Combined Promoter App Solution Design

Airtel Payments Bank

V0.2

This document contains proprietary information of HCL Technologies Ltd. and the filled in information would be for the exclusive use of Airtel Payments Bank and HCL only. No part of this document may be reproduced, stored, copied, or transmitted in any form or by means of electronic, mechanical, photocopying or otherwise, without the express consent of Airtel Payments Bank or HCL Technologies Ltd.

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Prepared/Modified By** | **Reviewed By** | **Significant Changes** |
| 0.1 | 14-Oct-2020 | Dharmesh Yadav |  | Initial Draft |
| 0.2 | 19-Oct-2020 | Dharmesh Yadav |  | Included Retailer promoter and Merchant promoter flows in Solution Design |

Table of Contents

[Revision History 2](#_Toc53648529)

[1 Introduction 3](#_Toc53648530)

[1.1 Purpose of Document 3](#_Toc53648531)

[1.2 Existing Apps & Portal 3](#_Toc53648532)

[1.2.1 Retailer Promoter App 3](#_Toc53648533)

[1.2.2 Merchant Promoter App 4](#_Toc53648534)

[1.2.3 Retailor Promoter Portal 4](#_Toc53648535)

[1.3 Project Design Guidelines 5](#_Toc53648536)

[1.4 Work Scope 5](#_Toc53648537)

[1.5 Roles & Responsibilities: 5](#_Toc53648538)

[1.5.1 Product Owner (APB) 6](#_Toc53648539)

[1.5.2 Scrum Master (HCL) 6](#_Toc53648540)

[1.5.3 Development Team (HCL) 6](#_Toc53648541)

[2 Solution Design 7](#_Toc53648542)

[2.1 Promoter App Design 7](#_Toc53648543)

[2.1.1 Promoter App Tech Architecture 9](#_Toc53648544)

[2.1.2 App Distribution Mechanism 12](#_Toc53648545)

[2.2 Promoter Portal Design 13](#_Toc53648546)

[2.2.1 Promoter Portal Tech Architecture 14](#_Toc53648547)

[2.3 APIs/Services Design 20](#_Toc53648548)

[3 Combined Promoter App Tech Stack 28](#_Toc53648549)

[4 Product Execution 29](#_Toc53648550)

[4.1 Discovery Sessions (APB) 29](#_Toc53648551)

[4.2 EPIC (Feature) (APB) 29](#_Toc53648552)

[4.3 User story (APB) 29](#_Toc53648553)

[4.4 Backlog grooming (APB) 29](#_Toc53648554)

[4.5 Estimation (HCL) 30](#_Toc53648555)

[4.6 Prioritize backlog (APB) 30](#_Toc53648556)

[5 Sprint planning (HCL) 31](#_Toc53648557)

[6 Sprint execution (HCL) 32](#_Toc53648558)

[6.1 Timing 32](#_Toc53648559)

[6.2 Sprint theme 32](#_Toc53648560)

[6.3 Daily Scrum 32](#_Toc53648561)

[6.4 Task Board 32](#_Toc53648562)

[6.5 User Stories in sprint 32](#_Toc53648563)

[6.6 Sprint demo 33](#_Toc53648564)

[6.7 Retrospective 33](#_Toc53648565)

[7 Deliverables (HCL) 34](#_Toc53648566)

[7.1 Product increment 34](#_Toc53648567)

[7.2 Unit test 34](#_Toc53648568)

[7.3 Code coverage report 34](#_Toc53648569)

[7.4 Test Automation 34](#_Toc53648570)

[7.5 Demo 34](#_Toc53648571)

# Introduction

## Purpose of Document

Airtel Payments Bank has two kinds of customer-facing partners offline - merchants and retailers. While merchants accept payments through QR codes, do recharges for customers, retailers primarily help open Savings Accounts for customers and provide other services such as remittances, cash withdrawals etc.

Currently there are two distinct sales teams with their own hierarchies looking after retailers and merchants respectively. A promoter is the last mile in this hierarchy responsible for servicing the retailers and merchants. Two different apps are used by promoters for retailers and promoters for merchants.

The promoter app for merchants was developed in-house and the one for retailers was developed for Ernst and Young. A lot of functions that the app serves for these sets of promoters is common.

Objective is to combine the two promoter apps to help offer a better experience and features to the promoters by also tapping into the synergies of a combined system.

This document sets out design and approach of the integration of two promoter apps.

## Existing Apps & Portal

The in-scope applications to be used by target stakeholders have an existing version. These versions will be used as the base for development work as part of this project.

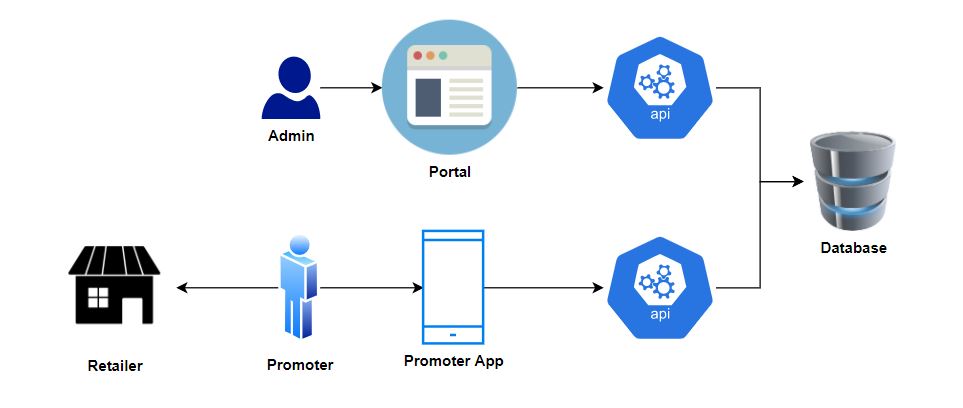
### Retailer Promoter App

Exclusively in use by Promoters for Retailers for purposes like:

* User Registration
* Mark Attendance
* View daily targets
* Viewing team
* Check in and check out Procedure
* Help and Support

This App is supported by a Web Portal Application.

Following is a high-level representation of Retailer Promoter App:



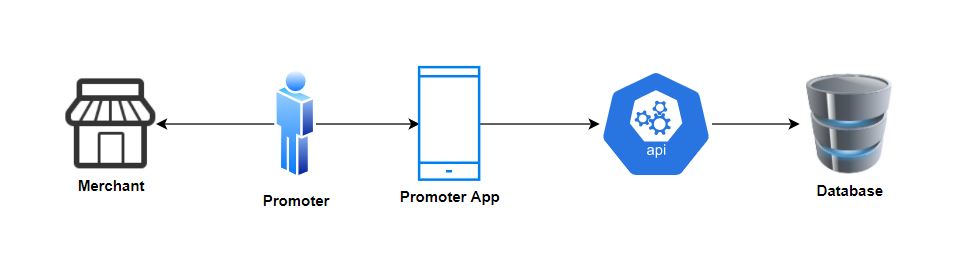
### Merchant Promoter App

Exclusively in use by Promoters for Merchants for purposes like:

* User Registration, Login
* Merchant On boarding
* Update Merchant Info and more

This App has NO Web Portal Interaction. It gets fed by Back End APIs.

Following is a high-level representation of Merchant Promoter App:



### Retailor Promoter Portal

Exclusively in use by Admin for Promoters for Retailers for purposes like:

* Data Upload
* Search
* Reports
* Training
* App Version and more

## Project Design Guidelines

At a high-level, project objective is to combine the two promoter apps to help offer a better experience and features to the promoters by also tapping into the synergies of a combined system.

Following are the high level guidelines for the project:

1. The Existing Retailor Promoter app will serve as the base for New Combined Promoter App.
2. The Merchant Promoter app will be decommissioned when the New App goes Live
3. The New Promoter App will integrate with a Web Portal to offer a set of identified features.

