Problem statement:

The incumbent employee benefit & lifestyle player Akme has a big presence in physical store merchants acquiring business globally. However they wish to now expand to digital payments so that payments can be made to online food ordering and delivery platforms like Uber Eat and online travel aggregators like Booking.com.
Akme is seeking your help to design a system which can accept payments from various sources of funds - Own benefit vouchers (meal, travel, gym, gadget etc.), Third party credit card, debit card, net banking etc.
Submission should include:
1. A MVP: which is basically a document that provides exhaustive explanation of:
APIs
Feature set
Back end modules
User interface catering to every possible user persona (wireframes to a certain level of detail)
User personas you wish to cater to via the proposed user interface
MVP should address the below mentioned use cases most importantly

2. A comprehensive product roadmap for MVP: which details APIs and interface that Akme can give to the merchants or the merchant aggregators. This should be mapped on a timeline basis any prioritization framework of your choice.

Use case to be mandatory addressed (map these use cases to proposed API and modules):
As a customer, I am placing a food order worth INR 300 on Uber Eats, I have Akme Food Vouchers worth INR 200 and remaining INR 100 should be charged to my credit card. The MVP should let me do the payment to Uber Eats in 1 transaction.
As Uber Eats, I should be able to refund entire amount i.e. INR 300
As Uber Eats, I should be able to refund partial amount say INR 50

Note: Do not assume that funds will be added to Akme benefit vouchers. Rather design a system which will draw funds from (i) Akme and (ii) credit card, debit card, net banking etc independently but present this as 1 transaction to merchants.

This is a system design assignment. The assignment will be reviewed for technical soundness and aesthetics and how well you are solving for stated use cases.
Business, growth or marketing aspects need not be addressed in your submission.

zeta

Objective

Akme is an employee benefit and lifestyle provider with a presence in a large number of physical stores. Akme now wants to enable digital payments so that their services can be used on online platforms.

Goal

Design a system capable of accepting digital payments from various sources of funds.

Sources of Funds

- Credit Cards
- Debit Cards
- Net Banking
- UPI
- Benefit Vouchers (travel, gym, meal, gadget)
- Wallets
- Cash on delivery (Enable for few platforms like food delivery/e-commerce)

⚠ **Assumption** - *For Cash on delivery cases, the user is not permitted to do part transactions, ie pay Rs 100 using a voucher and Rs 200 using cash if the total bill is Rs 300. Hence, for COD this solution is not applicable.*

Feature Set

As we are designing a minimum viable product, our feature set needs to include core features required to effectively deploy the product. It should cater to core needs of the early adopters, who would help us with feedback. The feature set for Akme’s digital payment system is as follows.

⚠ **Assumption**: Assuming the company pays a senior engineer 180,000 per month. Cost of an engineer per hour is 800/hr. Assuming that 3 engineers are available for the MVP.
⚠ **Statement**: KP Factor is a value between 1 to 100 indicating how much it contributes towards KPI(Key Performance Indicator).
Value = ((KP Factor/ Cost)*100)

Feature No.	**Feature**	**Hours**	**Cost**	**KPI**	**KP Factor**	**Value**
F1	Link between Merchant app and Akme system.	One engineer, 20 hours.	16,000	Number of transactions in the last 30 days.	80	0.5
F2	Pay using Voucher	Two engineers , 30 hours.	48,000	80% of payments through vouchers.	80	0.166
F3	Pay using voucher + other sources	Three engineers , 60 hours each.	1,44,000	Cover 100% of online payment options	100	0.069
F4	Process refunds on transactions.	Two engineers , 40 hours each.	64,000	Speed of processing refunds and success in reconciliation	100	0.156
F5	Mobile SDKs	One engineer - 80 hours, two engineers - 50 hours each	1,44,000	Number of transactions through mobiles.	90	0.062
F6	Testing and Quality Assurance	One tester, 40 hours.	16,000	Fault tolerant app	100	0.625

The Value column from the above table provides us with a reasonable metric on how much bang for the buck a feature can bring to Akme. This could also help us to plan the product roadmap.

Design

User Personas

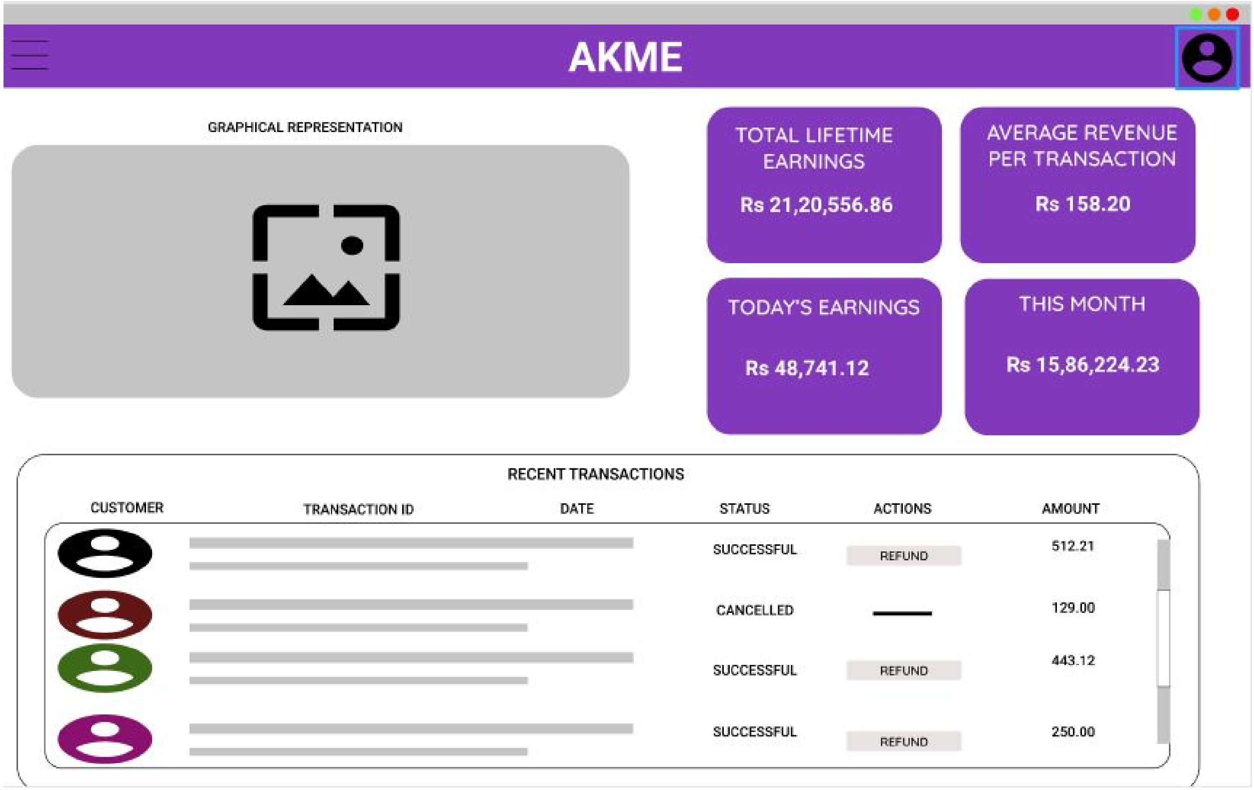
The user personas defined below are imaginary characters who would be different types of users of the product.

