

K2 VENTURES INSTANT PAY

Solution Architecture and HLD Document

Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change Reference | Author |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Purpose: 7](#_Toc55125603)

[2. Background 7](#_Toc55125604)

[3. Scope 7](#_Toc55125605)

[3.1. In Scope 7](#_Toc55125606)

[4. Functional View 8](#_Toc55125607)

[4.1. Functional View Details 9](#_Toc55125611)

[4.1.1. General Component – 9](#_Toc55125612)

[4.1.1.1. Security 9](#_Toc55125613)

[4.1.1.2. Exception Handling 9](#_Toc55125614)

[4.1.1.3. Back Job 9](#_Toc55125615)

[4.1.1.4. Payment Provider 9](#_Toc55125616)

[4.1.1.5. Notifications 9](#_Toc55125617)

[4.1.1.6. Audit Trail 10](#_Toc55125618)

[4.1.2. Admin 10](#_Toc55125619)

[4.1.2.1. Dashboard 10](#_Toc55125620)

[4.1.2.2. Roles and Permission 10](#_Toc55125621)

[4.1.2.3. Merchant Transaction 10](#_Toc55125622)

[4.1.2.4. Customer Payment Transaction 10](#_Toc55125623)

[4.1.2.5. Subscription Management 10](#_Toc55125624)

[4.1.2.6. Split Payment Management 10](#_Toc55125625)

[4.1.2.7. Manage Customer / Merchant Management 11](#_Toc55125626)

[4.1.2.8. Payment Method Management 11](#_Toc55125627)

[4.1.2.9. Alert Management 11](#_Toc55125628)

[4.1.2.10. Instant Pay User Management 11](#_Toc55125629)

[4.1.2.11. Sales Report 11](#_Toc55125630)

[4.1.3. End Customer 11](#_Toc55125631)

[4.1.3.1. Registration – 11](#_Toc55125632)

[4.1.3.2. Login 12](#_Toc55125633)

[4.1.3.3. Dashboard 12](#_Toc55125634)

[4.1.3.4. Transaction Details 12](#_Toc55125635)

[4.1.3.5. Card Details 12](#_Toc55125636)

[4.1.3.6. Payment Details 12](#_Toc55125637)

[4.1.3.7. Refund Payment 12](#_Toc55125638)

[4.1.3.8. My Profile 12](#_Toc55125639)

[4.1.4. Merchant 12](#_Toc55125640)

[4.1.4.1. Registration 13](#_Toc55125641)

[4.1.4.2. Login 13](#_Toc55125642)

[4.1.4.3. Dashboard 13](#_Toc55125643)

[4.1.4.4. Transaction Detail 13](#_Toc55125644)

[4.1.4.5. Payment Detail 13](#_Toc55125645)

[4.1.4.6. Subscription Plan 13](#_Toc55125646)

[4.1.4.7. Settings (User & Roles) 13](#_Toc55125647)

[4.1.4.8. My Profile 14](#_Toc55125648)

[4.1.5. Marketing 14](#_Toc55125649)

[4.1.5.1. Login 14](#_Toc55125650)

[4.1.5.2. Sign Up 14](#_Toc55125651)

[4.1.5.3. About Us 14](#_Toc55125652)

[4.1.5.4. Contact Us 14](#_Toc55125653)

[4.1.5.5. Subscription Plan Information 14](#_Toc55125654)

[4.1.5.6. Help 15](#_Toc55125655)

[4.1.5.7. API Docs 15](#_Toc55125656)

[5. Implementation View 15](#_Toc55125657)

[5.1. High level Component Details: 16](#_Toc55125658)

[5.1.1. React JS 16](#_Toc55125659)

[5.1.2. Web API 16](#_Toc55125660)

[5.1.3. SQL Database 16](#_Toc55125661)

[5.1.4. Redis Cache 16](#_Toc55125662)

[5.1.5. Back Job 16](#_Toc55125663)

[5.1.6. API Filter 16](#_Toc55125664)

[5.1.7. DB Context 16](#_Toc55125665)

[5.1.8. SMS/Email Service – 17](#_Toc55125666)

[6. Logical View 17](#_Toc55125667)

[6.1. Technology Stack: 18](#_Toc55125668)

[6.2. Logical View Detailed Explanation: 18](#_Toc55125669)

[6.3. Logical Flow: 18](#_Toc55125670)

[6.3.1. One click solution API 18](#_Toc55125671)

[6.3.2. Split checkout solution API 18](#_Toc55125672)

[6.3.3. Admin Interface 18](#_Toc55125673)

[6.3.4. Merchant Interface 18](#_Toc55125674)

[6.3.5. Customer Interface 19](#_Toc55125675)

[6.3.6. Merchant Marketing Interface 19](#_Toc55125676)

[6.3.7. Back Job 19](#_Toc55125677)

[6.3.8. DB Layer 19](#_Toc55125678)

[6.3.9. Redis Cache 19](#_Toc55125679)

[6.3.10. Payment Provider 19](#_Toc55125680)

[6.3.11. N-Logger 19](#_Toc55125681)

[6.3.12. SMS Gateway 19](#_Toc55125682)

[6.3.13. Email Service – 19](#_Toc55125683)

[6.3.14. Encrypted token 19](#_Toc55125684)

[6.3.15. Service layer 19](#_Toc55125685)

[7. Security 20](#_Toc55125686)

[7.1. Merchant API Security Diagram 20](#_Toc55125687)

[7.2. Web App Security Diagram and Cookies store Diagram 21](#_Toc55125688)

[7.3. Overview of process flow of request and response 22](#_Toc55125689)

[7.3.1. Authentication Token key 22](#_Toc55125690)

[7.3.2. Content Type 22](#_Toc55125691)

[7.3.3. URL 22](#_Toc55125692)

[7.3.4. Request type 23](#_Toc55125693)

[7.3.5. Response 23](#_Toc55125694)

[8. Process Flow 24](#_Toc55125695)

[8.1. Instant Pay Checkout Process Merchant Website WORK FLOW 24](#_Toc55125696)

[8.2. Instant Pay Checkout Process Merchant Website Process Flow – One Click Checkout 25](#_Toc55125697)

[8.3. Instant Pay Checkout Process Merchant Website Process Flow – Split Payment Checkout 25](#_Toc55125698)

[8.4. Merchant Marketing Portal Work Flow 27](#_Toc55125699)

[8.5. Merchant Registration Process Marketing Process Flow 28](#_Toc55125700)

[8.6. End Customer Registration from Marketing Portal 28](#_Toc55125701)

[8.7. End Customer Panel Workflow: 29](#_Toc55125702)

[8.8. Merchant Panel Workflow: 31](#_Toc55125703)

[8.9. End Customer Panel Essential Flow 32](#_Toc55125704)

[8.9.1. Payment Method Setup 32](#_Toc55125705)

[8.10. Merchant User Panel Essential Flow 32](#_Toc55125706)

[8.10.1. Merchant User Panel Subscription Process flow 32](#_Toc55125707)

[9. Sequence Diagram 33](#_Toc55125708)

[9.1. One Click Payment Process 33](#_Toc55125709)

[9.2. Split Payment Process 35](#_Toc55125710)

[9.3. Merchant Registration Process 36](#_Toc55125711)

[9.4. Customer Schedule Payment Process 37](#_Toc55125712)

[9.5. Customer Refund Payment Process 38](#_Toc55125713)

[9.6. Merchant Refund Process 39](#_Toc55125714)

[10. Merchant API DOCUMENTATION 40](#_Toc55125715)

# Purpose:

The Solution Architecture and HLD Document provides a comprehensive architectural overview of the solution or application. It presents a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system. This document is intended to provide the Architecture and High-level design view of the K2 Venture Instant Pay Payment Processing Application and its different Feature Modules.

# Background

K2 Ventures is looking to design and develop an application which will facilitate the feature of One Click and Split payment option for its registered users utilizing the platform to process online payments for their purchases from the Merchant websites who are also the subscribed members of the K2 Venture Instant Pay platform. The is aimed towards development of Web interfaces for the 3 categories of users the K2 Venture Users, Merchants who are the Subscribers of the Instant Pay Service for their online portals and the End Customers or Purchaser.

# Scope

The Scope of work include the design and development of a Payment Processing application which is used by Online Customer in this document mentioned as “End Customer hereafter” who will be visiting various Instant Pay Partner portal “which are mentioned as Merchants in this document hereafter “to process the purchased goods or services using the convenience of the Integrated One Click and Split Payment option of Instant Pay Payment processing system which is intended to be developed with below interfaces;

**System Actors / Primary Users:**

* Instant Pay Admin Users
* Merchant Users
* End Customer Users

## In Scope

**Instant Pay Admin Panel**: This will be an Admin Access for the Instant Pay User group to maintain the services and users of the Instant Pay service subscribers which are the Merchants and also the registered End Customers who will utilize the payment processing services of the Instant Pay.

**End Customer Web Panel**: The End users will be given the access to an Interface where the End Customers to view their payment history using Instant Pay, Manage their payment options and profile.