Following are the technical guidelines for the project:

1. The Retailor Promoter App is built as a Native Android App. This App will continue as the base App.
2. New Features / Components in the App will be built using React Native.
3. For existing use cases that require updates, Android Native will be the chosen technology.
4. The React Native Components will integrate with the base Android App to provide expected end-to-end functionality to the users.
5. The existing portal is developed using Angular 7. The portal will be updated/enhanced to include required features. All updates to the portal will be done in Angular 7.

## Work Scope

Following are included in the Work Scope of HCL Delivery team:

1. New app Development
2. Portal Enhancements
3. APIs/Services for App and Portal consumption.
4. Unit testing and SIT of in-scope items
5. Code Coverage > 70%

## Roles & Responsibilities:

|  |  |
| --- | --- |
| **HCL Team** | New App Development & Testing with considerations as mentioned in Section 1.3 |
| Portal enhancements to meet defined requirements. |
| Sprint Planning and execution in collaboration with APB Product Team. |
| Weekly Demos of progress to product team / stakeholders. |
| **APB Team** | Enable HCL team with access to APB dev environment, Tools and Frameworks. |
| Matured Sprint Stories, Sessions with HCL team on Story grooming, Sprint Planning |
| Sprint stories to be finalized 2 weeks ahead of the Sprint start date. |
| Access to test environment (if available). |
| Remediation of queries during dev phase. |

### Product Owner (APB)

* + Provide all necessary documentation on requirement.
  + Detail on requirements.
  + Explain use cases.
  + Backlog grooming.
  + Shall write user stories or help team write user stories.
  + Review & approve story changes done by HCL team.

### Scrum Master (HCL)

* + Plan & arrange communication between Product and development team.
  + Share meeting notes.
  + Help enabling development team on query resolution.
  + Ensure all discussions are planned before time, closed in time, all needed participants join.

### Development Team (HCL)

* + Enrich user stories with more details.
  + User story development
  + Peer code review
  + Testing (Feature, automation, regression)
  + Ask queries
  + Design solution, define technical approach to solution, explain solution
  + Demo completed work

Other roles not mentioned above can participate as well, but they would be helping entire team only without any direct responsibility in sprint execution.

# Solution Design

The deliverables of the project include the following artefact:

1. Promoter App
2. Promoter Portal
3. APIs/Services required to support App and Portal

**All User Interfaces (Ux Designs) of Promoter App and Portal will be provided by the APB Product team.**

## Promoter App Design

The New Promoter App is expected to serve requirements for both set of Promoters – Retailor as well as Merchant.

The App will support RBAC (Role Based Access Control).

The intended users will login into the App using Preconfigured Role provided by the Web Portal.

Following are the expected roles for the Promoter App:

1. Retailer Promoter
2. Merchant Promoter
3. Team Leader

The table below highlights the features that will be supported by the New Promoter App. A set of these features are already existing in the current Retailor App, few features will require enhancement, while the New Features will be developed.

Each feature is detailed on the action required for the New App:

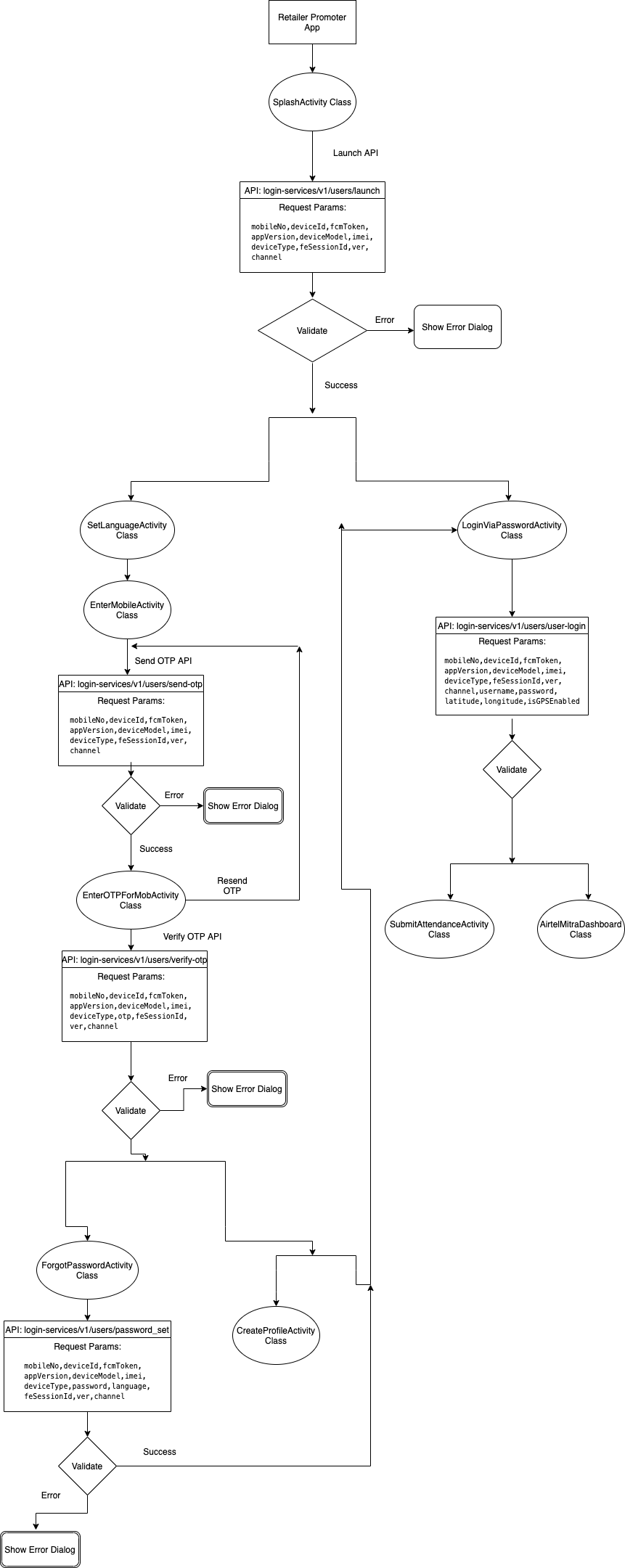
|  |  |  |  |
| --- | --- | --- | --- |
| S.No | App Feature | Detail | Type of Change / Enhancement |
| 1 | **Registration** | A promoter must be able to register himself on the app | API will be modified. |
| 2. | **Login** | A promoter must be able to login on the app. | API will be modified. |
| 3 | **Mark Attendance** | A promoter must be able to mark his attendance on the app. | No change |
| 4 | **View Attendance** | A promoter must be able to view his attendance history on the app. | No change |
| 5 | **View My Outlet** | A promoter must be able to view the list of outlets assigned to him, achievements with respect to each outlet and the distance of each outlet on the app. | No change |
| 6 | **Today’s Plan** | A promoter must be able to see the outlets assigned to him as his daily target. | No change |
| 7 | **View My Team** | A TL must be able to see the list of all the promoters mapped under him on the app. TL must also be able to see the KBI breakdown of each promoter mapped under him. | No change |
| 8 | **Check-in/Check-out** | A promoter must be able to Check-in & Check-out at every outlet they visit. This will both be manual and automatic process. | No change |
| 9 | **Capture Compliance** | A promoter must be able to capture compliance (pictures) of the outlet which will be displayed under the check-in page. | No change |
| 10 | **Dashboard/Homepage** | A promoter must be able to view dashboard on the app. | Frontend will be modified.  API will be modified. |
| 11 | **Rewards & Recognition** | A promoter will be able to see all the rewards or recognition they got on the basis of their performance. | No change |
| 12 | **Training** | Online training for promoters can be conducted and the details of which will be displayed under this module. A promoter must be able to see the list of all the training activities and the training dashboard with assigned date, due date, training score etc. | No change |
| 13 | **Help & Support** | Under this module promoter can raise complaints on the behalf of retailers/merchants. They can see the list of all the previously raised complaints with their status. | No change |
| 14 | **Notifications** | A notification window where user can see all the push notifications and other news. | No change |
| 15 | **Search (for Admin)** | Admin must be able to search within the app. | No change |
| 16 | **On-board Merchant** | A promoter must be able to onboard a Merchant on the app. | Frontend will be modified.  Existing Merchant On-board project will be merged with retailer promoter app. |
| 17 | **Link QR Code** | A promoter must be able to link QR code with Merchant account on the app. | Frontend will be modified. |
| 18 | **Update Merchant** | A promoter must be able to update the details of merchants on the app. | Frontend will be modified. |
| 19 | **Upgrade Merchant** | A promoter must be able to upgrade a merchant on the app. | This will be a completely new implementation. |
| 20 | **Merchant Business Summary** | A promoter must be able to view reports related to Merchant business on the app. | Frontend will be modified. |
| 21 | **FASTag** | A promoter must be able to sell FASTag on the app. | This will be implemented as a webview component. |
| 22 | **Lending Kart** | A promoter must be able to sell insurance to merchant on the app. | This will be implemented as a webview component. |
| 22 | **Logout** | A promoter must be able to logout of the app. | No change |
| 23 | **Personal Details - photo and name** | A promoter must be able to view his photo and name on the app. | No change |
| 24 | **Lead Generation** | A promoter must be able to log Leads in this module. | No change |
| 25 | **Attendance Approval** | A promoter must be able to approve attendance of its team members. | No change |

