role Based Access of the DiGiSENSE platform. Any change in platform through API/WEB interface should reflect the same in service listing |  |
| 2.03 | RoleListing | user profileble roles for the current user (differentiate driver and owner) |  |
| 2.04 | RoleCreation | Create a new role (if allowed). Creates the role with a unique name. |  |
| 2.05 | RolePermissions | Update permissions for a given role name. This updates the service available |  |
| 2.06 | RoleAssign | Assign roles to a group of users. This include available users (permitted to be listed under current credentials) to be listed and modify (add/delete) role to the user |  |
| 3.01 | [listVehicles](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_listVehicles) | Get the list of vehicles. All the vehicles under the login with vehiclePlatform, vehicleModel, variant, registration number, current status, last updated timestamp, last known location, priority alert status, current address of the location to be sent.  **If the Platform is MTBD**, additional information to be sent are  Speed  runningHours  driverName | 1.05 |
| 3.02 | [listVehicles/mtbd](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_listVehicles_mtbd) | **If the Platform is MTBD**, additional information to be sent are  Speed  runningHours  driverName  This API to return only MTBD vehicles | 1.06 |
| 3.03 | listVehicles/farm | **If the Platform is FARM**, additional information to be sent are  runningHours  ptohours  fuellevel  This API to return only FARM vehicles | 3.01 |
| 4.01 | [vehicleSummary](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_vehiclesSummary_vehicleRegNohttps://swaggerapis.firebaseapp.com/) | Summary information for the vehicle.  registrationNumber:  vehicleLastUsedOn: | 1.07 |
| 4.02 | [vehicleSummary](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_vehiclesSummary_vehicleRegNo) | **If the Platform is MTBD** information to be sent are  totalUsage:  vehicleRunningHours  vehicleIdleHours | 1.08 |
| 4.03 | [vehicleSummary](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_vehiclesSummary_vehicleRegNo) | **If the Platform is FARM** information to be sent are  vehicleRunningHours  vehiclePTOHours | 3.02 |
| 5.01 | [liveVehicleInformation](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_liveVehicleInformation_vehicleRegNo) | Get the live vehicle parameters. A client which refresh this API at 1 minute to get latest information. Following information to be returned  registrationNumber  status  lastupdated | 1.09 |
| 5.02 | [liveVehicleInformation](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_liveVehicleInformation_vehicleRegNo) | **If the Platform is MTBD** information to be sent are  highEngineTemperature  engineRPM  fuelLevel  vehicleHealth  fuelEfficiencyA  fuelEfficiencyB  averageVehicleSpeed  engineOilPressure  driverName | 1.10 |
| 5.03 | [liveVehicleInformation](https://swaggerapis.firebaseapp.com/#!/Vehicle/get_liveVehicleInformation_vehicleRegNo) | **If the Platform is FARM** information to be sent are  highEngineTemperature  highengineRPM  lowfuel  vehicleHealth  airfilterclog  clutchoverride  keyremoved  engineOilPressure  batterynotcharging  geofence  curfew | 3.03 |
| 6.01 | [Locatedealers](https://swaggerapis.firebaseapp.com/#!/Dealers/get_vehicles_vehicleRegNo_dealers) | Get list of dealers for the current (based on platform and model). The list to be sorted in the distance from vehicle’s current location. Nearest dealer to be the first one in the response. Response should have dealer name, address, phone number, location(latitude, longitude) |  |
| 6.02 | [Locate dealers and RSA for MTBD](https://swaggerapis.firebaseapp.com/#!/Dealers/get_vehicles_vehicleRegNo_dealers) | These points along with reverse geo coded lat,lons to be inserted in to the Database tables for MTBD.  All of these are not dealers. This include Road Side Assistance and authorized service centers. | 1.11 |
| 7.01 | Spares | Get a list of spares for current vehicle (based on platform and model).  Response should include  Sparename  Spareid  Details |  |
| 7.02 | Spares | Request for a spare. This should link to the dealer management system to order the spare along with current location of the vehicle.  This should return a request id with nearest dealer’s information where the request has been registered |  |
| 7.03 | [VehicleAlerts](https://swaggerapis.firebaseapp.com/#!/VehicleAlerts/get_vehicleAlerts) | List of alerts for all vehicles under the current user,  Name, type, location, priority, timestamp. All the alerts specified for the given platform should be enumerated in the above mentioned format  IF the parameters are blank, return default 48 hours alert | 1.12 |