**Merchant Web Panel**: Here the Merchant or the Subscribers of the Instant Pay Payment Service will be able to access to manage their internal users, manage and apply for subscription and view sales transaction reports, view dashboards and notifications

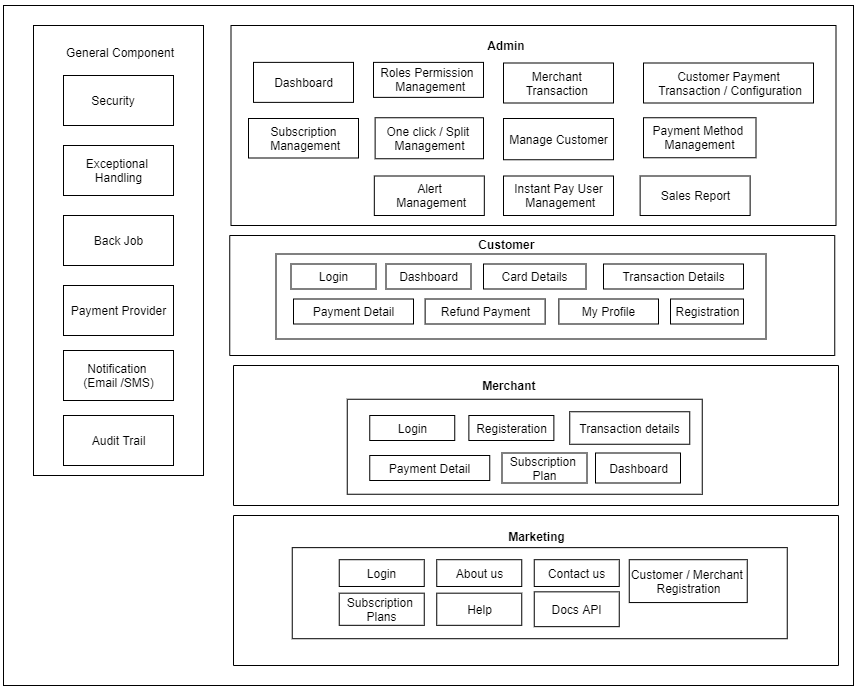
**Marketing Portal for End Customer and Merchants:** This portal will represent the Marketing Activities and service offerings of K2 Ventures Instant Pay Payment Services and pages related to the Merchant API Information and various pages including clientele, About US, Term and Conditions and Policies for the user’s groups. The portal will also contain the Sign and Signup option for the End Users and Merchants along with the Subscription Pricing information for the Instant Pay services. The content will be maintained to be maintained by Instant Pay Admin from the Admin Panel.

**Merchant Check Out API Developments:** Merchants will be provided with the Payment APIs which will broadly for One Click Payment, Split Payment and Both options combined which once subscribed by Merchant will be available to the Merchants along with the code to implement the Instant Pay Payment Service on to their respective Merchant Website check-out pages.

**Payment Gateway Integration:** The Instant Pay Payment processing application will require to be integrated with payment gateway service provider in order to process End Customer transactions placed through the Merchant Websites along with processing payment requests from Merchant Panel for Subscription of Instant Pay services by the Merchant. Payment gateway services will also be required to manage and store the Payment Information details of the End customersto enable End Customers to avail the facility to quick check out.

# Functional View

Functional View





## Functional View Details

## General Component –

## Security

This Block will handle all the Security related aspects of the system. Like, Authentication, Authorization, CSRF (Cross Site Request Forgery), Security for the external request will be handled here, every request will be validated and authenticated and then only grant for the Further action. Any Unwanted request will be refused.

## Exception Handling

K2 System runtime exceptions will be captured and stored in file system also this logs will be in 10 MB chunks so that it should not be heavy while reviewing. Notification mechanism will be applied over Exception handling so that Tech team can take appropriate action on it. This notification will be in pre-set time period.

## Back Job

4 Back jobs will be created for the K2 system,

1. Emailer / SMS – Notification
2. Split Payment – Dues clearance
3. Refund Processor
4. K2 System merchant’s subscription plan tracker. (this will check merchant has valid)

All the back jobs will take respected chunks of data from database and keep it application level and start processing on it.

Back job will help K2 system to make back end process effective.

## Payment Provider

K2 System Payment provide block will contain payment gateway. Payment gateway will be integrated in such a way that if any new payment provide K2 admin team needs to include then it will not affect whole system, just that components will be replaced and K2 system can start using new payment provider.

## Notifications

Notification plays important role in any system, which helps to understand any system functionality. In K2 System two types of notification will be involved, SMS and email.

SMS notification will be for OTP and Payment related transaction

Email Notification for – New customer / Merchant registration, New payment transaction, Split payment transaction details from back job and refund request initiator email.

## Audit Trail

K2 system is based on money transaction, so every activity has to logged, this activity will be parallel. Also this logger will be configurable.

## Admin

## Dashboard

The Instant Pay Admin panel will have Dashboard for user with representation of various data cards and Grid information that provides overview of the overall statistics of the Instant Pay Payment Service to the Instant Pay Admin users. Currently data intended to be displayed are Total End Customer Count, Total Merchant Count, Transaction Count, Transaction Value Count.

## Roles and Permission

This component will mainly manage the roles and user permission for the internal Instant Pay Admin users and the access and page rights for the Admin Panel. This will have different user roles with permission to access the pages in the panel and also the Admin user can create new roles and permissions to the list,

## Merchant Transaction

All the merchant related transaction will be represented in this component which is intended to list the total transactions for the payment processed by all subscribed Merchants. The data will provide the details of the Instant Pay transaction those have been processed from a single Merchant

## Customer Payment Transaction

All End Customer who have been registered with the Instant Pay Payment services who are using the service will be listed here which will provide information on what are the recent purchases made by the end user at various Merchant Websites

## Subscription Management

This component will manage the Instant Pay Payment Subscription type where the Admin Users will be able to Add Edit the Delete Merchant based Subscriptions which will reflect changes in the Subscriptions that are available to be applied by the Merchant

## Split Payment Management

The Spit Payment management section will allow the Instant Pay Admin user to configure the Duration against the Billing Value of the order to determine to the End Customer the different options of selection for the duration and instalment options while End Customers are processing payment using Instant Pay services through a Merchant portal.

## Manage Customer / Merchant Management

This section in the Admin Panel allows Instant Pay Admin Users to view the list of Merchants and their Subscription details along with the information on the number of transactions and value which are been associated with the transactions. Instant Pay Admin will also be able to Approve newly registered Merchant Accounts from this component.

## Payment Method Management

The Admin user will be able to keep track from this listing about the details of the transaction and the status. The user will get detailed information with the transaction id, details of the End Customer, Merchant, Payment Processer and Method details with their respective status.

## Alert Management

The Application Alert Management related to the Notification and Email templates which is triggered from different user actions thought out the application panels and marketing portal will be controlled and edited from this section

## Instant Pay User Management

This section will be able to Add new Instant Pay Users and assign the defined roles to the users and also delete the users which are internal Admin users

## Sales Report

Admin users will be able to generate different sales reports using advance search filters and the users will be allowing to export the searched data via Export to CSV option

## End Customer

## Registration –

End Customer Registration process will be managed when end customer registered from the Sign Up section of the Marketing Portal and logs to the End Customer panel after verification of the users Phone and Email. The registered profile is maintained in End Customer panel

## Login

End User who has already registered will be managed in under this component for the user authentication and redirection to the End Customer accounts panel for the customer to view the account information

## Dashboard

End customer will be able to view cards with Transaction Details with Total Success and Failure of the Transactions and Total Amount Spend with the List of recent transaction which will be managed in this component

## Transaction Details

End Customer all transaction listing will be managed here where the End Customer will be able to view total transactions executed by the End Customer using Instant Pay Payment Services and using which Merchant Portal

## Card Details

End Customer will be able to Add new Card information which the details of the Card Number, Name, Expiry and other details and save the card information for future Quick Payment options using Instant Pay Merchant portals

## Payment Details

End Customer will be able to view the Saved Card information here and will have the ability to delete or set the certain cards as default mode of payments for the transactions processed using Instant Pay

## Refund Payment

Cancellation and refunds initiated using the Merchant portals which are transacted using Instant Pay Payment service will be managed with the status here for the End Customers

## My Profile

User will be able to edit profile details and change password from this section also will have the ability to change an update profile picture.

## Merchant

## Registration

The Merchant Users will register from the Marketing Portal similar to the End Customer registration, newly registered Merchant Account will be Approved or Declined by the Instant Pay Admin Users from the Admin Panel

## Login

Login process for the Merchant Users will be managed here which will authenticate the Merchant users and redirect them to the respective Merchant Panel

## Dashboard

The Merchant users will be able to view the various transactional data on the Merchant Dashboard as a part of the landing page after Login process. Data related to daily sales, monthly sales, No. of End Customers, Total Orders are to be handled in the Dashboard component

## Transaction Detail

The transaction details for the Merchant to view all the End Customer transaction processed from the Merchant Portal using the Instant Pay Payment Processing option during the Check-out process

## Payment Detail

This will maintain the subscription payments list which will be associated with the Subscription plan to maintain a log for the payments made by Merchant users for the subscriptions availed towards the Instant Pay Payment APIs.

## Subscription Plan

This section will provide Merchant user with the available plans and the ability to choose from the plan after the Merchant user is Registered and Logged in. Upon selection of the subscription plan the Merchant user will be able to proceed towards payment using Instant Pay option or via entering Card Details and also has the ability to change and cancel current subscription plan

## Settings (User & Roles)

Merchant users will be able to access the Settings section after login, Merchant can add new users with appropriate roles and rights. also master data content can also be modified and added in the setting section.

## My Profile

The Merchant user will be able to maintain profile and make changes to the Profile Pictures and change password from this proceed

## Marketing

## Login

This will allow the End Customer and Merchants to Login in to the respective End Customer and Merchant Panel using User ID and Password

## Sign Up

From Sign Up component the Merchants and End Users will have option to sign up for the Instant Pay Payment services as End Customer and Merchant Partner which will manage the Sign Up process with the required input fields

## About Us

This will maintain content related to the Instant Pay and K2 Ventures About Us information

## Contact Us

The section will be including the registered contact address and customer support contact

## Subscription Plan Information

The will be including the different types of Subscription Plan for the Merchant and features and pricing of the plans that the Merchants can choose from. This section will have details plans and literature on the feature

## Help

Can maintain FAQs and basis sections for customer support and maintenance

## API Docs

This will have detailed maintained documents related to Merchant APIs for the integration of Instant Pay Payment processing service. This will have the technical Request Response documentation and APIs code details that can be informative for the Merchants to implement the code in their portals for the Instant Pay Payment Checkout options

# Implementation View

Implementation View



## High level Component Details:

K2 system will have front end, backend and DB components with back end job service component This each Components will be responsible for transaction of data into k2 ventures system K2 ventures system will have six physical components where the data will travel through these physical components.