### Promoter App Tech Architecture

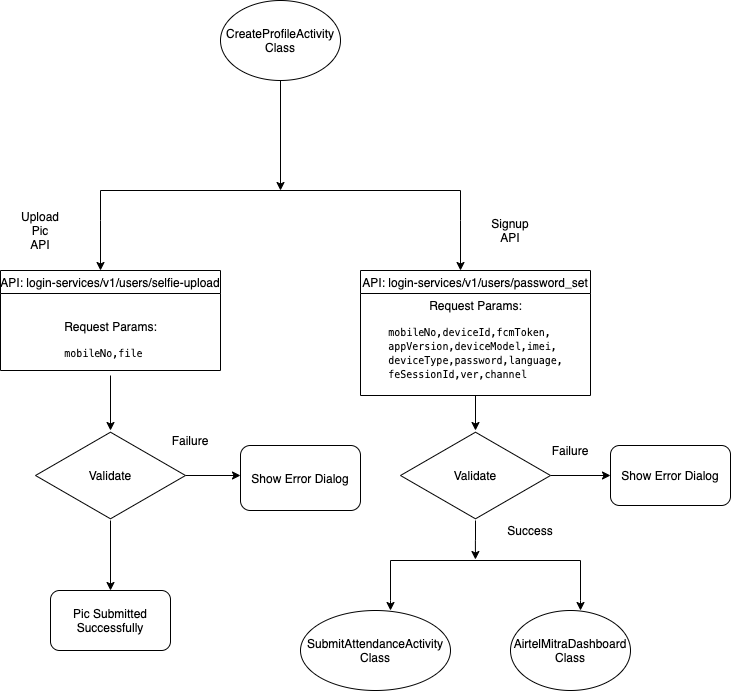
React & Android Integration

(WIP)

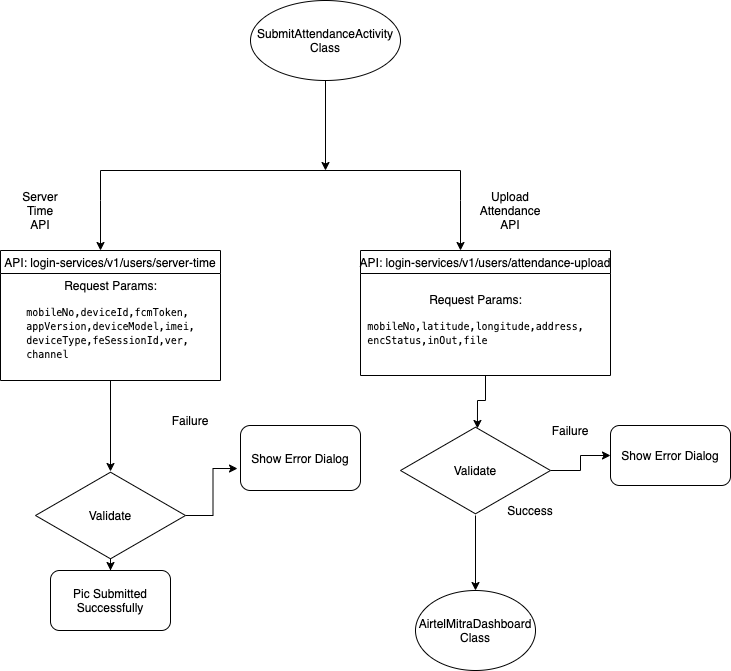
#### Combined Promoter App Login Flow



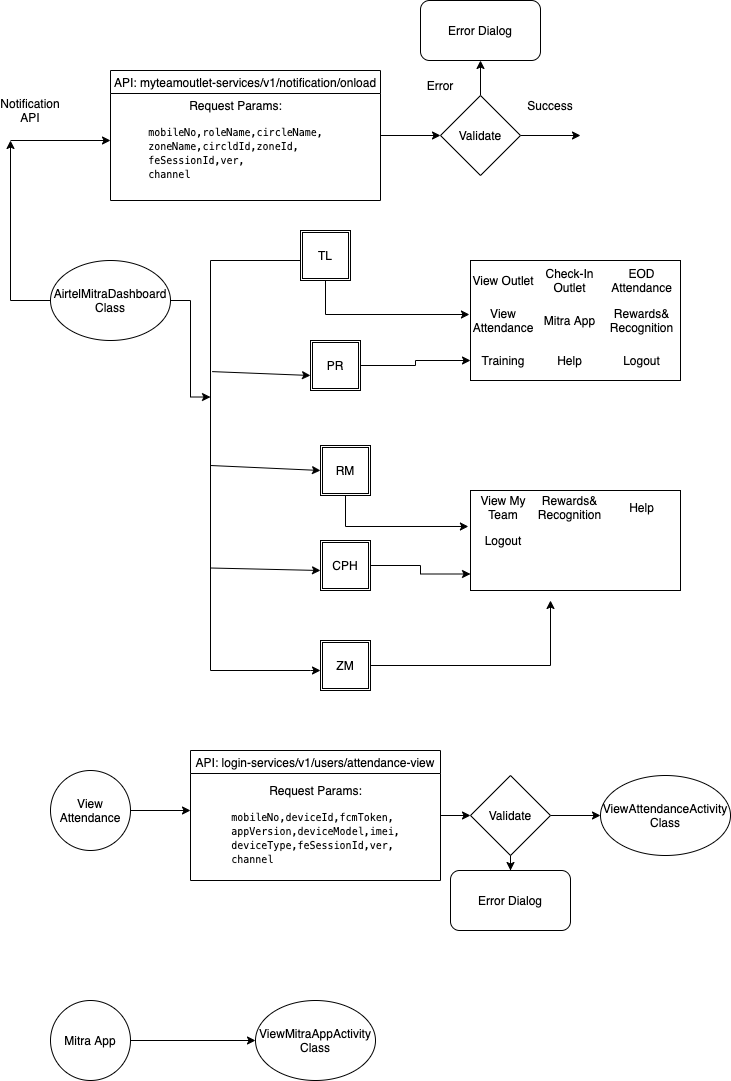
#### Combined Promoter App Profile Activity Flow