Persona	Description
P1	Harsha works as a software developer at a food delivery app, and wants to add integration of Akme's payment services to his app. He will be needing to look at the API documentation while building the integration.
P2	Rashi works as a customer service person at a travel booking app, and wants to tend to the payment problems of the customer. She looks at the current status of payment and also has the power to refund the amount paid.
P3	Rahim is an employee at a company which distributes Akme vouchers for tax benefits on food. He wants to use the food voucher at an online food delivery app to buy food from his favorite restaurant.

User Interface

User Interface for Persona P2:

Dashboard for Akme:



AKME

GRAPHICAL REPRESENTATION

TOTAL LIFETIME EARNINGS

Rs 21,20,556.86

AVERAGE REVENUE PER TRANSACTION

Rs 158.20

TODAY'S EARNINGS

Rs 48,741.12

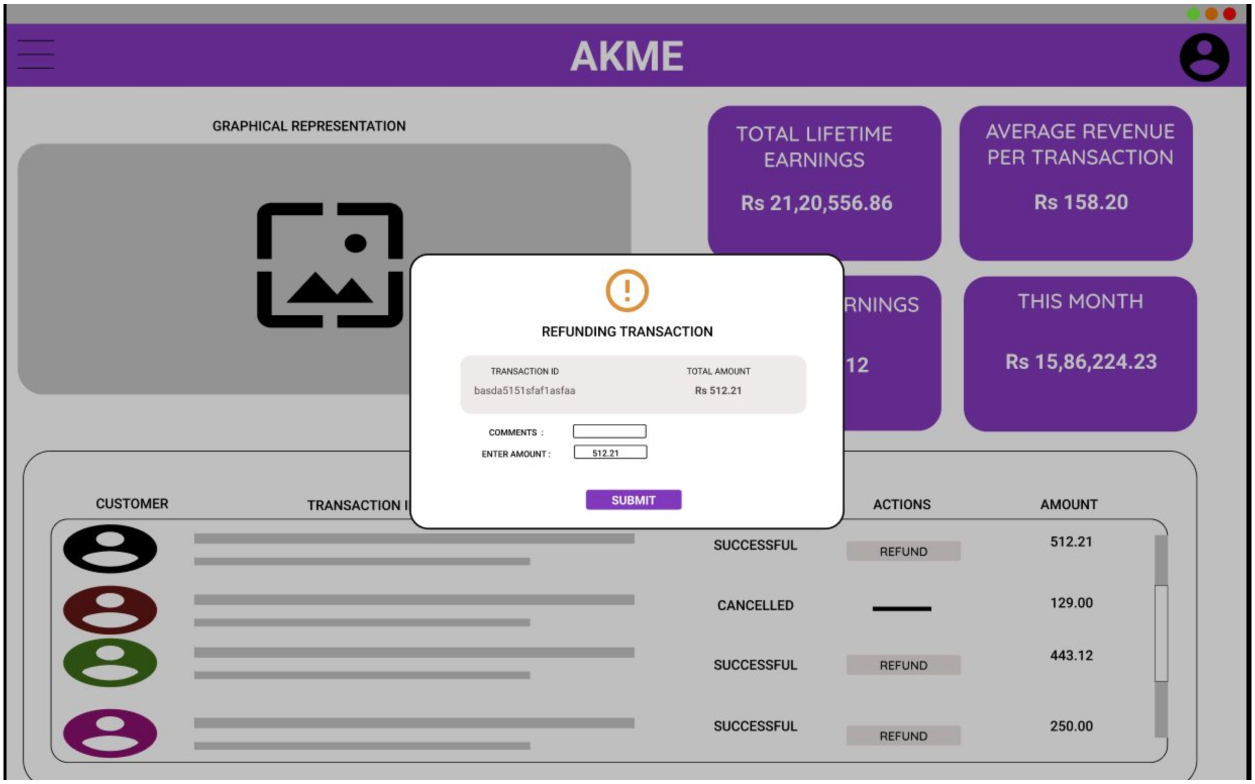
THIS MONTH

Rs 15,86,224.23

RECENT TRANSACTIONS

CUSTOMER	TRANSACTION ID	DATE	STATUS	ACTIONS	AMOUNT
			SUCCESSFUL	REFUND	512.21
			CANCELLED	—	129.00
			SUCCESSFUL	REFUND	443.12
			SUCCESSFUL	REFUND	250.00

The Full/Partial Refund option after clicking on “REFUND” button:



TOTAL LIFETIME EARNINGS
Rs 21,20,556.86

AVERAGE REVENUE PER TRANSACTION
Rs 158.20

RNINGS
12

THIS MONTH
Rs 15,86,224.23

REFUNDING TRANSACTION

TRANSACTION ID	TOTAL AMOUNT
basda5151sfaf1asfaa	Rs 512.21

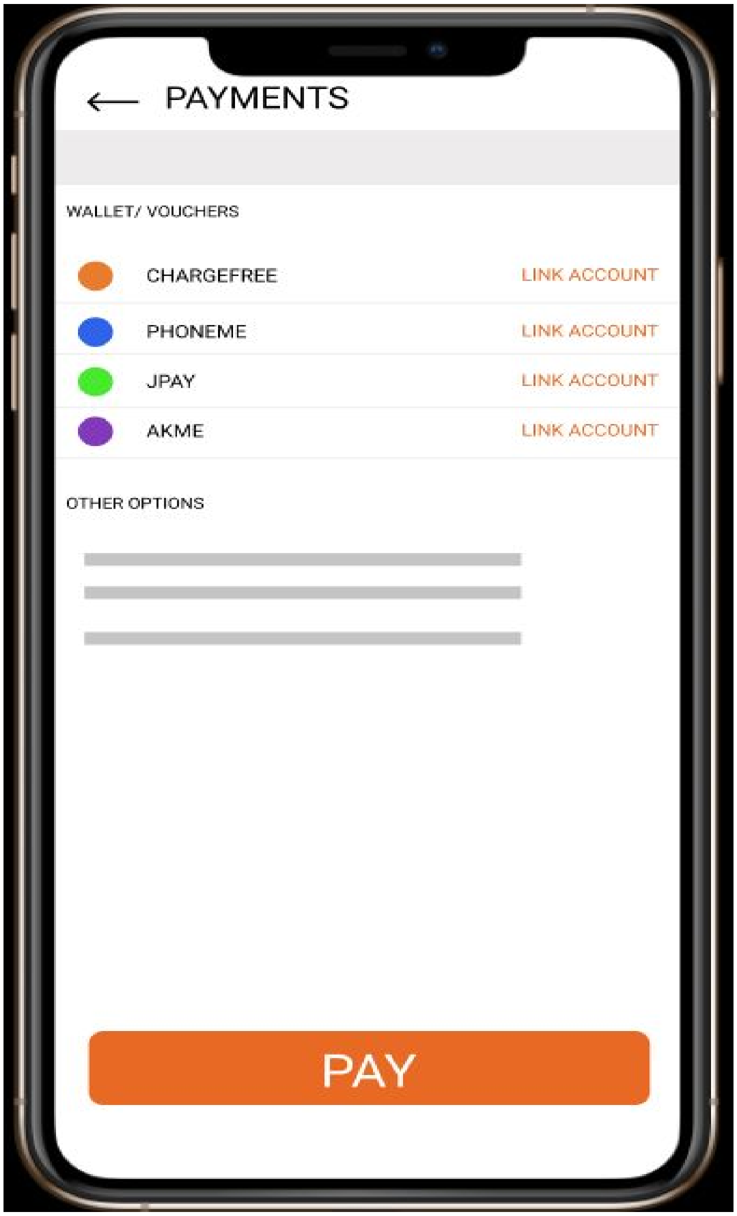
COMMENTS :
ENTER AMOUNT : 512.21

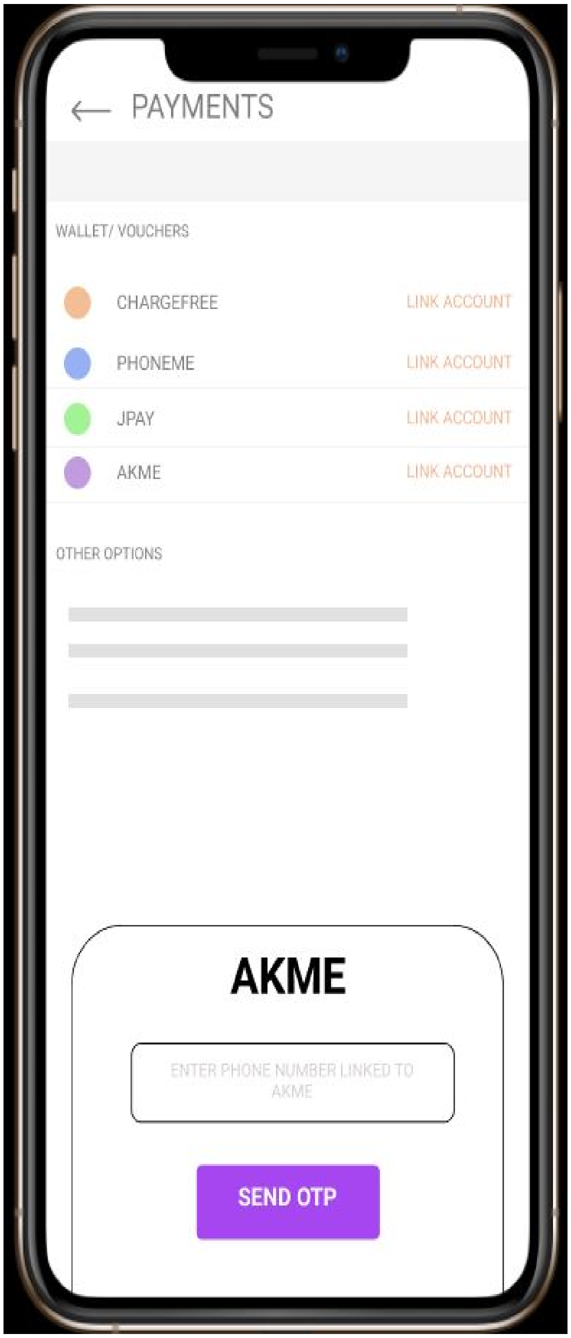
SUBMIT

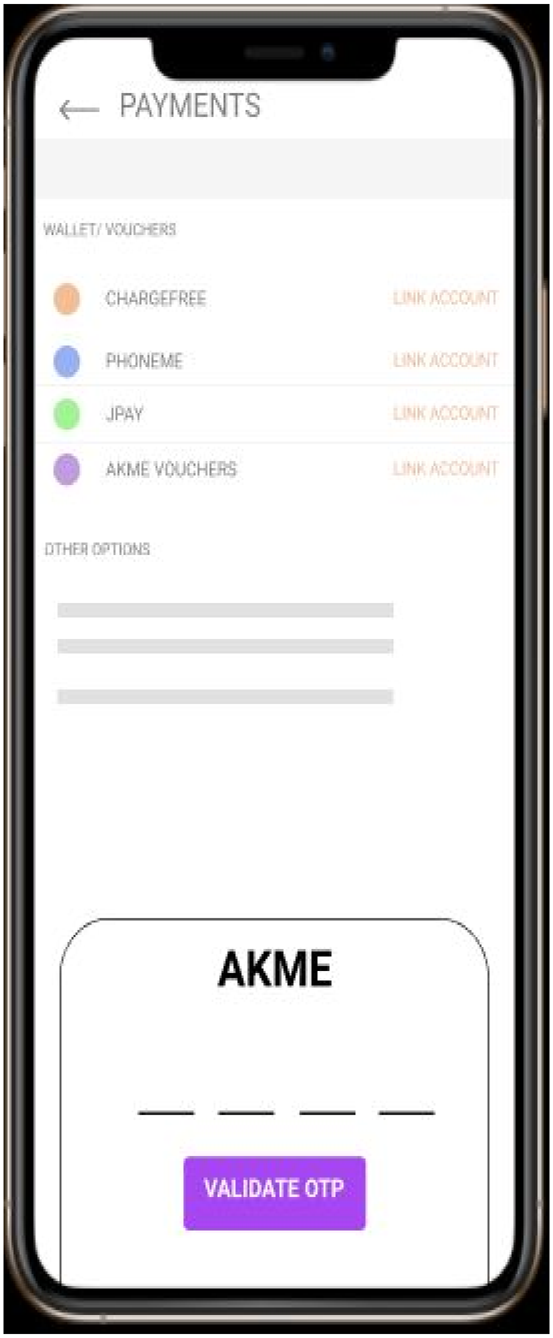
CUSTOMER	TRANSACTION I		ACTIONS	AMOUNT
		SUCCESSFUL	REFUND	512.21
		CANCELLED	—	129.00
		SUCCESSFUL	REFUND	443.12
		SUCCESSFUL	REFUND	250.00