## React JS

React JS Component will be standalone component; this component will receive JSON to show on UI. React Component is purely JavaScript application where request will be made as per request response description.

## Web API

Web API is also standalone component, where this will be hosted on server side and URL will exposed outside world but it can only to access via encrypted Token only, it will be abstracted module component where this is the major component for data manipulation and business Logics. This component will be consisting of API Filter, Controller, Service layer, DB Context, Notification and SMS and Email service.

## SQL Database

SQL Database will be the third component. SQL Component stores all Customer Merchant and Admin Profile and Configurable data and some of the business logics will be stored inside SQL.

## Redis Cache

Cache Server is very helpful for faster data response from server level. Most of the Master level configuration will be stored inside the Cache server so request from client component will have quick response instead of traveling full request till Database level.

## Back Job

Back jobs are services which helps to full fill activities like Customer split payment balance amount and refund activity. Also merchant subscription plan checker.

## API Filter

API filter is the very first component to hit when any new request comes in K2 API Layer, this Component takes care of the security of API layer. API filter will contain all encryption decryption logic.

Filter will always check for http request header so that if expected request headers are not present then that request will not be allowed.

## DB Context

DB Context Layer in the internal component of API layer, this layer communicates to Database directly.

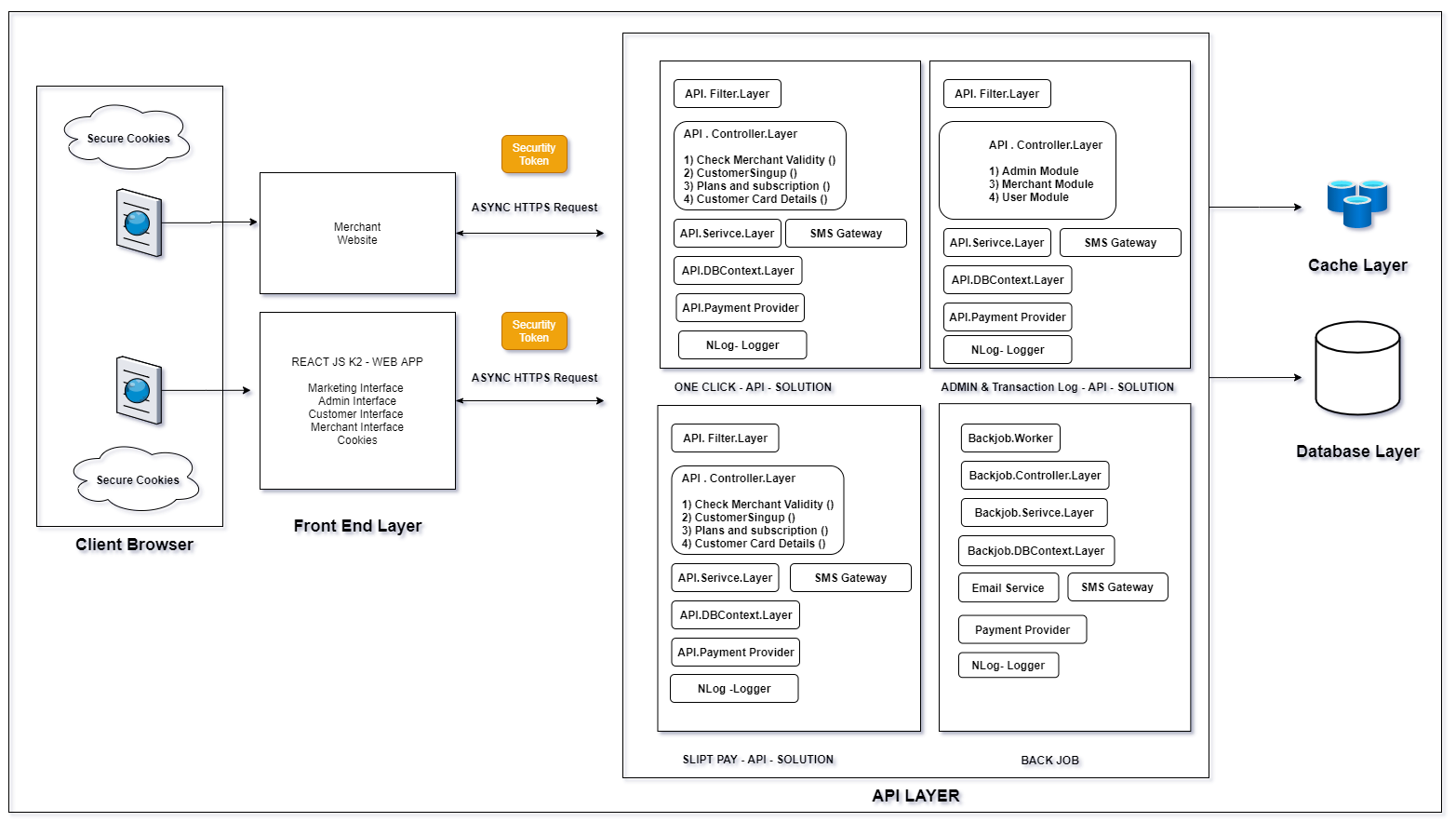
ADO.net will be used and in DB Context Layer will be connected architecture, in all the fetching mechanism data reader approach will be used.

## SMS/Email Service –

For every payment transaction success and Failure or forgot password SMS or Email notification will be triggered.

SMS and Email service component will be configurable and provider will have their own API callers so that this activity should not hamper other functionality of the system.

# Logical View



Logical View

Logical view is the to get clear idea of the architecture and its components, API Layer designed in such a way for the high availability to the Customers. Also, applications maintenance and trouble shoot will be easy.

## Technology Stack:

|  |  |  |
| --- | --- | --- |
| # | Technology | Usage |
| 1 | React JS | For Presentation layer (UI) |
| 2 | Web API | C# .net core version 3.1 web api For transmission of data, This will connect to database |
| 4 | MSSQL DB | Database storage. To store information of system. |
| 5 | CDN | Document and Image storage. |
| 6 | Redis Cache | To Cache information |

## Logical View Detailed Explanation:

Application is divided into following section

* Frontend Layer (Customer Interface + Merchant Interface+ Admin Interface + Merchant Marketing)
* API Layer – Request Filters, Controllers, services, database and notification (Email /SMS) , Payment provider.
* Back Jobs- Refund and Split payment dues will be handled in this back jobs.

## Logical Flow:

When any request comes from browser, Request will hit filter first, in filter request will be authenticate, if request is authenticated then only request will be pass to controller else request will not able to enter in controller layer.

Once request enters in controller layer required resources will be accessed like, Services, DB context SMS Gateway or Payment services.

## One click solution API

This Solution will be handling transaction which is related to one click instant payment only. This solution will be hosted separately. From Merchant website request will come here in this solution directly.

## Split checkout solution API

This solution will handle app request which are related to split payment, this solution will be hosted separately.

## Admin Interface

This will be React JS admin portal, login for this will be same as Merchant login – Login url will be private.

## Merchant Interface

This Merchant interface will be in react js component only and login for the merchant will be private URL. Login URL and credentials will be shared on his email id.

## Customer Interface

End Customer will access this interface and login with dashboard and transaction will be seen here and Customer see his/her details in this section.

## Merchant Marketing Interface

This interface will be publicly expose, this will have Signup functionality for both merchant and end customers.

## Back Job

For Refund+ Scheduled Payment processing + and SMS and email notification will be here with payment providers.

## DB Layer

Every transaction to be stored in DB with success/failure result

## Redis Cache

All master related information will be saved in Radis cache. To reduce database calls every time. Redis cache will be helpful.

## Payment Provider

Customer and Merchant payment done through any third party provide like (CAS or Spreedly), This will be handled here in this section

## N-Logger

Every failure / success / info will be logged in particular file using logger, This files will be in chunks

## SMS Gateway

Customer or merchant registration SMS to be triggered for OTP validation also for transaction happened.

## Email Service –

For every success / failure payment / Forgot password/ payment instalment mail service provided

## Encrypted token

This will be generated in to API layer. Across all application communication will contain Token Key in header to authenticate request and which will contain Merchant information or customer information and as well encrypted Customer information which will be converted into bearer Token Key.