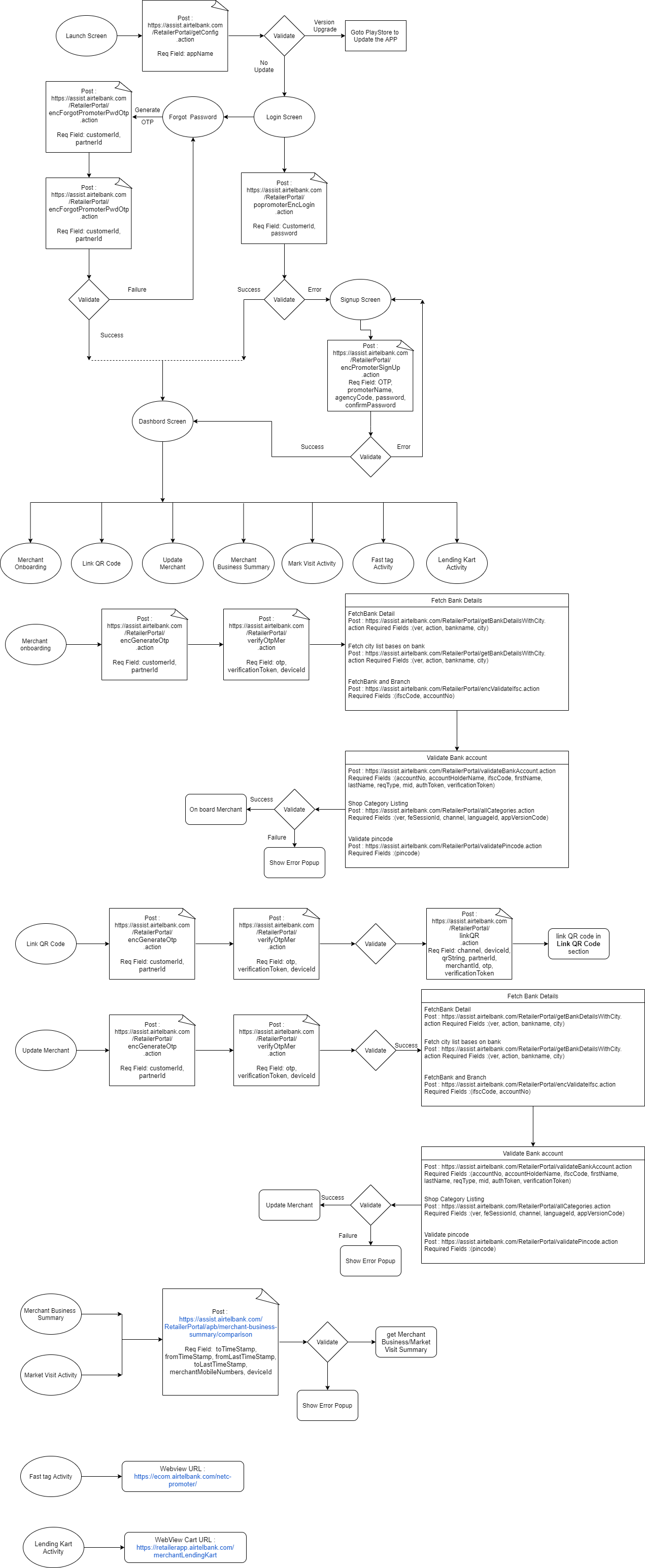
#### Combined Promoter App Attendance Activity Flow



#### Airtel Mitra Dashboard Flow



#### Merchant Promoter Features



### App Distribution Mechanism

This is expected to be APB responsibility. Detailing to be done based on discussion with APB team.

## Promoter Portal Design

The New Promoter Portal is expected to serve requirements for both set of Promoters – Retailor as well as Merchant.

The intended user for Portal will be Admin. Admin will login into the Portal to perform various actions.

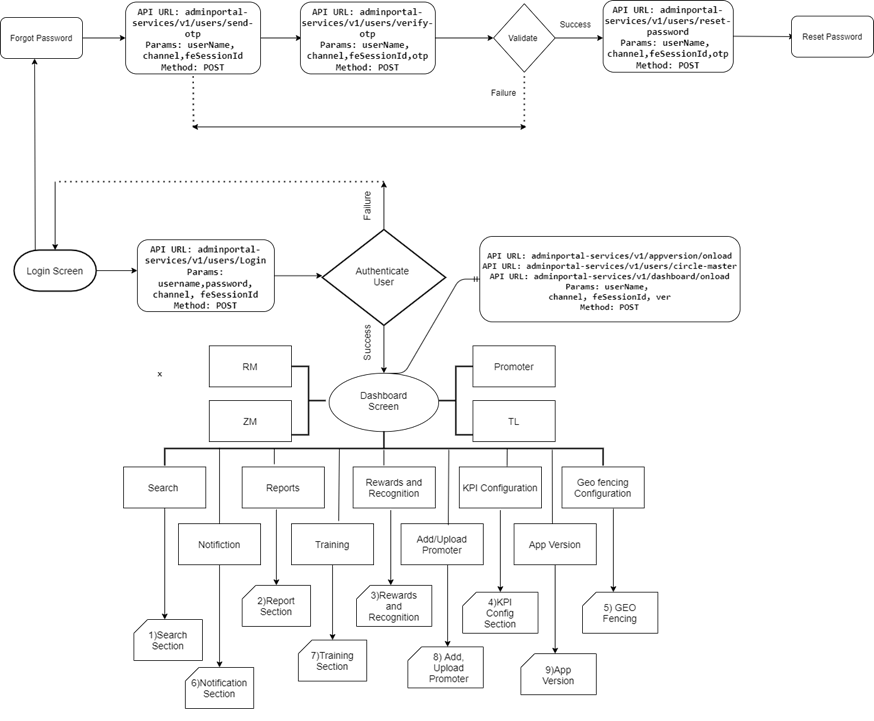
The table below highlights the features that will be supported by the Promoter portal.

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Portal Feature | Detail | Type of Change / Enhancement |
| 1 | **Search** | Displays data of ZM/TL/PM or individual of respective circle and zone. | No change |
| 2 | **Reports** | Reports related to attendance, compliance, market visit and training could be generated. | No change |
| 3 | **Rewards and Recognition** | Rewards and recognition could be sent to All/Circle Wise/Zone Wise or Individual group/person. | No change |
| 4 | **KPI Configuration** | KPI configuration information could be added. | No change |
| 5 | **Geo Fencing Configuration** | Geo Fencing configuration information could be added. | No change |
| 6 | **Notification** | Notification could be sent to All/Circle Wise/Zone Wise or Individual group/person. | No change |
| 7 | **Training** | Training invite could be sent to All/Circle Wise/Zone Wise or Individual group/person. | No change |
| 8 | **Add/Upload Promoter** | Promoter and KPI mapping could be added or uploaded. | CSV and upload-retailer API will be modified to handle Merchant Promoter data. |
| 9 | **App Version** | Information of app updation could be sent through this feature. | No change |

### Promoter Portal Tech Architecture

Promoter portal is implemented as a web application in Angular 7. It uses Spring boot micro services as APIs.