| 7.04 | [VehicleAlerts](https://swaggerapis.firebaseapp.com/#!/VehicleAlerts/get_vehicleAlerts_vehicleRegNo) | List of alerts for a given vehicle with  Name, type, location, priority, timestamp. All the alerts specified for the given platform should be enumerated in the above mentioned format  IF the parameters are blank, return default 48 hours alert | 1.13 |
| 8.01 | reportSummary | Get Summary of reports for given platform |  |
| 8.02 | [ReportSummary/**MTBD**](https://swaggerapis.firebaseapp.com/#!/Report/get_reportsummary_platform_mtbd) | Summary with relative urls for all reports to be sent as per the format in Swagger | 2.08 |
| 8.03 | Report | Every report information to be enumerated with  Report type (bar,pie, line, table, Label) Name, value series/values. The report should take default week (past 7 days)   * + - 1. [Speed Report](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclemovement_speeddata_vehicleRegNo)       2. [Usage Time](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclemovement_usagetime_vehicleRegNo)       3. [Engine RPM](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclemovement_enginerpm_vehicleRegNo)       4. [Vehicle Movement Summary](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclemovement_vehiclemovementsummary_vehicleRegNo)       5. [Vehicle Movement Report](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclemovement_vehiclemovementreport_vehicleRegNo)       6. [Vehicle Alerts](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_alerts_vehiclealerts_vehicleRegNo)       7. [Vehicle Alert-Violations](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_alerts_violation_vehicleRegNo)       8. [Alert Summary](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_alerts_alertssummary_vehicleRegNo)       9. [Delivery: Vehicle Usage](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_delivery_vehicleusage_vehicleRegNo)       10. [Delivery: Vehicle Usage Summary](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_delivery_vehicleusagesummary_vehicleRegNo)       11. [Vehicle Health: Vehicle Health Summary](https://swaggerapis.firebaseapp.com/#!/Report/get_report_mtbd_vehiclehealth_vehiclehealthsummary_vehicleRegNo) | 2.09 |
| ~~8.04~~ | ~~Report~~ | ~~MTBD: Vehicle Movement should send the following information~~  ~~Label : Distance Travelled, value: 1500 KM~~  ~~Label: Average Speed, value: 45KMPH~~  ~~Label : Driving Time, value: 250 Hours~~  ~~Label: Idle Time, value: 15 Hours~~  ~~Label: Usage Time, value: 225 Hours~~ |  |
| 8.05 | [Report](https://swaggerapis.firebaseapp.com/" \l "!/Report/get_reportSummary_platform_platform_health)  [Fleet Health](https://swaggerapis.firebaseapp.com/" \l "!/Report/get_reportSummary_platform_platform_health) | Fleet Health to be returned for a given platform (mtbd/farm/mce/…)  Expected the numbers in groups of Good, Warning and Bad | 1.18 |
| 8.06 | [Report](https://swaggerapis.firebaseapp.com/" \l "!/Report/get_reportSummary_platform_platform_status)  [Fleet Status](https://swaggerapis.firebaseapp.com/" \l "!/Report/get_reportSummary_platform_platform_status) | Fleet Status to be returned for a given platform.  Expected the numbers in groups of Running, stopped, idle, not used and unknown | 1.18 |
| 8.07 | ReportSummary/**FARM** | **If the Platform is FARM,** report summary should contain  Vehicle Usage:  Cumulative Engine Hours: HH:MM  Cumulative PTO Hours: HH:MM  No. of Geofence Violated: Number  Vehicle Details:  Vehicle Number : AB 12 CD 3456  Last Used On : Date, Time  Last Known Location: Address  Usage hours report:  Pie chart : Engine hours, Halt hours (per week/month)  Vehicle Utilization Report:  Graph: Engine Hours, PTO hours vs. Time (per week/month)  Engine Hours Report:  Cumulative Engine Hours Vs. RPM (per week/month) | 3.04 |
| 9.01 | GeoFence | Create GeoFence. The API takes 3 or more geographical points as input along with a unique name. Points should be within Indian Political boundaries. None of the points should be in ocean. | 2.01 |
| 9.02 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/post_geofencehttps://swaggerapis.firebaseapp.com/) | Read GeoFence details. This provides points for the named geofence | 2.02 |
| 9.03 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/patch_geofence_name) | Update the GeoFence. Add new points. Change existing points. Similar condition checks like 9.01 | 2.03 |