## Service layer

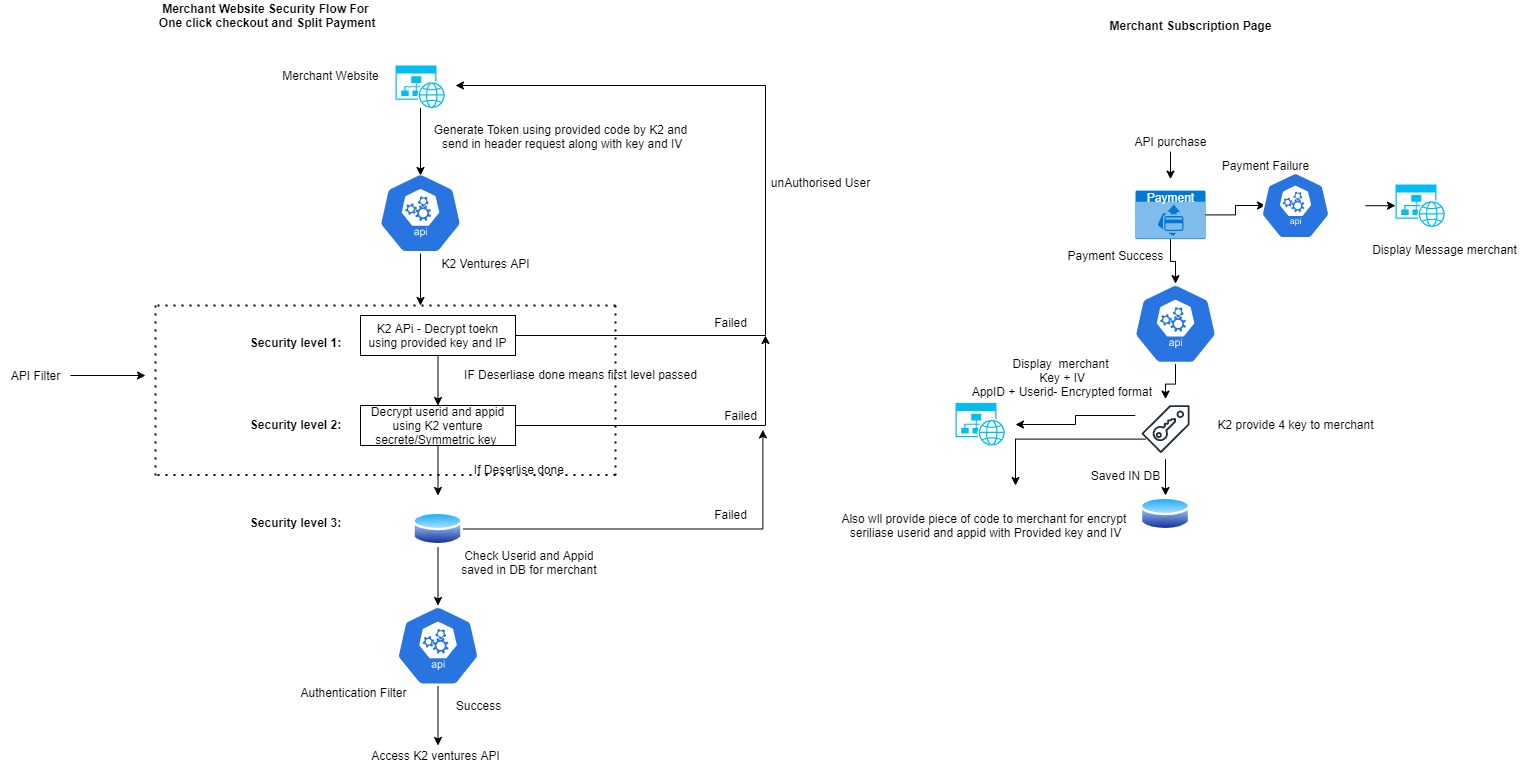
This will be generated in to API Gateway layer. Across all application communication will contain OAuth Token Key to authenticate request and which will contain tenant ID and as well encrypted Customer information which will be converted into bearer Token Key.

# Security

Container will contain Checker for SSL authentication schemes because Basic authentication and forms authentication send unencrypted credentials. To be secure, these authentication schemes ***must*** use SSL, addition, SSL client certificates can be used to authenticate clients.

Which will be handled in the Security Container.

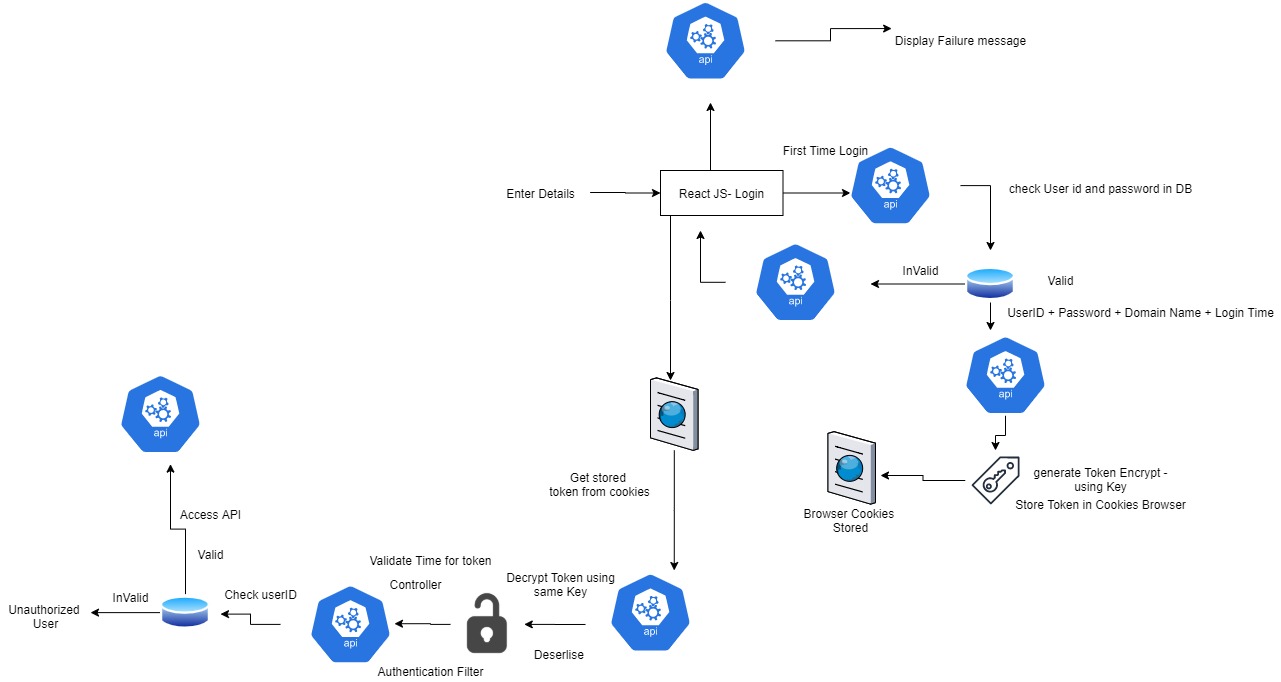
## Merchant API Security Diagram



Merchant API Security Diagram

* One Customer purchase product from merchant he/she will select k2 venture one click or split payment option as soon as they select will send encrypted token to k2 ventures domain
* Level 1: K2 ventures decrypt those token using provided key and iv by merchant
* Level 2: If level1 success, and whatever level1 decrypted result will get – USERID and APPID these 2 encrypted parameter k2 ventures will again decrypt using his secrete key which k2 ventures only have
* Level3 If those decryptions done successfully will check those USERID and APPID in our DB which is saved against that merchant
* If level 3 also crossed will authenticate those Customer using authentication filter and allow to access k2 ventures next features or API.

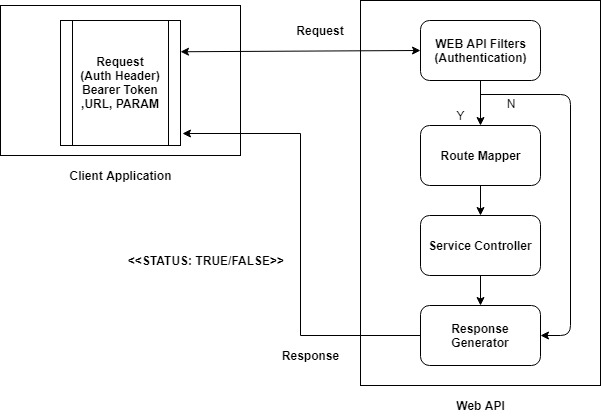
## Web App Security Diagram and Cookies store Diagram



Web App Security Diagram And Cookies Store Diagram

* Step 1: Once customer login to application user id and password will be authenticate in DB
* Step 2: If user validation success in DB generate token using userid +password+ Domain Name +Login Time and stored in cookies- if validation failed display message to website (Invalid user)
* Step 3: If customer second time logged in his stored token from browser will use for login mechanism
* Step 4: If token decrypt using same key and decrypt those value and time matches within login time (ex- if 24 hrs limit given then login time and current time will match and process for Further)
* Step 5: After deserialization success Credentials check in DB and if those customers validate with DB he will allow for access K2 APIS

## Overview of process flow of request and response



Request Response Lifecycle of Instant System

## Authentication Token key

Web API service which can be accessed over HTTP by client. So providing security to Web api is very important. which can be easily done with the process called Token based authentication.

Token-based authentication is a process where the user sends his credential to the server, server will validate the user details and generate a token which is sent as response to the users, and user store the token in client side, so client do further HTTP call using this token which can be added to the header and server validates the token and send a response.

## Content Type

Content Type defines the request content format; Request may be in different format according to the needs.

For K2 System API request content type will be application /JSON which means request formation will be in JSON always with default encoding is UTF-8.

## URL

Web API contains multiple attributes to call end point. End point is where data can be accessed, URL is the definer for end point. In URL web API understands for which method /action request has come.