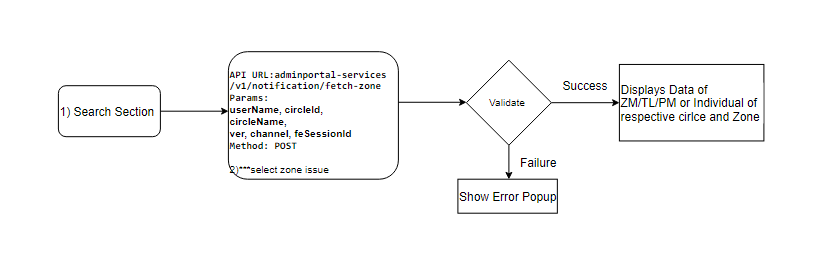
Below is the representation of Portal Architecture along with the APIs being used:



Each feature in above representation is described through the individual flow diagrams below:

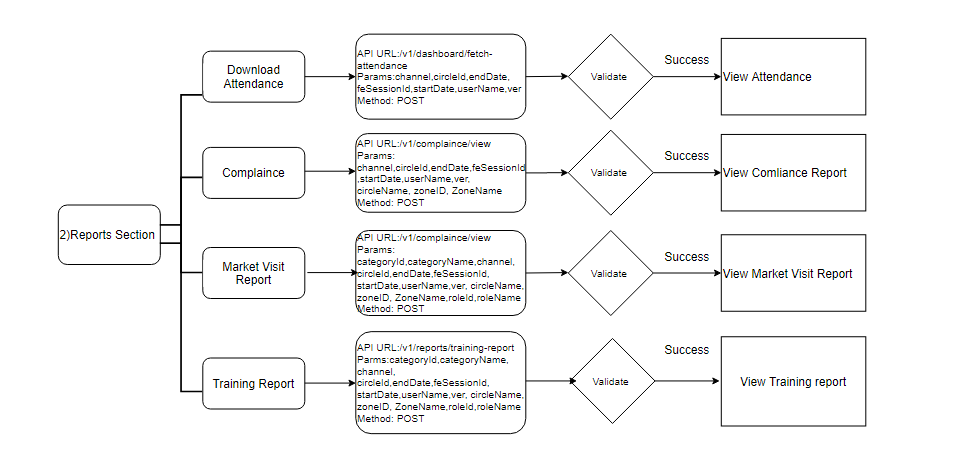
#### Use Case: Do search

|  |  |
| --- | --- |
| Use Case | Do search |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to search section. Admin is able to see data of ZM/TL/PM or Individual of respective circle and Zone. |



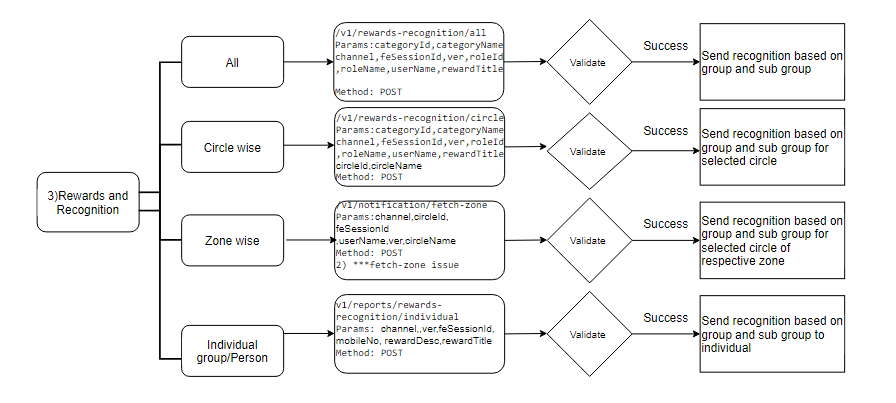
#### Use Case: Display reports

|  |  |
| --- | --- |
| Use Case | Display reports |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to reports section. Admin is able to see reports related to attendance, compliance, market visit and training. |



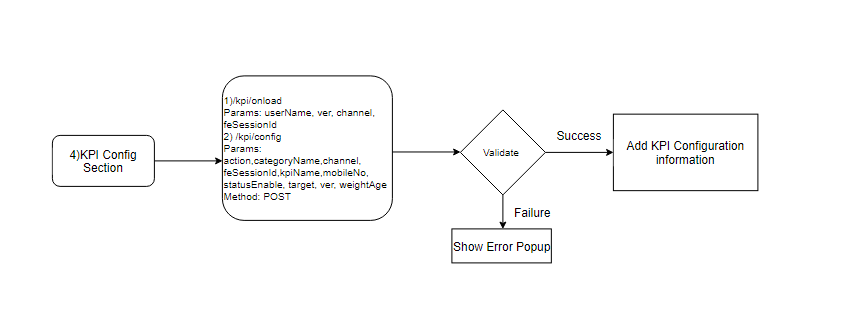
#### Use Case: Send rewards and recognition

|  |  |
| --- | --- |
| Use Case | Send rewards and recognition |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to rewards and recognition section. Admin is able to send recognition to a group or a sub group from here. Admin is also able to send recognition to a selected circle, a selected circle of respective zone or an individual. |



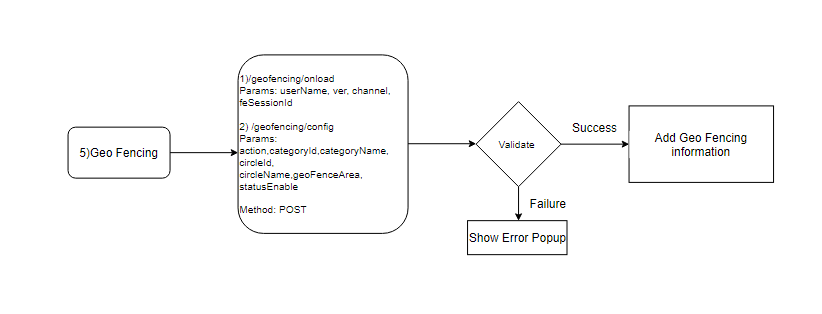
#### Use Case: Add KPI configuration information

|  |  |
| --- | --- |
| Use Case | Add KPI configuration information |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to KPI config section. Admin is able to add KPI configuration information. |



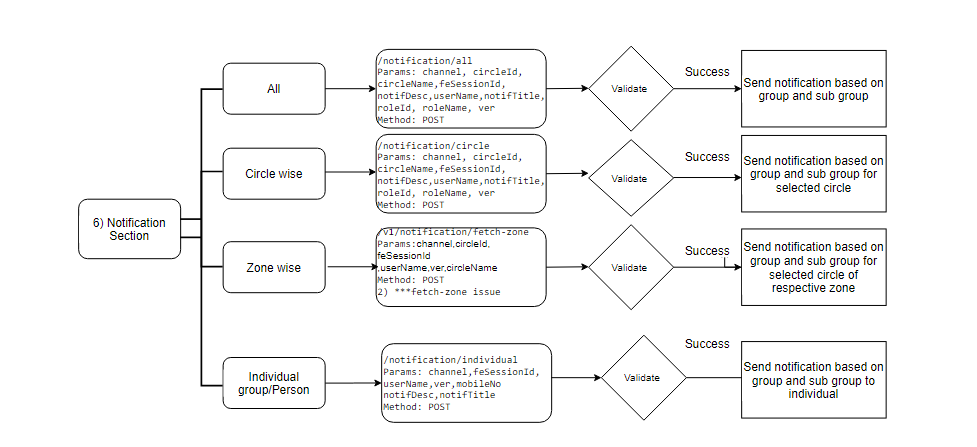
#### Use Case: Add Geo Fencing information

|  |  |
| --- | --- |
| Use Case 5 | Add Geo Fencing information |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to Geo Fencing section. Admin is able to add Geo Fencing information. |



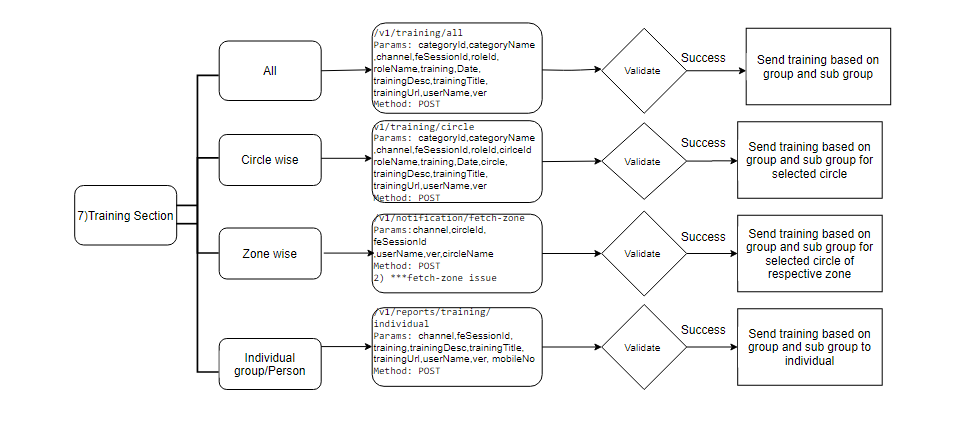
#### Use Case: Send Notification

|  |  |
| --- | --- |
| Use Case 6 | Send Notification |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to Notification section. Admin is able to send notification to an individual. Admin can also send notification to a group or a sub group. Admin is also able to send notification based on circle or selected circle of respective zone. |