User Interface for Persona P3:

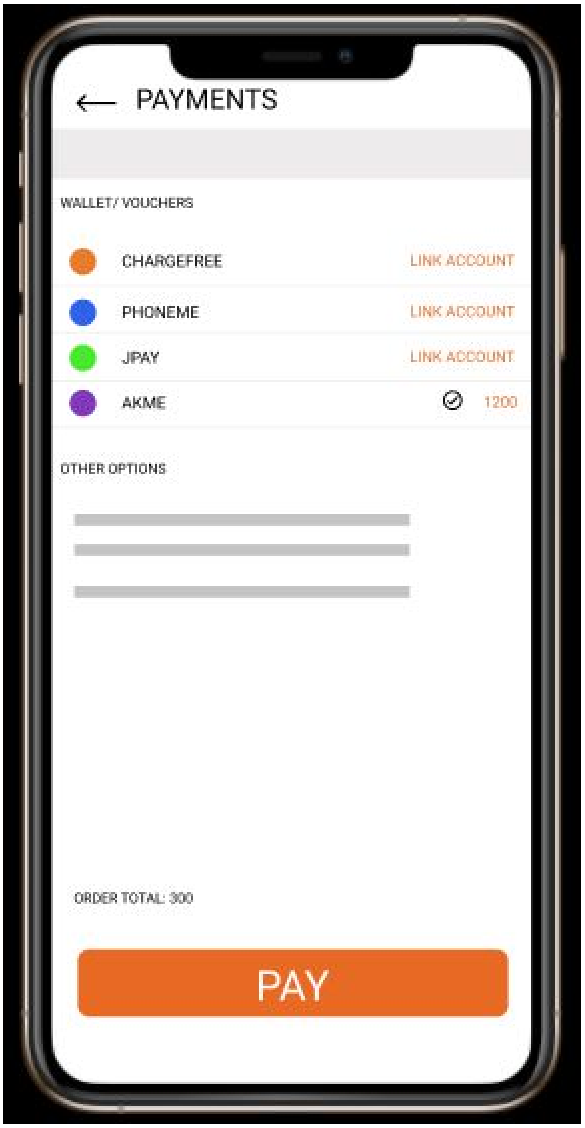
Linking Akme with the Merchant app:



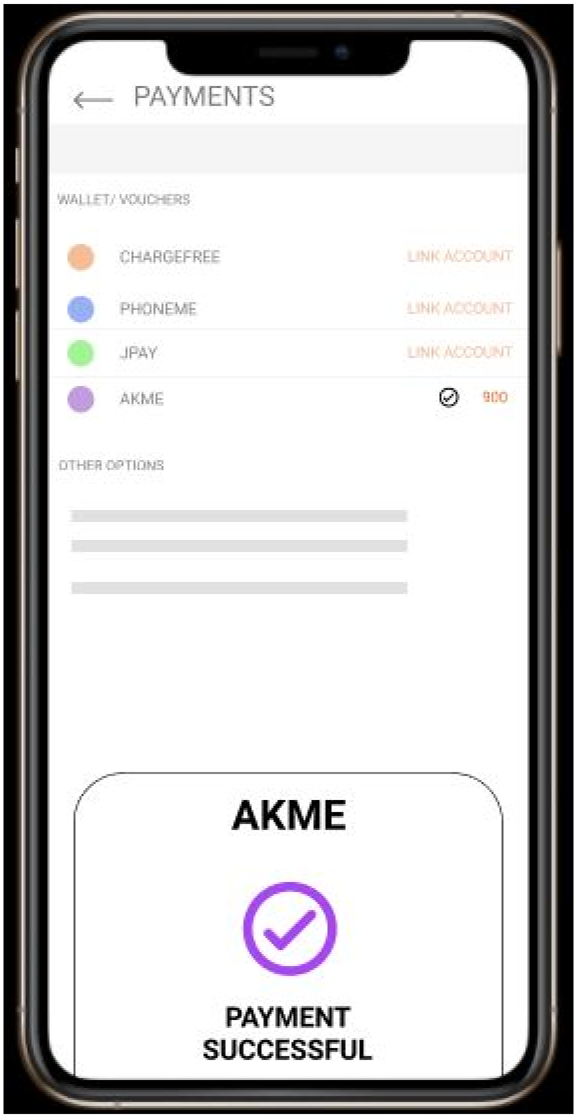

← PAYMENTS
WALLET/ VOUCHERS
CHARGEFREE
LINK ACCOUNT
PHONEME
LINK ACCOUNT
JPAY
LINK ACCOUNT
AKME
LINK ACCOUNT
OTHER OPTIONS
AKME
ENTER PHONE NUMBER LINKED TO AKME
SEND OTP


← PAYMENTS
WALLET/ VOUCHERS
CHARGEFREE
LINK ACCOUNT
PHONEME
LINK ACCOUNT
JPAY
LINK ACCOUNT
AKME VOUCHERS
LINK ACCOUNT
OTHER OPTIONS
AKME
VALIDATE OTP

Payment flow when there is enough balance in the Akme Voucher:

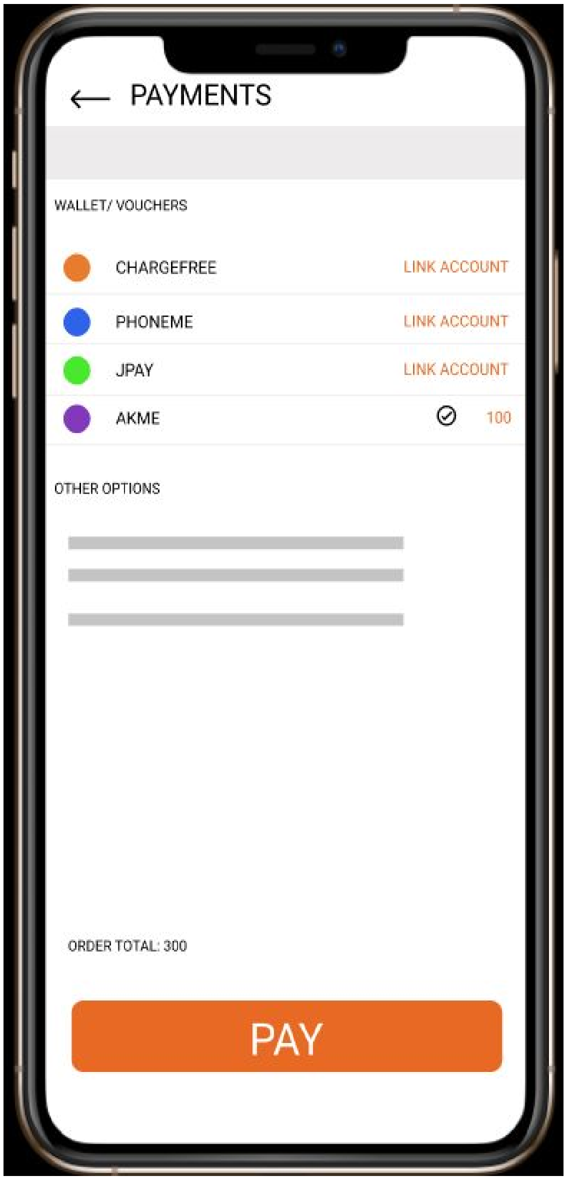


ORDER TOTAL: 300



PAYMENT SUCCESSFUL

Payment flow when there’s not enough balance for the transaction:



← PAYMENTS

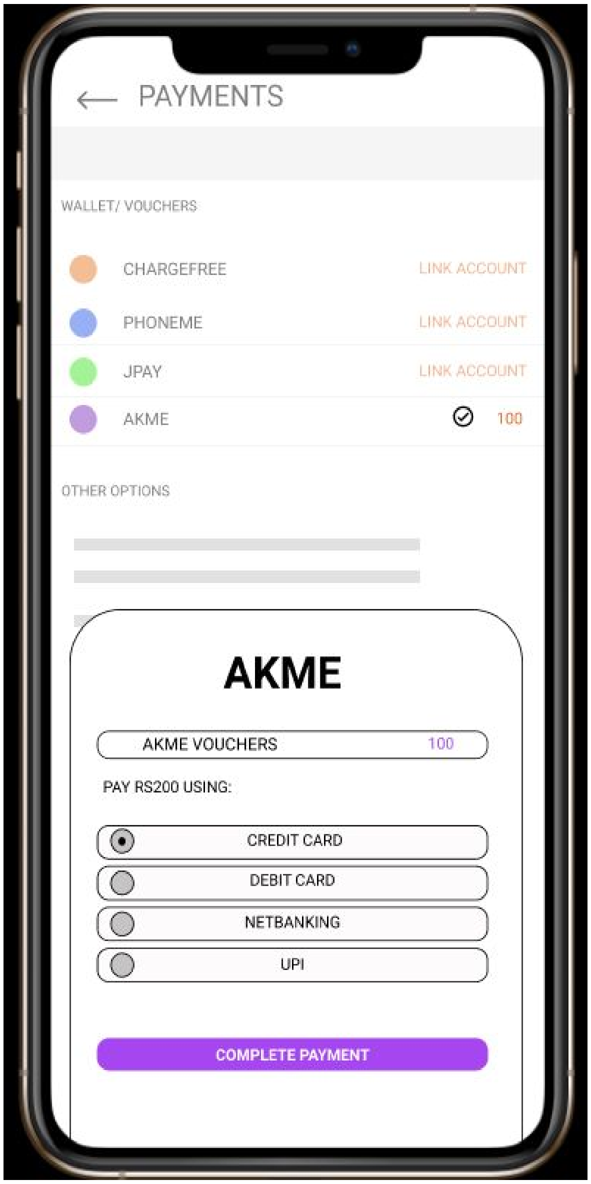
WALLET/ VOUCHERS

●	CHARGEFREE	LINK ACCOUNT
●	PHONEME	LINK ACCOUNT
●	JPAY	LINK ACCOUNT
●	AKME	✓ 100

OTHER OPTIONS

ORDER TOTAL: 300

PAY



← PAYMENTS

WALLET/ VOUCHERS

●	CHARGEFREE	LINK ACCOUNT
●	PHONEME	LINK ACCOUNT
●	JPAY	LINK ACCOUNT
●	AKME	✓ 100

OTHER OPTIONS

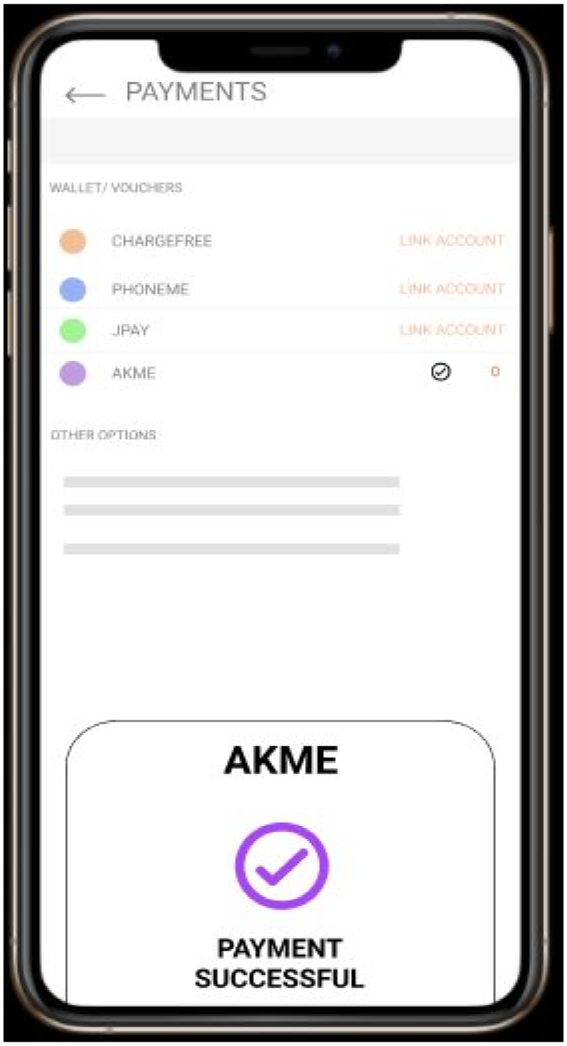
AKME

AKME VOUCHERS 100

PAY RS200 USING:

- ● CREDIT CARD
- ○ DEBIT CARD
- ○ NETBANKING
- ○ UPI

COMPLETE PAYMENT



← PAYMENTS

WALLET/ VOUCHERS

● CHARGEFREE		LINK ACCOUNT
● PHONEME		LINK ACCOUNT
● JPAY		LINK ACCOUNT
● AKME	✓	0

OTHER OPTIONS

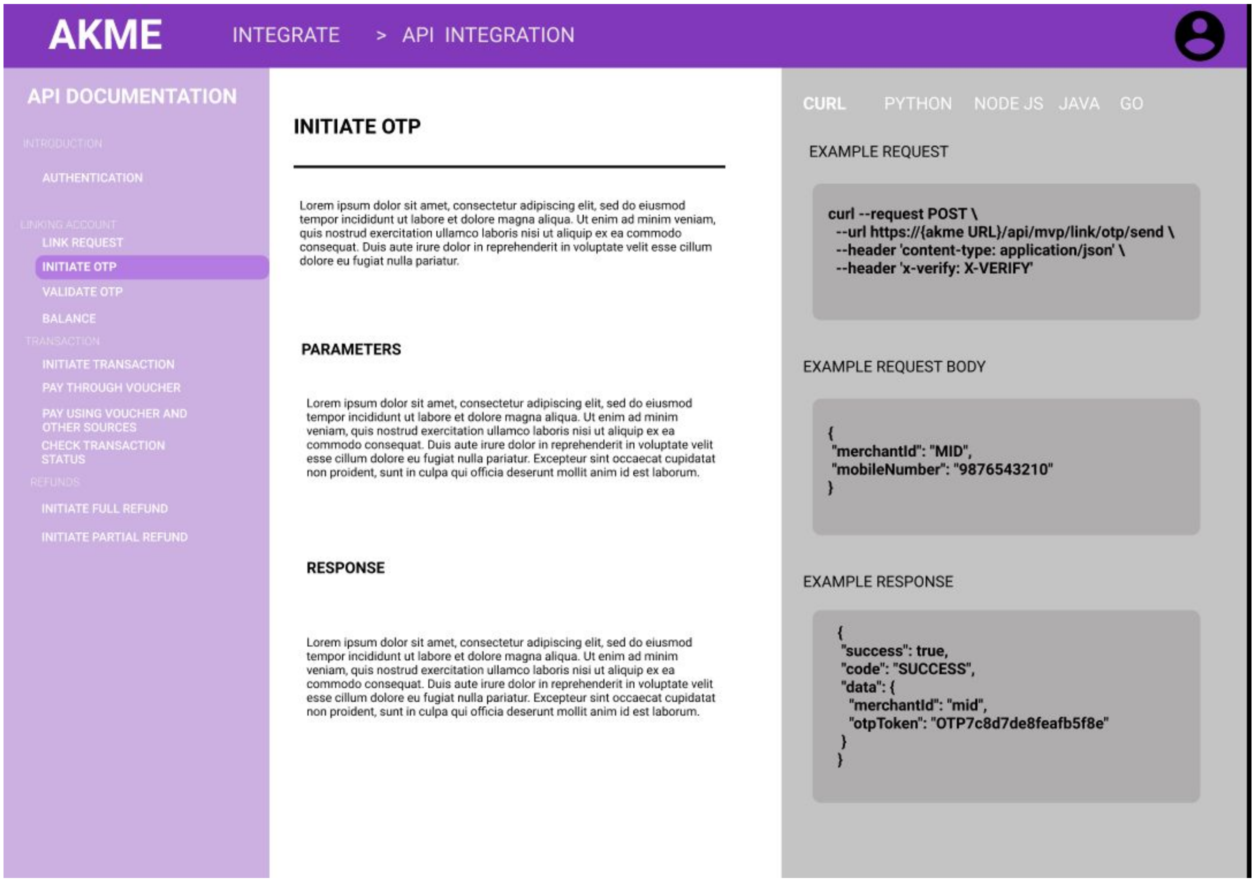
AKME

✓

PAYMENT
SUCCESSFUL

Persona P1:

P1 is an Integration Engineer and will be using the API documentation.



INITIATE OTP

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

PARAMETERS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

RESPONSE

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

EXAMPLE REQUEST

```
curl --request POST \
  --url https://{akme URL}/api/mvp/link/otp/send \
  --header 'content-type: application/json' \
  --header 'x-verify: X-VERIFY'
```

EXAMPLE REQUEST BODY

```
{
 "merchantId": "MID",
 "mobileNumber": "9876543210"
}
```

EXAMPLE RESPONSE

```
{
 "success": true,
 "code": "SUCCESS",
 "data": {
   "merchantId": "mid",
   "otpToken": "OTP7c8d7de8feafb5f8e"
 }
}
```

Technology

⚠ **Assumption**: Akme Vouchers already exist in the form of voucher cards. They are created and topped up by the organisation which provides them.

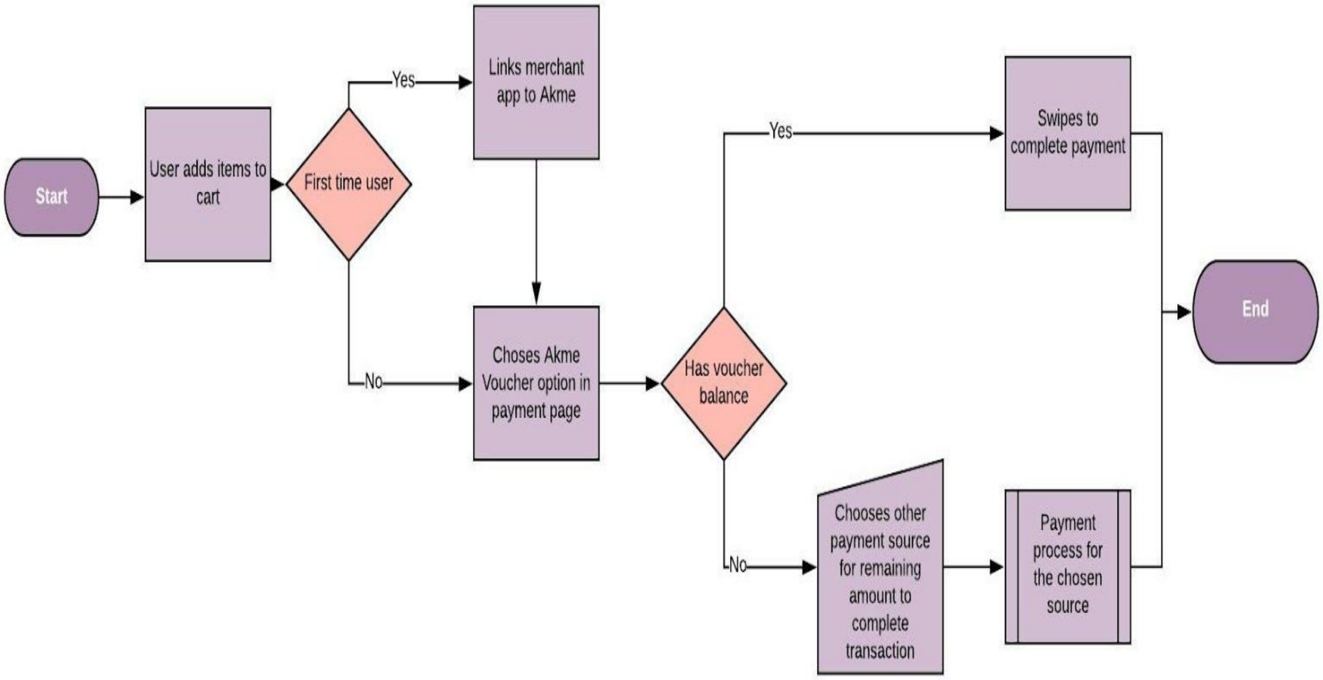
Use cases

The use cases defined below are the key job-to-be-done(JTBD) for the online payment engine.