Example : https://{domain}/{controler}/{action}

Here:/{controller}/{action} are end points (URL)

## Request type

The REST API is used to expose the Web Services of our ticketing application by using simple HTTP requests, which includes the following:

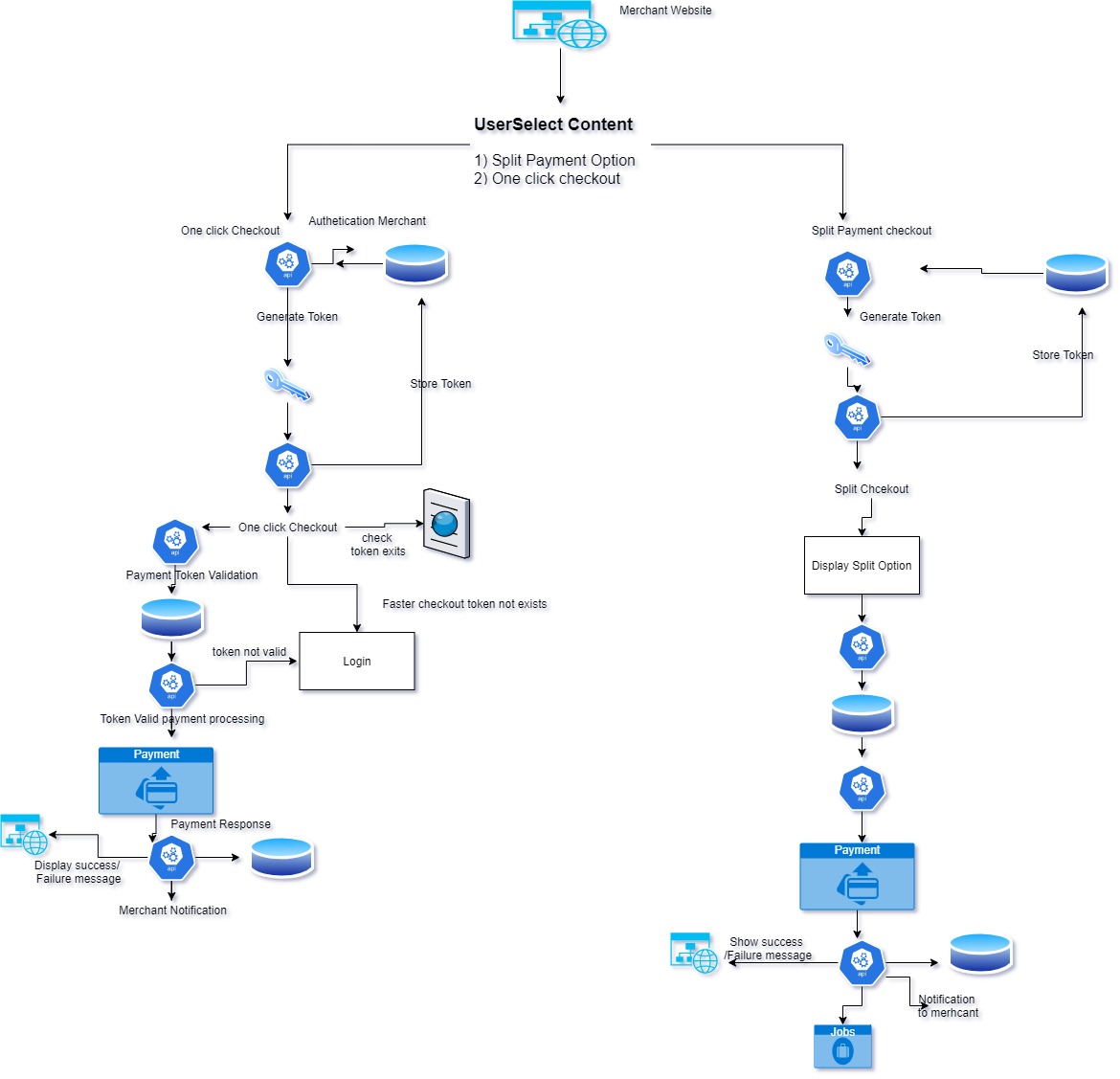
* POST method for creating a new Object on server
* GET method for getting single/list of Objects from server
* PUT method to update an Object on server
* DELETE method for deleting an Object from server

## Response

Once Request Received at Server Side .net Core Web API then the Authentication token will be valid first using filters. Ones the validation of Authentication token is passed then the request will be routed to Route Table to that the appropriate controller service will be responsible for the request. Once The data is generated for response it will be filled in the request and the response will be returned in same request. Response will be in JSON format, where nodes will be created according to data. Once the client application receives the response the lifecycle of the request response ends.

# Process Flow

## Instant Pay Checkout Process Merchant Website WORK FLOW



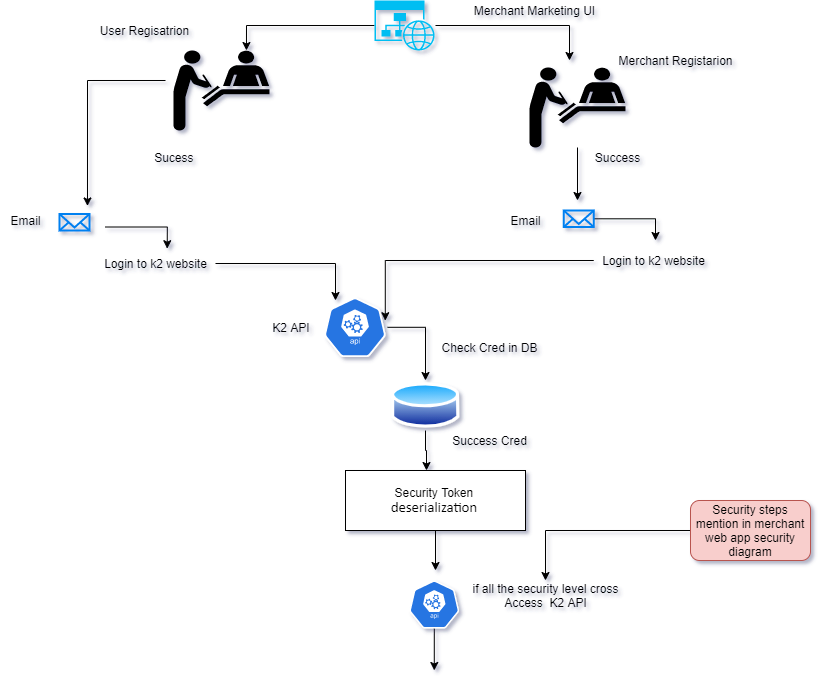
## Instant Pay Checkout Process Merchant Website Process Flow – One Click Checkout

* Step 1: End Customer lands at the Merchant Checkout Page
* Step 2: The End Customer chooses to checkout using the Instant Pay One Click Checkout
* Step 3: Upon clicking the One Click checkout the system will validate the Merchant Subscription validity
* Step 4: End Customer login credentials is validated
  + In the cookies if the End Customer have chosen to stay signed in to Instant Pay Quick Checkout service
  + If not signed in End Customer is shown Sign in screen with in the Merchant Check out Page to Login in by using User ID and Password
  + If the End Customer is not a registered user of Instant Pay services, they can click on Sign Up option on the Login Screen to get redirected to the Instant Pay Marketing Portal for Registration Process
  + After the registration process the End customer enters the Payment information on End Customer Panel and saves information to again proceed towards the Merchant Checkout Page to get the order confirmation
* Step 5: The system validates credit card validation and available limit for the card
* Step 6: The End Customer Successfully completes the Order using Instant Pay One Click payment option
* Step 7: Step 9: System will trigger notification via email/sms and notification once the order is confirmed

## Instant Pay Checkout Process Merchant Website Process Flow – Split Payment Checkout

* Step 1: End Customer lands at the Merchant Checkout Page
* Step 2: The End Customer chooses to checkout using the Instant Pay Split Payment Checkout
* Step 3: Upon clicking the Split Payment checkout the system will validate the Merchant Subscription validity
* Step 4: End Customer login credentials is validated
  + In the cookies if the End Customer have chosen to stay signed in to Instant Pay Quick Checkout service
  + If not signed in End Customer is shown Sign in screen with in the Merchant Check out Page to Login in by using User ID and Password
  + If the End Customer is not a registered user of Instant Pay services, they can click on Sign Up option on the Login Screen to get redirected to the Instant Pay Marketing Portal for Registration Process
  + After the registration process the End customer enters the Payment information on End Customer Panel and saves information to again proceed towards the Merchant Checkout Page and reselect the Split Payment option and proceed payment
* Step 5: The system validates credit card validation and available limit for the card and block the order amount before proceeding towards payment processing
* Step 6: The End Customer selects the Split Payment option and views the instalment and interval options which is displayed on the checkout screen of the Merchant Checkout page
* Step 6: End User selects the instalment amount and interval to proceed towards the payment.
* Step 7: The End User complete the payment and proceeds to Order Confirmation
* Step 8: The system will be insert the schedule instalment details on the backend DB to process future scheduled instalment using Backjob
* Step 9: System will trigger notification via Email/SMS and notification once the order is confirmed and for the scheduled instalment triggers

## Merchant Marketing Portal Work Flow



## Merchant Registration Process Marketing Process Flow

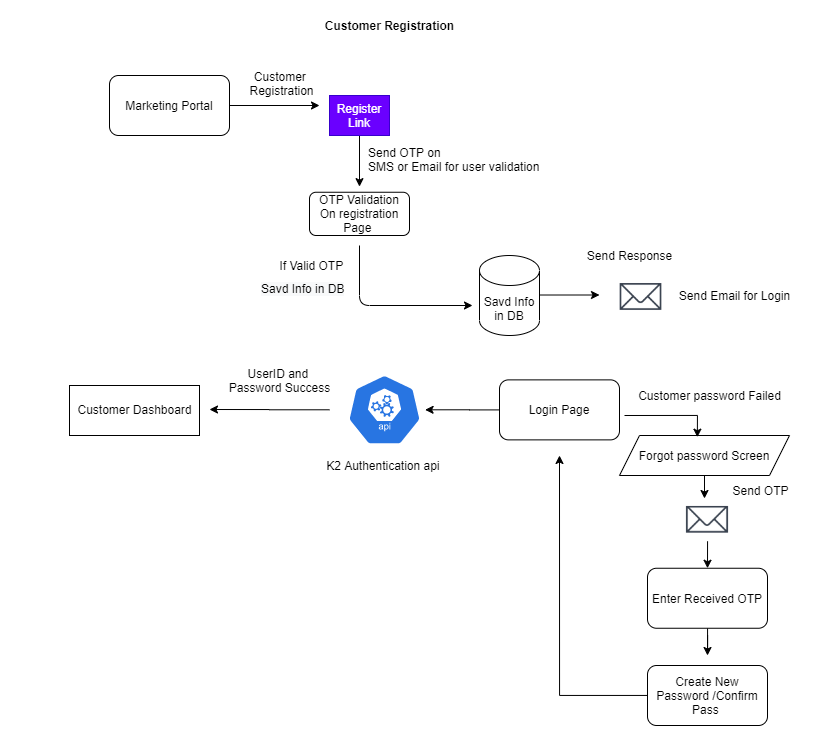
* Step 1: Merchant visit the Instant Pay Marketing Portal to register themselves with Phone No., Email Address, to send a valid OTP to via SMS or Email
* Step 2: Once OTP validated the End Customer will be prompt to Create Password and Confirm Password to complete the registration process
* Step 3: The Merchant User will now be able to visit the Instant Pay marketing to sign in using the Login credentials which will be validated from the End User Database
* Step 4: Merchant User will have an option of Forget password which will ask for Phone No / Email Address to send the OTP which after entering will allow user to Reset New Password and Confirm Password
* Step 5: The Merchant User will now be able to Login to the Merchant Panel to view dashboard and apply for a Subscription plan and complete the payment towards the plan