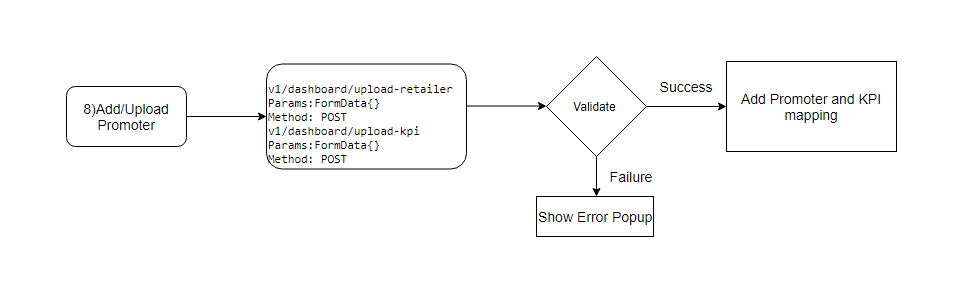
#### Use Case: Send Training

|  |  |
| --- | --- |
| Use Case 7 | Send Training |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to Training section. Admin is able to send training to an individual. Admin can also send training to a group or a sub group. Admin is also able to send training based on circle or selected circle of respective zone. |



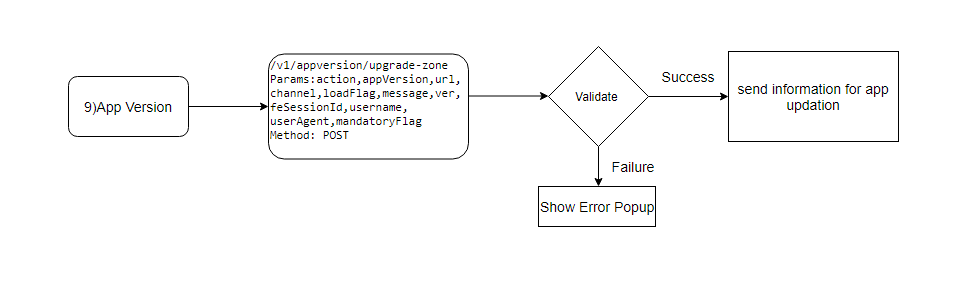
#### Use Case: Add Promoter

|  |  |
| --- | --- |
| Use Case 8 | Add Promoter |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to Add/Upload Promoter section. Admin is able to upload promoter data in CSV format. |



#### Use Case: Send information for app updates

|  |  |
| --- | --- |
| Use Case 9 | Send information for app updates |
| Actor | Admin |
| Basic Flow | Admin logs into web portal. Navigates to App version section. Admin is able to send information for app updates to the users. |



## APIs/Services Design

Existing promoter apps are being supported by a number of APIs. Some of these APIs will need updates in order to work for the new combined promoter app.

Below is a list of APIs being used for **Login** functionality:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | API Name | API Parameters | Detail | Type of Change / Enhancement |
| 1 | **Launch** | Post : /v1/users/launch(mobileNo, deviceId, fcmToken, appVersion, deviceModel, imei, deviceType)  **Required field** : deviceId | This API will be used to check Mobile App Update version |  |
| 2 | **Send OTP** | Post : /v1/users/send-otp (mobileNo, deviceId, fcmToken, appVersion, deviceModel, imei, deviceType)  **Required field** : mobileNo | This API is used to send OTP. In login and forget password |  |
| 3 | **Verify OTP** | Post :v1/users/verify-otp (mobileNo, deviceId, fcmToken, appVersion, deviceModel, imei, deviceType, otp, otpVerficatonCode, isUserDeviceExist, isUserExist)  **Required field** : mobileNo, otp, otpVerifCode, isMobileNoExist, isUserDeviceExist, is required | This API is used to verify OTP. In login and forget password. |  |
| 4 | **Login** | Post :v1/users/login (mobileNo, deviceId, fcmToken, appVersion, deviceModel, imei, deviceType, password, latitude, longitude, isGpsEnabled, userName)  **Required field** : mobileNo, password deviceId is required | Enter Password in order to login into APP |  |
| 5 | **Selfie-upload** | Post :v1/users/selfie-upload (mobileNo, file) | This API is used to upload Selfie file in Sign Up section |  |
| 6 | **password-set** | Post :v1/users/password-set (mobileNo, deviceId, fcmToken, appVersion, deviceModel, imei, deviceType, password, language)  **Required field**: mobileNo, password, deviceId is required | Once selfie upload then this API is used to set password of user for registration. |  |
| 7 | **attendance-upload** | Post :v1/users/attendance-upload  **Required field** : (mobileNo, latitude, longitude, address, file, encStatus, inOut) | Promoter will upload the attendance by sending pic. |  |
| 8 | **attendance-view** | Post :v1/users/attendance-view (mobileNo) | API to fetch Retailer Pramoter past attandance record. |  |

Below is a comprehensive list of APIs used for **airtel-mobile-crm** services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | API Name | API Parameters | Detail | Type of Change / Enhancement |
| 1 | **Fetch Complaints Faq List** | Post : /v1/help/fetch-complaints  **Required field** : mobileNumber | This API will serve help section for Retailer & Merchant both and for CRM Portal & Mobile application. |  |
| 2 | **Register Customer Complaint** | Post : /v1/help/customer-complaints  **Required field** : loginNo, actor\_type, issueName, customerNo, description | This API will register a fresh complaint in the system. |  |
| 3 | **Fetch Complaints against usertype** | Post path : /v1/help/fetch-complaints  **Required field** : loginNo, actor\_type, issueName, customerNo, description | This API will fetch all complaints registered for a particular userType. |  |

Below is a list of APIs used for **My Team Outlet** Service:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | API Name | API Parameters | Detail | Type of Change / Enhancement |
| 1 | **Fetch Notifications** | Post : /v1/notification/onLoad  **Required field** : mobileNo, roleName, circleName, zoneName | This API will serve on Click Notification icon all notification will be visible.  . |  |
| 2 | **Get Notification details** | Post : /v1/notification/onClick  **Required field** : mobileNo, roleName | This API will fetch the particular notification details. |  |
| 3 | **Update GPS On/Off Status** | Post : /v1/notification/gpsOff  **Required field** : mobileNo, roleName | This API will update the user GPS status in system. |  |
| 4 | **Fetch Training details** | Post : /v1/training/my-training  **Required field** : mobileNo | This API will get the details of training against userType. |  |
| 5 | **Fetch Reward and Recognition details** | Post : /v1/rewards-recognition/onload  **Required field** : mobileNo, roleName, circleName, zoneName | This API will get the details of reward & recognition against userType. |  |
| 6 | **Activity Tracker** | Post : /v1/myoutlet/activity-tracker  **Required field** : SnapWorkRequest Model | This API is used to store seqId, mobileNo, latitude, longitude Of user. |  |
| 7 | **Check In** | Post : /v1/dashboard/check-in  **Required field** : mobileNo | This API is used to checkIn save lat, lng, address, mobileNumber, name |  |
| 8 | **Check Out** | Post :/v1/dashboard/check-out  **Required field** : mobileNo | This API is used to checkOut save lat, lng, address, mobileNumber, name |  |
| 9 | **My Outlet View** | Post : /v1/myoutlet/view  **Required field** : mobileNo, roleId, roleName | Data will be fetched based on role PR/TL/ZM |  |
| 10 | **Fetch Retailer KPI Detail** | Post : /v1/myoutlet/kpi-details  **Required field** : mobileNo, roleId, roleName, userName | This API is used to get KPI details based on mobileNumber. |  |