| 9.04 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/put_geofence_name_vehicles) | Assign GeoFence to one or more vehicles. Parameters are geofence-name, start date, end date, start time,end time, inward/outward and list of vehicles | 2.04 |
| 9.05 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/delete_geofence_name) | Delete a named geofence. Cascade to mapping of vehicles to the corresponding geofence | 2.05 |
| 9.06 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/get_geofence_vehicles) | Get Geofences for a given vehicle. Lists currently active geofences (based on the date) | 2.06 |
| 9.07 | [GeoFence](https://swaggerapis.firebaseapp.com/#!/GeoFence/get_geofence_name_vehicles) | Get vehicles mapped to a given geofence with dates. | 2.07 |
| 10.01 | RoutePlan | Get all the available plans |  |
| 10.02 | RoutePlan | Create a route plan. Routeplan takes options shortest/fastest/custom. In custom, for each point, there will be *associated priority (None/1/2/3/4/5…) , pickup/delivery, location, contact*  Name of the route  Toll Charges and expenses to be entered as “expenses”. |  |
| 10.03 | RoutePlan | Assign Route to one or more vehicles. Parameters are geofence-name, start date, end date, start time,end time and vehicle registration numbers. This creates one trip per vehicle for the given dates. |  |
| 10.04 | RoutePlan | View assigned trips for a vehicle. This should include geofences in the route for the date of travel. If it is a completed trip, show geofences active on that dates for the given geographical area of the route. This should include geofences active at that time, but deleted later |  |
| 10.05 | RoutePlan | Get vehicles (trip details) mapped to the given routeplan |  |
| 10.06 | RoutePlan | Update ongoing trip. Similar to Create Route. But applicable with existing points which are not marked with delivery/pick up completed could be either deleted. Or Additional pick up/delivery to be added. This has to create a new entry in the routeplan. |  |
| 10.07 | RoutePlan | Live Trip details. Get the details/status of ongoing trip (started, delivery status, time) |  |
| 10.08 | RoutePlan | Delete Route. Cascade to mapping of vehicles to the corresponding route for ongoing/upcoming |  |
| 11.01 | T[ripManagement](https://swaggerapis.firebaseapp.com/#!/TripManagement/get_trip_tripName_vehicleRegNo) | Get active trip for a vehicle. Returns the status, current waypoints for the route, pickup/delivery details | 2.10 |
| 11.02 | [TripManagement](https://swaggerapis.firebaseapp.com/#!/TripManagement/put_trip_tripName_vehicleRegNo_status) | Status update: start trip. Update the status of trip as **started** | 2.11 |
| 11.03 | [TripManagement](https://swaggerapis.firebaseapp.com/#!/TripManagement/put_trip_tripName_vehicleRegNo_status) | Status update: complete the trip. Update the status of trip as **completed** | 2.12 |
| 11.04 | [TripManagement](https://swaggerapis.firebaseapp.com/#!/TripManagement/put_trip_tripName_vehicleRegNo_activity) | Status update: completed a delivery. For a given trip for the vehicle, update a delivery/pick-up as **done**.  ActivityID is like 10th point : pickup  Status : done/failed  Done in case of completed and failed if it is not possible in current trip | 2.13 |
| 11.05 | TripManagement | Live update on the notification in case of trip update from the server.  **Any update on the trip, has to be sent as push notification to the client** |  |
| 11.06 | TripManagement | Expense Summary. For each trip (on-going/completed), get the expenses (as Label, Value Pairs).  CreatedBy value to be taken from the current login.  This is to be linked to the Expenses for vehicle with active date and time of the trip |  |
| 12.01 | [BreakDownAssist](https://swaggerapis.firebaseapp.com/#!/BreakDown/put_breakDown_vehicleRegNo) | Given the vehicle registration number, if the platform permits it, call a breakdown assist. This will cascade to the corresponding Customer Care Centre | 1.14 |
| 12.02 | BreakDownAssist | Get a list of parts available with dealers. |  |
| 12.03 | BreakDownAssist | Get a list of parts available with dealers. For a given spare, get the nearest dealer address, and availability |  |
| 12.04 | BreakDownAssist | Order a part. Should accept one value from list of options or “Others”. Additional information to be captured as “description” of service required, nearest landmark/location for arranging the service. |  |
| 12.05 | BreakDownAssist | Order a part. Should cascade to dealer management network with selected dealer, location of vehicle, landmark and additional information. Should create a request number for tracking |  |