## End Customer Registration from Marketing Portal

* Step 1: End Customer visit the Instant Pay Marketing Portal to register themselves with Phone No., Email Address, to send a valid OTP to via SMS or Email
* Step 2: Once OTP validated the End Customer will be prompt to Create Password and Confirm Password to complete the registration process
* Step 3: The End Customer will now be able to visit the Instant Pay marketing to sign in using the Login credentials which will be validated from the End User Database
* Step 4: End User will have an option of Forget password which will ask for Phone No. / Email Address to send the OTP which after entering will allow user to Reset New Password and Confirm Password
* Step 5: The End Customer will be during log in will be prompted to be Logged In for Faster future checkout

## End Customer Panel Workflow:

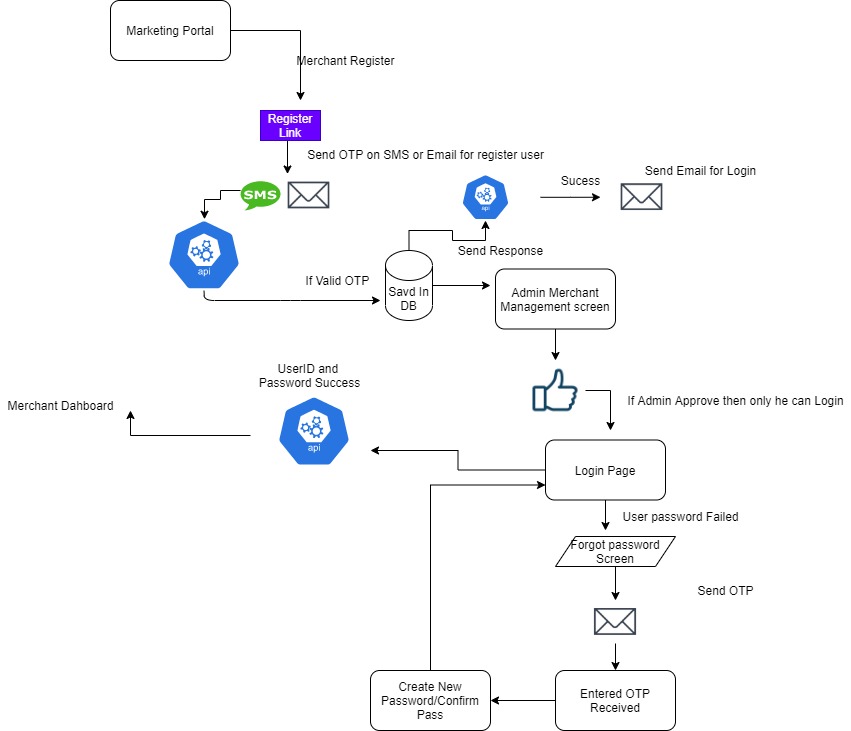
End Customer panel workflow





End Customer User Panel

## Merchant Panel Workflow:



Merchant User Panel

## End Customer Panel Essential Flow

## Payment Method Setup

Here the user will be required to fill in the profile and payment information with the card details and billing information

* Step 1: First Time Registered Users will be prompt on the notification to complete the payment information details
* Step 2: User navigate the payment page clicks the tab
* Step 3: User will be asked to fill in information in the Profile Tab
* Step 4: User will Enter the Credit Card Details in the Payment Tab
* User can choose to Add Multiple Cards
* User will have to choose at least One Card as a default for using it with Instant Pay transactions
* Step 5: User will add Default Billing Address in the Address Tab

## Merchant User Panel Essential Flow

## Merchant User Panel Subscription Process flow

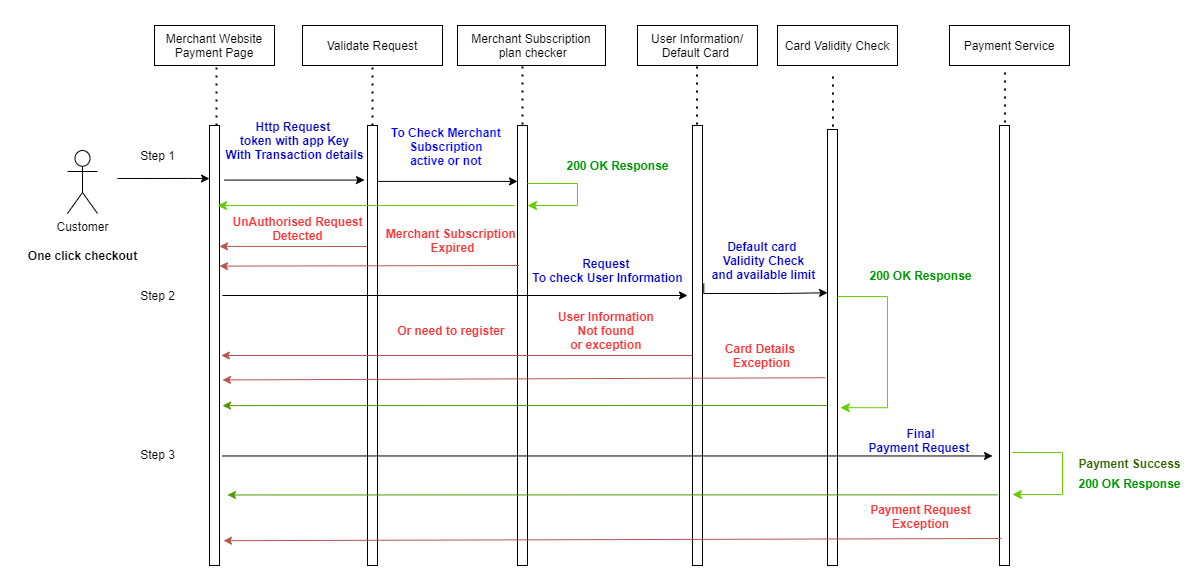
Merchant will be able to view and manage the subscription from this page and also upgrade and cancel the subscription

* Step 1: Merchant will be able to view the Subscription Tab in the Merchant Interface after Login
* Step 2: Merchant will click on the Subscription Tab to view;
* Subscription Name: (One Click / Split Payment / One Click and Split Payment Both)
* Subscription Information: (This field will be providing a brief information on Subscription type and facilities available towards the subscription
* Subscription Validly: Merchant will view the validity period against each subscription
* Step 3: Merchant user will choose the ideal plan for the subscription by choosing the Select Plan CTA which will redirect the Merchant to the next page where payment will be processed
* Step 4: Merchant will be will able to complete the payment by entering the Card Information and Billing details and click on Pay CTA to complete the payment.
* Step 5: Merchant will receive the confirmation email via email on the registered email and a notification on the Merchant Interface for the subscription and also the confirmation email will content information on the APIs and its keys.
* Step 6: The Merchant user will have access to the view form of the subscription where they can view the Subscription Name, Type, Validity and a CTA for Change and Cancel as explained below;
* Merchant can Change to a different Subscription plan by clicking on the Change CTA which will redirect to the Choose Subscription Plan as mentioned in Step 2 above and run by the same process.
* Merchant can choose to Cancel the Subscription at given time by clicking on Cancel subscription and this will ask for a confirmation using a Pop-Out with a Confirm or Cancel CTA for the user to choose from
* For both change in plan will trigger email to Merchant user similar to that of Step 5 and for the cancellation the Merchant will receive the confirmation email and simultaneously the system will disable the APIs and the Keys on the merchant websites to avoid future checkouts for Customers using Instant Pay for that Merchant