Following is a list of APIs used for **Admin Portal**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | API Name | API Parameters | Detail | Type of Change / Enhancement |
| 1 | **Login** | Post :/v1/users/Login  **Required field** : (userName, password) | This API will be used to login in to portal | No change |
| 2 | **Send otp** | Post :/v1/users/send-otp  **Required field** : (mobileNo) | This API will be used in forgot password section to send otp | No change |
| 3 | **Verify otp** | Post: /v1/users/verify-otp  **Required field** : (mobileNo, otp, otpverifiyCode) | This API will be used in forgot password section to verify otp | No change |
| 4 | **Reset -password** | Post: /v1/users/reset-password  **Required field** : (userName, password) | This API will be used in forgot password section to reset password | No change |
| 5 | **Circle-master** | Post: v1/users/circle-master  **Required field** : (mobileNo) | This API will be used to get circle listing of user at dashboard | No change |
| 6 | **fetch attendance** | Post : /v1/dashboard/fetch-attendance  **Required field** : (circle, startDate, endDate) | This API will be used to get attendance data in Report section | No change |
| 7 | **Download attendance** | Post: v1/users/download-attendance  **Required field** : (encrandomkey, encString, encString) | This API will be used to download attendance into a file in Report section | No change |
| 8 | **logout** | Post : /v1/users/logout **Required field** :(userName) | This API will be used to logout from the admin portal | No change |
| 9 | **Appversion upgrade** | Post:/v1/appverson/upgrade **Required field** :(userName) | This API will be used to get app version in Dashboard section | No change |
| 10 | **Training all** | Post: /v1/training/all  **Required field** :(username, roleName, categoryName, trainingTitle, trainingDesc, trainingUrl, trainingDate) | This API will be used to send training to all registered users in Training section | No change |
| 11 | **Training circle** | Post: /v1/training/circle  **Required field** :(username, roleName, circleName, categoryName, trainingTitle, trainingDesc, trainingUrl, trainingDate) | This API will be used to send training to particular circle's users Training section | No change |
| 12 | **Training zone** | Post: /v1/training/zone **Required field** :(username, roleName, circleName, zoneName, categoryName, trainingTitle, trainingDesc, trainingUrl, trainingDate) | This API will be used to send training to particural zone's users Training section | No change |
| 13 | **Training upload** | Post : /v1/training/upload  **Required field** :(username, file, categoryName, categoryName, trainingTitle, trainingDesc, trainingUrl, trainingDate ) | This API will be used to send training to individual Training section | No change |
| 14 | **Dashboard Market onLoad Details** | Post : /v1/dashboard/market-onload  **Required field** : (userName) | This API will be used to get dashboard market onload detail in Report section | No change |
| 15 | **FetchDashboardMarketVisitDetails** | Post: /v1/dashboard/market-visit  **Required Field** : ( roleName, categoryId, circleId, zoneId, startDate, endDate) | This API will be used to get dashboard market visit details in Report section | No change |
| 16 | **DownloadMarketDetails** | Get : /v1/dashboard/download-market  **Required Field** : (encrandomkey , encString, sessionKey) | This API is used to get market details from CSV file. | No change |
| 17 | **FetchTrainingReportDetails** | Post : /v1/reports/training-report  **Required Field** : ( roleName, categoryId, circleId, zoneId, startDate, endDate) | This API is used to fetch Training details from DB in Report section. | No change |
| 18 | **DownloadTrainingDetails** | Get: /v1/reports/download-training  **Required Field** : (encrandomkey , encString, sessionKey) | This API is used to download Training details from DB in Report section | No change |
| 19 | **fetchComplianceDetail** | Post : /v1/compliance/view **Required Field** : (userName , circleId, zoneId, startDate, endDate) | This API is used to fetch Compliance Details from DB in Report section | No change |
| 20 | **dashboardOnLoadDetails** | Post : /v1/dashboard/onload **Required Field** : (userName ) | This API is used to fetch dashboard onload from DB in Dashboard section | No change |
| 21 | **downloadComplianceDetails** | Get : /v1/compliance/download **Required Field** : (encrandomkey, encString, sessionKey ) | This API is used to download compliance details | No change |
| 22 | **uploadRetailerCSVDetails** | Post : /v1/dashboard/upload-retailer  **Required Field** : (mobileNo, file ) | This API is used to upload the promoter detail from add/upload Promoter section | API will be modified to support data for Merchant Promoter. |
| 23 | **uploadKPICSVDetails** | Post : /v1/dashboard/upload-kpi  **Required Field** : (mobileNo, file, startDate, endDate) | This API is used to upload the KPI Config information from add/upload Promoter section | No change |
| 24 | **fetchGeoFencingOnloadDetails** | Post : /v1/geofencing/onload **Required Field** : (userName) | This API is used to get the onload data from the Geofencing section | No change |
| 25 | **geoFencingConfigDeatils** | Post : /v1/geofencing/config **Required Field** : (circleId, categoryId, geoFenceArea, action, statusEnable ) | This API is used to add the config detail on click add button in Geofencing section | No change |
| 26 | **kpiConfigurationDetails** | Post : /v1/kpi/config **Required Field** : (mobileNo, categoryName, kpiName, weightAge, target, statusEnable, action ) | This API is used to add the config detail on click add button in KPI Configuration section | No change |
| 27 | **fetchKpiConfigurationOnloadDetails** | Post :/v1/kpi/onload **Required Field** : (mobileNo) | This API is used to get the onload data from the KPI Configuration section | No change |
| 28 | **sendNotificationToAll** | Post :/v1/notification/all **Required Field** : (userName, roleName, roleId, notifTitle, notifDesc) | This API is used to send notification to all user in Notification Section | No change |
| 29 | **sendNotificationToCircle** | Post :/v1/notification/circle **Required Field** : (userName, roleName, roleId, circleId, circleName, notifTitle, notifDesc) | This API is used to send notification to circles associated user in Notification Section. | No change |
| 30 | **sendNotificationToIndividual** | Post :/v1/notification/individual **Required Field** : (userName, mobileNo, notifTitle, notifDesc) | This API is used to send notification to individual user in Notification Section | No change |
| 31 | **sendNotificationToIndividualCsv** | Post :/v1/notification/upload **Required Field** : (userName, file, notifTitle, notifDesc) | This API is used to send notification to individual from a file in Notification Section | No change |
| 32 | **sendNotificationToZone** | Post : /v1/notification/zone **Required Field** : (userName, roleId, roleName, circleId, circleName, zoneId, zoneName notifTitle, notifDesc) | This API is used to send notification to zones user in Notification Section. | No change |
| 33 | **populateZoneNamesBasedOnCircle** | Post : /v1/notification/fetch-zone  **Required Field** : (userName, circleId, circleName) | This API is used to get the zone name on the basis of circle in Notification Section | No change |
| 34 | **sendRewardsToAll** | Post : /v1/rewards-recognition/all  **Required Field** : (userName, roleName, categoryName, rewardTitle, rewardDesc) | This API is used to send Rewards to all users in Rewards and Recognition section | No change |
| 35 | **sendRewardsToCircle** | Post : /v1/rewards-recognition/circle **Required Field** : (userName, roleName, categoryName, rewardTitle, rewardDesc, circleName) | This API is used to send Rewards to circle user in Rewards and Recognition section | No change |
| 36 | **sendRewardsToZone** | Post : /v1/rewards-recognition/zone **Required Field** : (userName, roleName, circleName, zoneName, rewardTitle, rewardDesc, categoryName) | This API is used to send Rewards to zone user in Rewards and Recognition section | No change |
| 37 | **sendRewardsToIndividual** | Post : /v1/rewards-recognition/individual **Required Field** : (mobileNo, rewardTitle, rewardDesc) | This API is used to send Rewards to individual user in Rewards and Recognition section | No change |
| 38 | **sendRewardsToIndividualCsv** | Post : /v1/rewards-recognition/upload **Required Field** : (userName, file, rewardTitle, rewardDesc) | This API is used to send Rewards to user based on CSV file in Rewards and Recognition section. | No change |
| 39 | **dashboardMarketOnLoadDetails** | Post : /v1/rewards-recognition/reports-onload **Required Field** : (userName) | This API is used to get the onload data from the Market section | No change |

# Combined Promoter App Tech Stack

This section covers system requirements required for Combined Promoter App.