Use case	**User need**
UC1	As a user, I want to initiate a transaction.
UC2	As a user, I want to pay using two methods of payment for a single transaction.
UC3	As a user, I want to cancel a transaction.
UC4	As a platform, I want to process refunds of a transaction.
UC5	As a platform, I want to process partial refunds of a transaction.
UC6	As both user and platform, we want to view the transaction details.
UC7	As a platform, I want to check the balance before initiating a transaction.
UC8	As a platform, I want the user to link the Akme vouchers with my platform.
UC9	As a platform, I want to know the current status of the transaction

Payment Flow

1. The user adds items to cart on the merchant website and is ready to checkout.
2. Among the payment options, the user chooses the Akme option. While using the option for the first time, they will need to link their Akme Voucher account to the merchant.
3. After being linked, it'll display the balance left in the voucher.
4. When the user clicks Akme, a transaction is created with details about the total amount.
5. The actual transaction execution takes place on Akme's domain.
6. Based on the balance in the voucher, the entire transaction amount can be divided between voucher and other sources of funds.
7. Once successful, Akme redirects back to the merchant with the status.

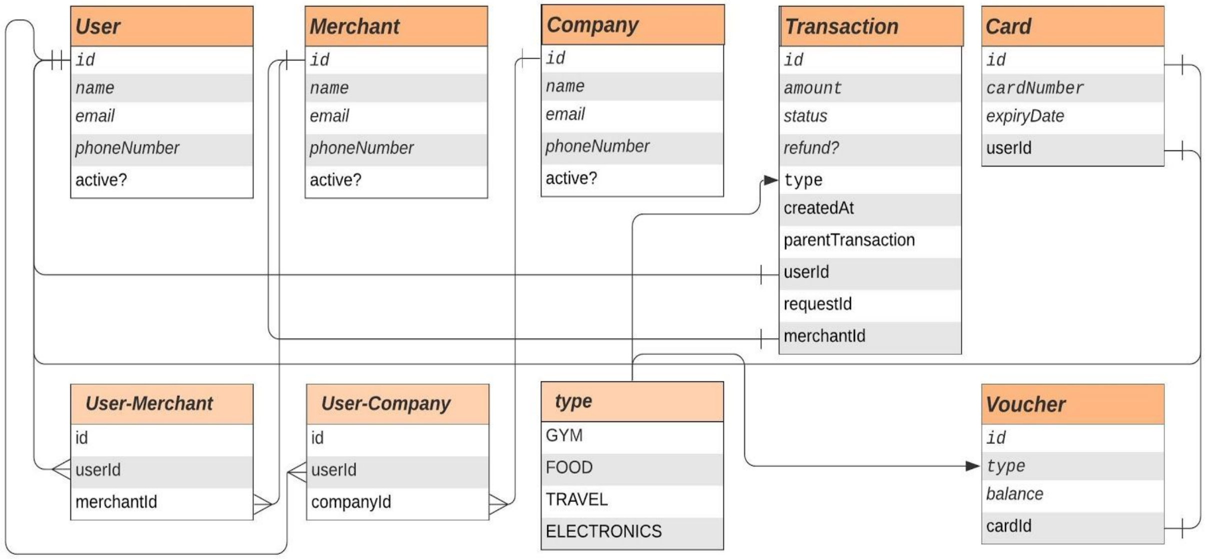

Start
User adds items to cart
First time user
Yes
Links merchant app to Akme
No
Choses Akme Voucher option in payment page
Has voucher balance
Yes
Swipes to complete payment
No
Chooses other payment source for remaining amount to complete transaction
Payment process for the chosen source
End

One card(Visa/Mastercard) is given to the employees. The cards are given to employees, who can use it at any payment terminal. This one card will have many vouchers under it like food, travel, gym, etc. Our payment system will deduct the amount from the voucher to which it is charged. For example, if the merchant is Swiggy, a `type: “FOOD”` flag is sent along with the request. Then our system will deduct from the food voucher under the card.

Backend Modules

Database

⚠ **Assumption**: An employee is given a card by VISA/Mastercard. This one card can be used for various purposes like food, gym, travel,etc.



APIs

⚠ **Assumption:** API version is mvp

API Path	Use case	Description
For Linking Akme with merchant		
`api/mvp/link/otp/send`	UC8 - Link merchant to Akme	Send OTP
`api/mvp/link/otp/verify`		Verify OTP code
`api/mvp/voucher/balance`	UC7 - Check balance	Returns balance
For initiating a transaction		
`api/mvp/transaction/debit`	UC1, UC2 - Debit amount.	Initiate Transaction
`api/mvp/transaction/cancel`	UC3- Cancel a transaction	Cancel a transaction
`api/mvp/transaction/{transactionId}`	UC6 - Details	Check the details of transaction
`api/mvp/source/save`	-	Remember a payment source

For refunding a transaction		
`api/mvp/transaction/refund`	UC4, UC5 - Full and Partial refund	Initiate a refund transaction.