# Sequence Diagram

## One Click Payment Process

**One click Payment Process**



One Click Payment Process divided into 3 steps

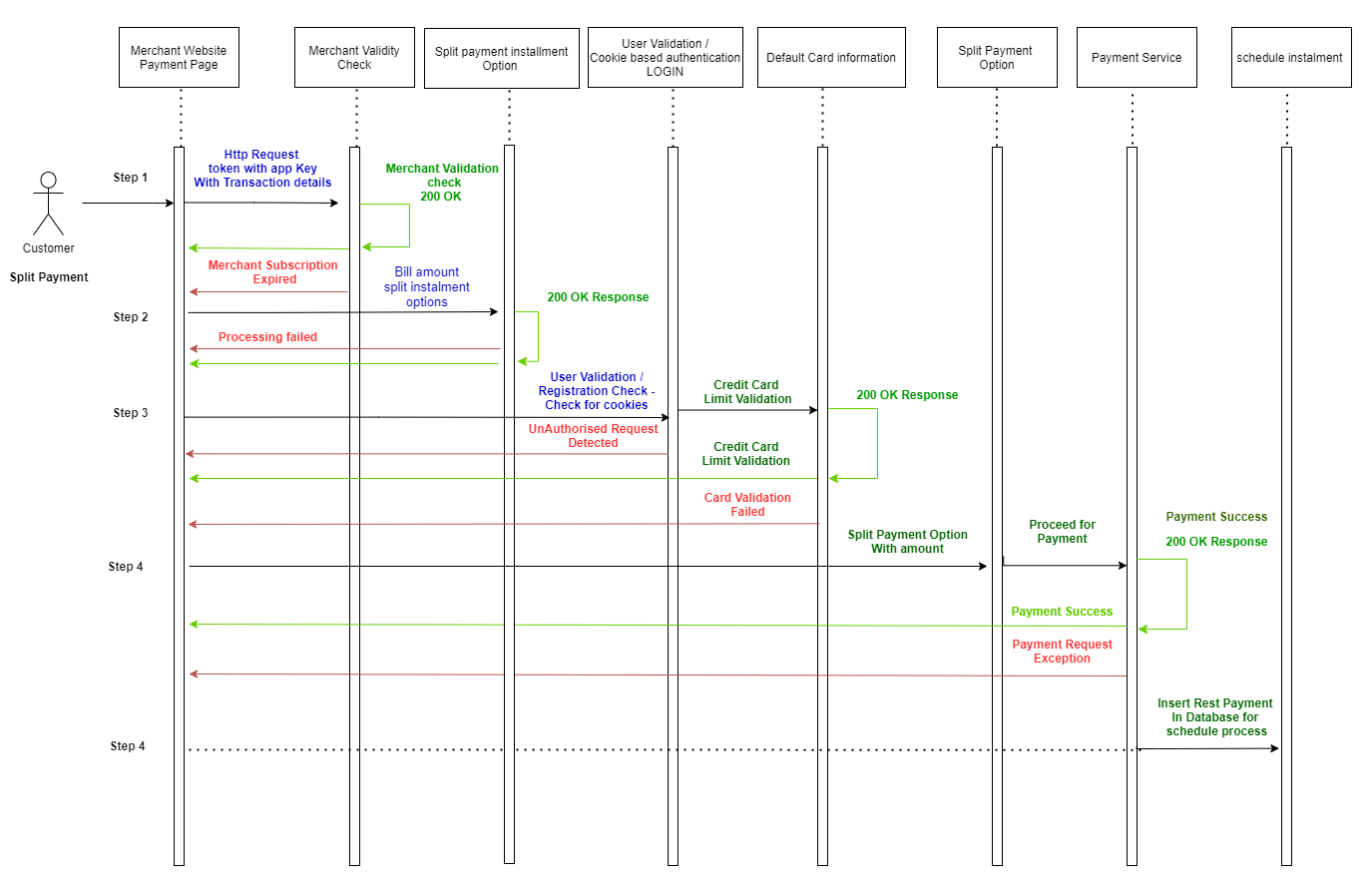
Step 1: Merchant Website Payment Page – When End Customer will be on checkout page, One HTTP request will be generated from Merchant website to K2 API which will contain Required details like end point headers and body parameters. This request will check for the merchant is valid or not also it will check Merchant Subscription plan is active or not. And gives back the success or failure response. If this request is authenticated and merchant is valid merchant, then new K2 transaction request ID will be generated and given back with in request also it will be set in cache server will date and time and transaction type Like {“THuhad325uh83q==: 10/11/2020 03:41:08 PM ,OneCLick”} , here “THuhad325uh83q==” will be encrypted key and value will be “10/11/2020 03:41:08 PM” (this will also be encrypted) So that in next step same transaction ID will be passed and then identification of the source will be done properly. In this step data will be returned from cache server. Cache server data will be updated every day. If cache server is not available, then data will be fetch from directly database. If Merchant subscription plan is not active, then API will return as Merchant Subscription Is over.

Step 2: Next request will be to validate User Information and Card validation, when second validation api request will be initiated with K2 transaction ID and first at merchant level it will be check that cookies are present or not if yes then session variable from cookie will be taken and sent to K2 api to validate, In this step K2 Transaction id will be validated first and system will check for the valid date time if request is has come after one hour is it not be valid request Merchant has to regenerate first level reference key again. Once user is validated then next step will be to check card is valid or not if any of the step has an issue then request will return will appropriate exception. If User is validated properly and card is also valid the another Second level K2 transaction ID will be generated returned from request and first level transaction ID will be deactivated. Previous transaction id will be marked as step done. Before returning successes response to request data of customer card (token) will be set in cache server so that very next step is payment step so that request should not go for database calling. And data will be set with key and pair format where new generated reference key will be key and value till be customer card token with date and time in encrypted JSON format.

Step 3: Final Step - This step will initiate when second step is completed, this step will be initiated when user will click on the pay button on merchant UI. Once user click on the pay button with second step reference key request will be send with amount and merchant transaction details to K2 api. Request will be validated first then will check for second step reference key if reference key doesn’t found in request then request will be rejected. Once request has appropriate second step reference key then value will be fetched and will check for the date and time form the cached value if time is more than 10 min then that request will be invalid and merchant has to reinitiate for the second step K2 transaction reference key again. If user is request timing is proper the card token and other transaction details will be used for payment gateway process. Payment will be initiated and once get the success response the data will be stored inside database and second reference key will be deactivated. Once second level key is deactivated then response will be given back to merchant website.

## Split Payment Process

**Split Payment Process**



Step 1: Merchant Website Payment Page – End Customer will be on checkout page on load of page merchant has to call K2 merchant validation API with api key provided to merchant. K2 system will validate the Merchant api request and will check for api request and then validate api key. Once api key is validated next step will be to check merchant is valid or not if merchant is valid and active then K2 transaction ID is will be generated and in response transaction ID will be given back, transaction ID details will be saved in cache server with date and time and transaction type. If merchant is not valid then exception will be thrown in api response. This information will be serve from cache server for faster response. If cache server is not able then from database information will be served.

Step 2: Upon clicking the Split Payment another api call will be there with first level transaction ID to get the split amount for purchase amount. This amount will return month wise or year wise list for payment. Once data receive from api merchant has to show on his website. In this step first level transaction ID will be validated first then next split option will be calculated, while giving response new second level ID will be generated and first level ID will be deactivated and then only response will be deactivated.

Step 3: When end customer selects spilt options and click pay button k2 system will check for the cookies in browser and token will be pass for the authentication, Here System will check for the browser secure cookies and fetch session variable token and also checks for the cookies expiry. If cookie is not expired and session variable token will be send to K2 API to validate user with in request Second level Transaction ID will be validated and then only user validation process will start, if api is authenticated successfully then new transaction key will be generated for further process and in response third transaction ID will be passed and transaction ID details will be saved in cache server so that next payment step card token will be fetched from cache not from database. In third step if authentication is not done properly or cookie is expired at merchant level then system will return login page so that end customer has to be authenticated again. Also if the customer is not registered then registration link will be prompted to end customer and he will be redirected to K2 marketing page.

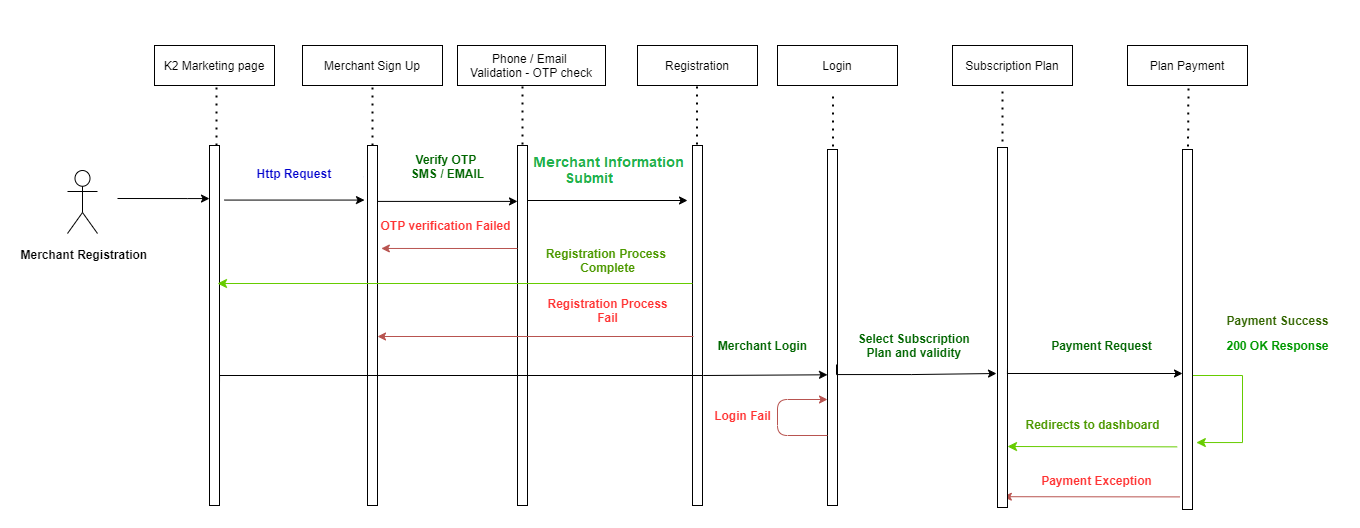
Step 4: Once third step is completed then next step is to validate end customer credit card for this card access API will be called with token and then in return card details will be provided in which, K2 system will check for the card available, if limit is enough for transaction then this transaction will be further moved else returned and exception as insufficient limit in card. Once this then is done then second level transaction ID will be deactivated and new transaction ID will be passed in response.

Step 5: In this Step End customer will select Split option and then split amount will be fetched from Server side and then displayed on the merchant website.

Step 6: Once split payment option is selected then system initiate payment request to K2 api with fouth level transaction ID and purchase details. In this process card access first step will be validating fourth level transaction ID and then card access payment api will be called and then rest details will be saved in database for further transaction. If the payment gets failed, then appropriate exception will be shown on the Merchant payment page.

## Merchant Registration Process

**Merchant Registration Process**



Step 1: Merchant registration process will start from K2 Marketing page. When merchant will enter all details, Merchant sign up page will be validated first while entering details then merchant will have to Validate email and SMS OTP.