Following is the tech stack for Combined Promoter App system:

* Sprint Boot based Microservices
* Oracle DB
* Android Native
* React Native
* Struts

Since Promoter Portal is a web application, we have a different tech stack for it.

Here is the Promoter Portal tech stack:

* Angular 7
* Oracle DB
* Spring Boot based Microservices

# Product Execution

With **Backlogs**, we can quickly plan our project by adding user stories or requirements to product backlog. Once we have our plan in place, then only we can start driving code development efforts.

Activities those will take place around product backlog are:

## Discovery Sessions (APB)

Those shallnot be done needed for every feature but are meaningful where an entirely new feature or very complex feature is to be done. Here PO and dev team will find out what is feasible and how to do it. Both APB and HCL team will participate but owner would be APB.

## EPIC (Feature) (APB)

For big features which contains more than one user story work, we will create an EPIC which shall be linked to multiple user stories. This must have business cases and entire flow.

## User story (APB)

This shall be the smallest piece of work which generates some output from end user perspective and can be done by a single developer in a sprint. Must things to be there in user story are **Summary, Priority, Description, Acceptance Criteria’s.**

Additional things to add are

* Link to Epic
* Link to other stories/defects
* Attachments
* Videos.
* Test data needed
* Test environment if functionality can run in a specific environment only.

Some ground rules

* If any story estimated more than a what can be done in a sprint, it must be broken down into multiple stories.
* If any story is not completed due to any reason within due sprint, it will be carried forward to new sprint entirely.
* Story will follow the life cycle – **New, Development, Demo, Code review, Testing/Automation, UAT**.
* If story does not pass the demo or code review or UAT, it will be moved back to development.
* Change in Acceptance criteria will move the story back to backlog and it will be queued for backlog grooming and followed by an upcoming sprint.

## Backlog grooming (APB)

Before development team can start work on any item, they need more details and walk through. Details include flow, scope, acceptance criteria, test data. These sessions would be conducted by product owner(s) where PO would explain requirement, user story and acceptance criteria. Development team would ask questions, tells feasibility, talk little on technical solutions, give estimates.

## Estimation (HCL)

Development team will give estimates for each item discussed during backlog grooming discussions. Estimates will be relative and assuming end to end work for a work item. Estimates would be in **story points** Story points will use Fibonacci sequence (1, 2, 3, 5, 8, etc.).

* Estimates includes everything development, testing, automation and documentation.
* For a small piece of code change which is very complex and require search there could be a big number in story points.
* For lengthy work which is easy in complexity and be given small estimates.
* Story points can not be mapped to days at all.
* Estimation would be done with whole team present and agreed. We can use any method like poker planning, or everyone says their estimates, but idea is that everyone agrees to estimates.
* Estimates are not done assuming someone has already worked on similar thing or some things would be copy pasted from other implementations. Estimates are so any one of developers can take that user story and he writes things his own.

## Prioritize backlog (APB)

APB (PO) can reorder backlog items to create a prioritized list of work. Review and prioritize backlog with development team is expected so that they know what's most important to deliver next but can happen sometime offline as well.

# Sprint planning (HCL)

Sprint planning is essential to ensure what we plan for sprint, team can deliver it. There are two keys things in Sprint Planning.

* Team Velocity

Velocity determines how much work the team can accomplish in a sprint. In any sprint, the velocity is calculated as the number of story points completed per sprint. For sprint 1 we would define a number, based on outcome we would revise this number. If we can complete then we try to take little more next time, if unable to deliver we reduce this number next time. Once we get a stable number this becomes our velocity going forward.

* Team Capacity

We measure team’s available capacity for the sprint and tries to fill that capacity effectively without overburdening and under committing the team members. The capacity of the teams may vary from one sprint to another, depending on holidays, leaves, or other commitments.

Sprint planning activity would be done with entire team and as part of this we would achieve followings:

* Create new sprint and define time.
* Calculate velocity and capacity.
* Take groomed & estimated stories from backlog to current sprint.
* This would be done on day 1 of sprint.

# Sprint execution (HCL)

Every sprint, the team executes a mini-project into itself—it performs all the work necessary to deliver a potentially shippable product implement. The team's work would be guided by the sprint goal and sprint backlog.

## Timing

The majority of the team's in sprint would be spent in sprint execution. We will have **2 weeks** sprints. For a two-week sprint, sprint execution time would likely count for 8 to 8.5 days. Close to 2 days are invested in backlog grooming, sprint planning, demos, retrospective and other discussions.

## Sprint theme

Sprint theme is a way to ensure we are doing all related user stories for a feature or to an area in application. It would help to ensure end of sprint a flow is completed rather than parts of different flows. This also reduces risk of change for live application as we are making change in one area only.

## Daily Scrum

We will do a daily 20 minutes catch up to share with everyone what is daily progress. Some to things to look forward in these discussions are:

* APB Product Team and HCL Development team both will participate.
* Everyone will tell progress, next thing to do, challenges (if any).
* Updates are for and addressed to entire team not to scrum master or PO.

If needed a carry forward session can be done post everyone shares updates to continue discussion needed between 2 or more team members or between development team & product team.

## Task Board

We will use JIRA as task board, which will show backlog, current sprint. Current sprint will show progress in task board which includes pending things in sprint, in progress and completed things. It will also show owners for each story.

## User Stories in sprint

Every developer will work on a user story, no work without user story in JIRA would be allowed. For a user story in sprint followings would apply:

No developer will work on user story if

* User story is not groomed
* Doesn’t have acceptance criteria
* Doesn’t have estimates
* Not part of active sprint

Once taken into sprint and picked by developer

* Developer understand story, ask question if any.
* Confirm estimates are valid
* Add approach as a comment in user story (for complex stories only, not needed for small stories).
* Create sub tasks [Dev, Demo, Unit tests, Testing, Automation etc], estimates them.
* Update sub tasks once there is progress.
* Update screenshot/videos in user story at end of sprint (for complex stories only, not needed for small stories).

## Sprint demo

At the end of sprint, team will give a demo of all user stories to product team. This will be a planned session with whole development team, product team and other stockholders.

Small feedbacks would be fixed in sprint itself; large number of feedbacks means gap in requirement and implementation and same would cause user story to be moved to next sprint or backlog.

## Retrospective

At end of each sprint an improvisation session would be done to understand what could be done to make things better and appreciate things well done. Everyone will give anonymous feedback. In case of huge number of feedbacks, 1 or 2 feedbacks would be taken as improvement items for next sprint.

# Deliverables (HCL)

At end of each sprint there would a defined set of deliverables. Following is list of deliverables:

## Product increment

All changes done by development team. These changes include code changes which would be in form of Git/repository commits. No code shall be kept at developer machine, in shared location or on emails.

## Unit test

Unit test for all code changes are must. We will use testng for unit testing. Developer would commit these in same branch as source code for user story and they would be part of code review as well. 80% coverage is expected.

## Code coverage report

Developer would be run unit tests and show code coverage to code reviewer, if asked. This will also happen in build pipeline in Jenkins.

## Test Automation

All the new flows introduced, or changes would be automated so that in next sprint a quick regression can be done to ensure flow is not breaking with other changes. Automation test would also be in git.

## Demo

At the end of sprint team will give a demo of all completed user stories to product team. This will be a planned session with whole development team, product team and other stockholders. Feedbacks if part of acceptance criteria would be added back to user story by story owner(developer) else new story in backlog would be created for enhancements and scope changes.