1. **Initiate OTP**

   API: `api/mvp1/link/otp/send`

   Method: POST

   Request Params:

   - `merchantId,` – Merchant receives it when they onboard with Akme.
   - `mobileNumber`

   Response Fields:

   - `success,`
   - `code,`
   - `data: merchantId, otpToken`

2. **Verify OTP**

   API: `api/mvp/link/otp/verify`

   Method: POST

   Request Params:

   - `otp,`
   - `otpToken`

   Response Fields:

   - `success,`
   - `code,`
   - `data: merchantId,`
   - `authToken` – To be used as header in future requests

3. **Balance**

   API: `api/mvp/voucher/balance`

   Method: GET

   Header: `authToken`

   Response Fields:

   ```
   success,
   code,
   merchantId,
   balances: [
   {type,
   balance}]
   ```

4. **Transaction Debit initiate**

   API: `api/mvp/transaction/debit`

   Method: POST

   Header: `authToken`

   Request Params:

   ```
   merchantId,
   requestId,
   amount
   ```

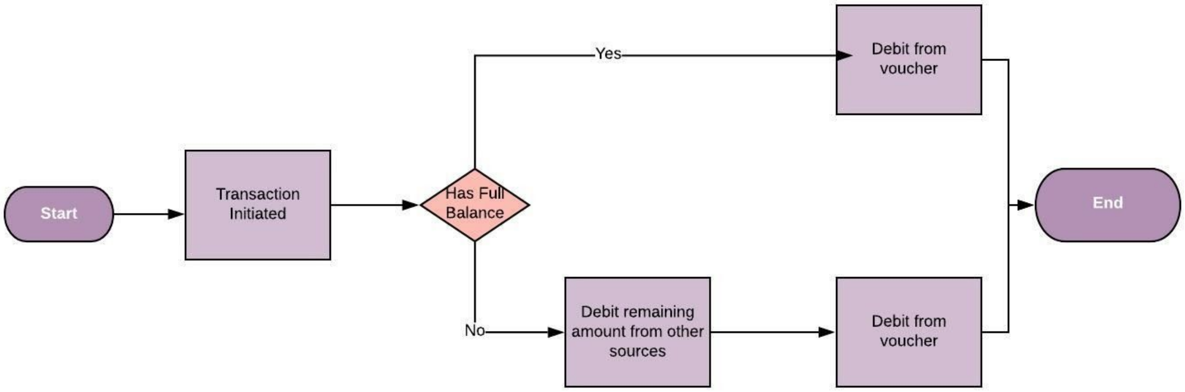
Response Fields:

```
success,
code,
merchantId,
requestId,
transactionId,
amount
```

Backend process:

If there’s enough balance, the amount is debited from the voucher. If there isn’t, then the transaction is internally divided into two child transactions, one for debit from another source and another for debit from voucher.

Initially when the request is first created, the status becomes “PENDING”. Then based on the below process, it keeps changing.



5. Transaction Cancel

API: `api/mvp/transaction/cancel`
Method: POST
Header: `authToken`
Request Params:

```
merchantId,
transactionId
```

Response Fields:

```
success,
code
```

6. Transaction Details

API: `api/mvp/transaction/{transactionId}`
Method: GET
Header: `authToken`
Response Fields:

```
success,
code,
transactionId,
requestId,
status,
amount,
createdAt
```

7. **Refund a Transaction**

API:

Method: POST

Header: `authToken`

Request Params:

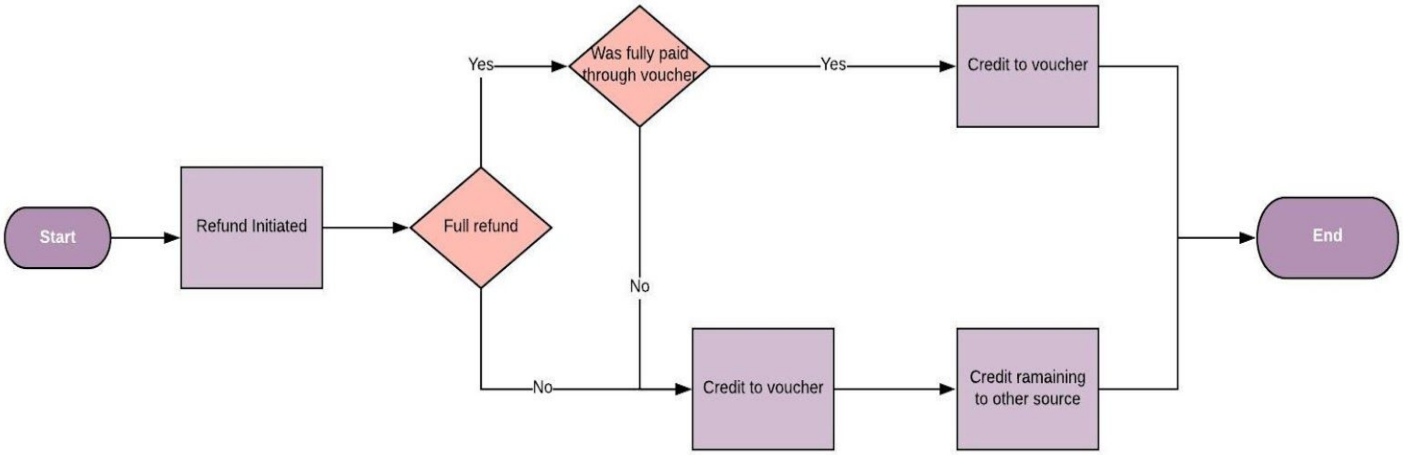
```
merchantId,
transactionId,
requestId,
amount
```
 – Amount for refund has to be a negative number.

Response Fields:

```
success,
Code,
refundTransactionId
```

Backend Process:

During the partial payment refund process, the first priority for credit is given to Akme vouchers and the second to other sources.



8. **Webhook Register**

API: `api/mvp/webhook/register`

Method: POST

Header: `authToken`

Request Params:

```
merchantId,
url,
event[]
```

Response Fields:

```
success,
code,
webhookId
```

9. **Software Development Kit (SDK) for Android and iOS.**

For mobile apps, Akme has to provide a SDK to help developers create applications. SDKs have to include the above APIs, interfaces.

Product Roadmap

Prioritization

Before prioritizing, let's list all items.

- Link merchant app to Akme voucher.
- Get balance in the voucher.
- Debit an amount from Akme voucher.
- Debit an amount from voucher + other sources and present it as one.
- Mobile SDKs for Android and iOS
- Design UI for linking account.
- Design UI for debit from voucher.
- Design UI for debit from multiple sources.
- Save a payment method.
- Design the database.
- Refund full amount.
- Refund partial amount.
- Design dashboard for merchants.
- Get stats for a certain merchant.
- Write API documentation.
- Webhook Register.

I'll be using the MoSCoW model of prioritization. The items will now be prioritised based on "Must have", "Should have", "Could have" and "Won't have"

Roadmap

All of the above items have been prioritised based on the MoSCoW model, divided into sprints, tasks and given timeframes in the below roadmap. You can view the roadmap by timeline by clicking on 'Calendar' on the top left above the table. Alternatively, you can view the tasks by priority by clicking in 'Priority' on the top left above the table.

⚠ Assumption: The timeline taken is between 22nd June to 24th July, i.e One month.

https://www.notion.so/0c77215f05b64447b88aeb999951e37d?v=407ae0dc50574953a81f68c66a6325b3