Step 2: Once all data is confirmed and validated form request will be submitted and notification (SMS / Email) will triggered to mentioned email id. In this email User id and auto generated password will be sent with login url.

Step 3: While Login User id and password will be checked and in login http request all the encrypted data will be passed in the header. Once merchant is authenticated then he will be redirected to dashboard, with subscription menu option on navigation bar. He can select subscription plans.

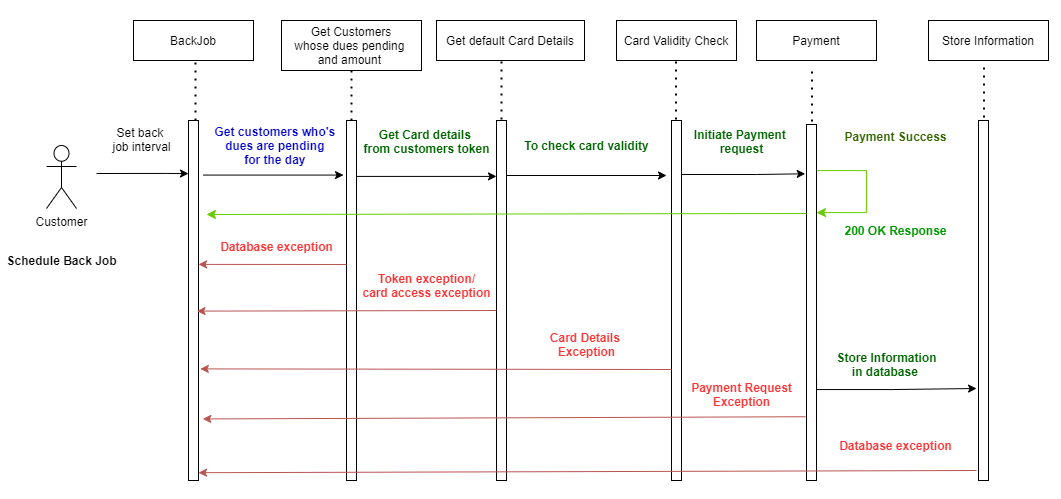
Step 4: All Subscription Plans will be fetched from database and merchant can select only one plan with listed available period.

Step 5: Once plan is selected then next step is payment process where card details will be validating and if card details are validated then only payment request will be initiate to card access payment API.

Notification mail and SMS will be triggered to merchant registered email id and sms. Also he can able to see his last transaction in the transaction option in navigation bar. After successful payment API key will be sent to merchant email id with appropriate documentation so that merchant can do integration with k2 API.

## Customer Schedule Payment Process

**Customer Schedule Payment Process**



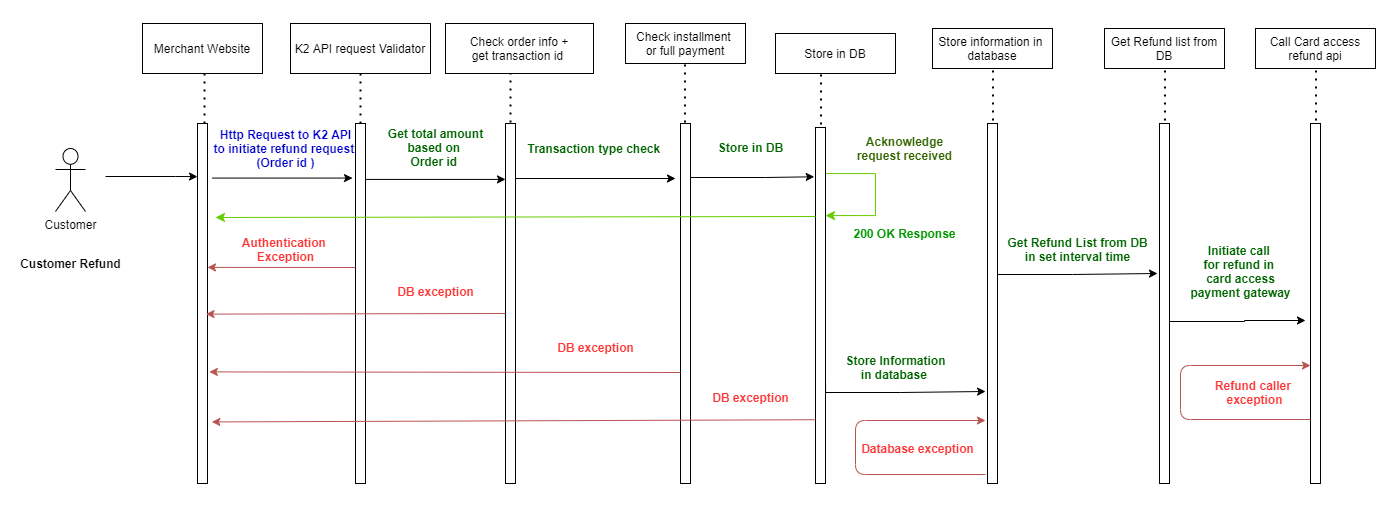
Step 1: Split payment dues will be clear from back job when scheduled back job will run in first step it will fetch details list from database, this list will contain Customers information with due amount.

Step 2: From customer’s information default card details will be fetched and then card validity and authorization will be check in this process card access card validity checker API will be called.

Step 3: If validity is proper then payment will card access payment API will be called and due amount will be passed, after successful payment notification will be trigger to customer with same fetched details.

## Customer Refund Payment Process

**Customer Refund Payment Process**



Step 1: Customer refund process will be in two phase, first phase will contain request acknowledgment and second will be back job will trigger refund payment request to card access API. In the initial phase request will be generated from merchant website and K2 refund API will be called order id will be passed,

Step 2: K2 API will take request and start validating request after validation done request will move further

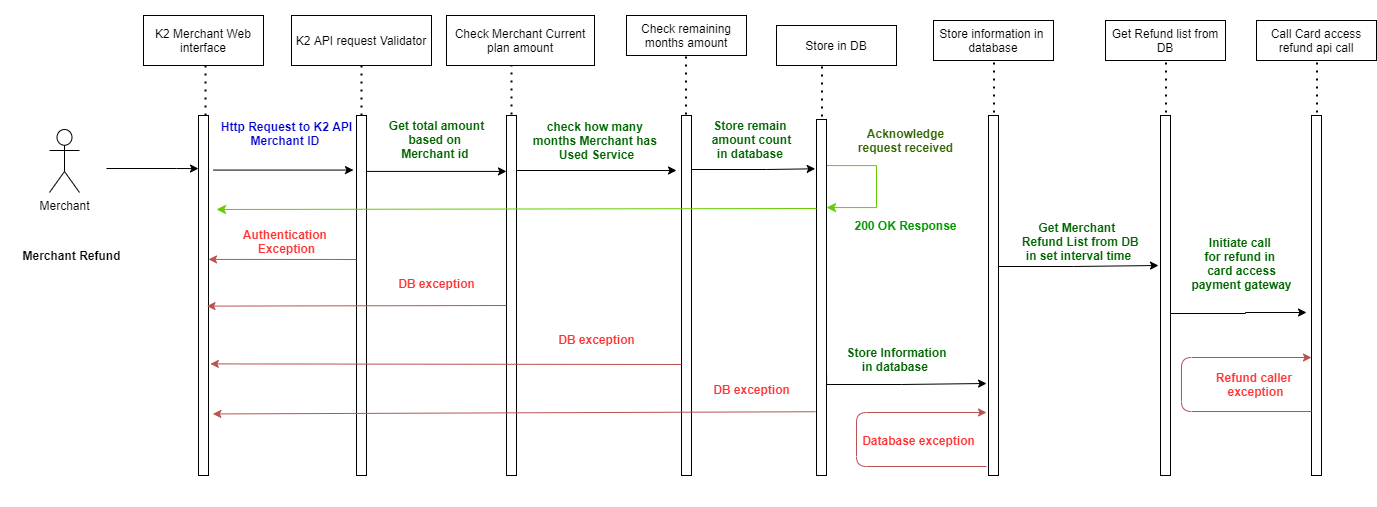
Step 3: From request body order id will be fetched and query will be done in database for transaction id and transaction details, once details are fetch what type of request (split / one click) according to that amount to be entered in database.

Step 4: once all details are fetched it will be inserted in database.

Step 5: On schedule time back job will run and fetch refund details and then send call card access refund API. once API calling is successful then notification will be triggered from back job to customers.

## Merchant Refund Process

**Merchant Refund Process**



Step 1: Merchant will initiate refund request from k2 web app, this request will pass to merchant refund API . API request will be verified first and then request will move forward.

Step 2: Merchant current plan details will be fetched and then amount will be decided.

Step 3: All the fetched details will be store inside database and acknowledgement response will be given back.

Step 4: Back job will get all merchants refund list with their details and call card access payment API for the refund to the merchant. Once request is successful notification will be triggered to merchant and then merchant plans will be suspended from database.

# Merchant API DOCUMENTATION

API Documentation will guide merchants for integration of K2 API request response structure.

Format of the Request Response will be in JSON.

**Request**

|  |  |  |
| --- | --- | --- |
| **SR NO** | **Name** | **Description** |
| 1 | URL | End point where , Merchants will be accessing methods of the K2 System. |
| 2 | Headers | All K2 API will contain headers and this Headers will be used for authenticating request,  API Header Like , Domain Name – App key – User id – Password  will be passed. |
| 3 | Body | Request data will be passed inside this property. |
| 4 | Content-type | Content-Type header field is used to specify the nature of the data in the body Property. |

**Response**

|  |  |  |
| --- | --- | --- |
| **SR NO** | **Name** | **Description** |
| 1 | Message | API response status code message will be shown in the property |
| 2 | Status | Status will be Boolean , true or false |
| 3 | Response data | This property will give data which was requested |
| 4 | Status Code | K2 API will return multiple status code with their description in message property |