| 13.01 | LocalRepair | Create a local repair entry with date, summary/description, expense amount, repair station details (address, phone) |  |
| 13.02 | LocalRepair | Get a list of high level local repair types. Eg: Battery, Tyre Puncture, JumpStart… |  |
| 13.03 | LocalRepair | Get details of local repair done for the vehicle. “Category/Title: Battery”, Date of repair, amount, repair station details |  |
| 14.01 | [Expense](https://swaggerapis.firebaseapp.com/#!/Expense/put_expense_vehicleRegNo) | Add incidental expenses by the driver. This for an ongoing/completed trip.  Inputs will be date, time, description, amount and the vehicle number. Created by field to be taken from logged in user.  New expenses should accumulate with existing expenses against label/category. **Total 12 categories to be shared by MTBD team.**  Linking the expense to a trip should be done based on date, time and vehicle number. | 1.15 |
| 14.02 | Expense | View incidental expenses for ongoing/completed trips |  |
| 14.03 | [Expense](https://swaggerapis.firebaseapp.com/#!/Expense/get_expense_vehicleRegNo) | View incidental expenses for a given vehicle for a duration (start date, end date with time)  *CreatedBy* field needs to be taken from login on creation. Client can pass empty string  Expenses should return zero values for all category if no records found. | 1.16 |
| 14.04 | [Expense](https://swaggerapis.firebaseapp.com/#!/Expense/get_expenseCategory_vehicleRegNo) Category for MTBD | View expense types for a given vehicle.  1. Fuel Cost  2. Toll Tax+ Entry tax  3. Police +RTO  4. Roti + Tea  5. Commission  6. Prasad, Oil and Soap  7. Air, Grease and Cloth  8. Weigh Bridge  9. Phone, Parking, Service and Guide  10. Salary  11. Other expenses | 1.17 |
| 14.05 | Create Expense Category for a given platform |  |  |
| 15.01 | [Reminders](https://swaggerapis.firebaseapp.com/#!/Reminder/get_reminders) | Get all the reminders based on previous configuration for each vehicle | ~~1.18~~ |
| 15.02 | Reminders | For a given vehicle, get all the reminders. Send additional information on fields which are editable for the current user |  |
| 15.03 | Reminders | For a given vehicle, update the reminders |  |
| 16.01 | [LocationHistory](https://swaggerapis.firebaseapp.com/#!/LocationHistory/get_locationHistory_vehicleRegNo) | For a given vehicle, get the waypoints along with date and timestamp. Inputs are start date, time and end date time.  In future, the geofences active for that particular travel has to be listed | 1.19 |
| 17 | Engine management | Get the list of Engine Properties(Engine operating Hours with load,Idle engine hour,Engine temperature,oil pressure) |  |
| 18.01 | [Driver](https://swaggerapis.firebaseapp.com/#!/Profile/get_driverProfiles) | Get the list of drivers | 1.20 |
| 18.02 | profile management  1. [View](https://swaggerapis.firebaseapp.com/#!/Profile/get_driverProfiles_driverId)  2[. Add](https://swaggerapis.firebaseapp.com/#!/Profile/post_driverProfiles_driver)  3[. Update](https://swaggerapis.firebaseapp.com/#!/Profile/patch_driverProfiles_driverId)  4. [Delete](https://swaggerapis.firebaseapp.com/#!/Profile/delete_driverProfiles_driverId) | For MTBD, driver profile (add, view, update, edit and delete).  There are 4 APIs  Delete a driver to be cascaded to un-map vehicle(s) before deleting | 1.21 |
| 19.01 | profile management  1. [View](https://swaggerapis.firebaseapp.com/#!/Profile/get_profile)  2. [Update](https://swaggerapis.firebaseapp.com/#!/Profile/patch_profile_userId) | Current user profile(view, update)  Two APIS (view and update) | 1.22 |
| ~~20.01~~ | ~~Track vehicle~~ | ~~For a given vehicle, get the list of locations for given date and time.~~ |  |
| 21.01 | [driver to vehicle mapping](https://swaggerapis.firebaseapp.com/#!/Vehicle/post_vehicleMappingWithDriver_vehicleRegNo_driverId) | Mapping the driver to vehicle (vehicle list and driver list). Assign mapping based on inputs.  **If no value provided for the driver, un-map the vehicle** | 1.23 |
| 22.01 | Alert configuration | Alert configuration to the vehicle (depends upon role  Editable). Eg: engine temperature, speed. |  |

## Non-Functional Requirements

|  |  |
| --- | --- |
| 1 | All the APIs to undergo security audit |
| 2 | Performance reports on API along with concurrent user lists (10,100,1000, 10000) to be done and made available |
| 3 | Coding standards to be followed based on the documentation |

# Annexure: I

# Annexure: II

END OF DOCUMENT