PROJECT REPORT ON

APIs for QR Code based Mobile Payments

Submitted by :
Sreyans Singhi
2nd Year, IT
IIIT, Bhubaneswar

Under the guidance of :
Dr. V.N. Sastry, Professor
IDRBT

INSTITUTE FOR DEVELOPMENT AND RESEARCH IN BANKING TECHNOLOGY, HYDERABAD

10th July, 2019

Contents

Certificate
Acknowledgement
Abstract

Chapter 1: Mobile Payment System 5
1.1 Introduction 5
1.2 History and Evolution 5
1.3 Types of Mobile Payment System in India 7
1.3.1 IMPS 7
1.3.2 UPI 9
1.3.3 AePS 11
1.3.4 NEFT 12
1.4 QR Codes 14
1.4.1 Introduction 14
1.4.2 Specifications 14
1.4.3 Scanning Process 15
1.4.4 Error Tolerance Pattern of QR Codes 16
1.4.5 Shapes of the QR Code 16
1.4.6 International Development 17
1.4.7 Bharat QR Codes 17
1.5 API 18

Chapter 2: Design and Development of APIs for QR Code based payments 20
2.1 Introduction 20
2.2 Modes of Digital Payments 20
2.3 Conclusion 34

Chapter 3: Code and Prototype Implementation 35
3.1 Introduction 35
3.2 Codes for generating QR Codes for various payment modes 35
3.3 Combining various Payment System 37

Chapter 4: Conclusion 41

Chapter 5: References 42

Institute for Development and Research in Banking Technology,

Road No. 1, Castle Hills, Masab Tank,
Hyderabad-500057

CERTIFICATE

This is to certify that Mr. Sreyans Singhi, pursuing B. Tech in Information Technology, 2nd Year, bearing ID No. B417039 from IIIT Bhubaneswar has done the project work on "APIs for QR Code based Mobile Payments" as an intern at Centre for Mobile Banking(CMB) lab at IDRBT, Hyderabad from May 13th,2019 to July 10th,2019 under my guidance.

Dr. V.N Sastry
Professor,
IDRBT, Hyderabad

ACKNOWLEDGEMENT

I am greatly indebted to the authorities of the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad for providing opportunity to work on the project titled "APIs for QR Code based Mobile Payments".

Firstly, I thank and express my solicit gratitude to Dr. V.N Sastry, Professor, IDRBT for his invaluable help and support which helped me a lot in successfully completing the project. His depth of knowledge in his field helped me a lot in keeping up with the project.

This project was beneficial for me at each and every step, at the same time it gave me the confidence to work in the real life and professional set up. I feel the experience gained during the project would lead me towards a good professional life.

I would also thank IIIT Bhubaneswar, for allowing me to participate in this Summer Internship Program.

Sreyans Singhi
2nd Year, IT
IIIT, Bhubaneswar

Abstract

The evolution in the payment technologies and the need for a secure and swift payment system led to the birth of Digital Payment System. Mainly designed with a vision to transform the payment system around the world into a digitally empowered system and knowledge economy, it made a huge impact on the payment system around the world. This project is about the development of various digital technologies for doing a transaction be it UPI, IMPS, NEFT or AEPS. The main focus is on creating QR Codes for different payment methods which eliminates the idea for the payer to fill the account number, IFSC Code or Aadhar number of the payee which is not only time saving but also efficient since the chances of payment going to a wrong entity is very low.. We have developed QR Codes for different modes of payments and also combined the QR Codes of different payments into single QR Code. Further, we also designed the APIs for each mode of payments and the request/response by entities involved in each mode of transaction. The main idea is: Once a user scans the QR Code from his/her payment app, he/she gets the account number, IFSC Code and other essential details of the beneficiary to do the transaction. After entering the amount and pin for authentication their transaction is being processed and completed

Chapter 1: Mobile Payment System

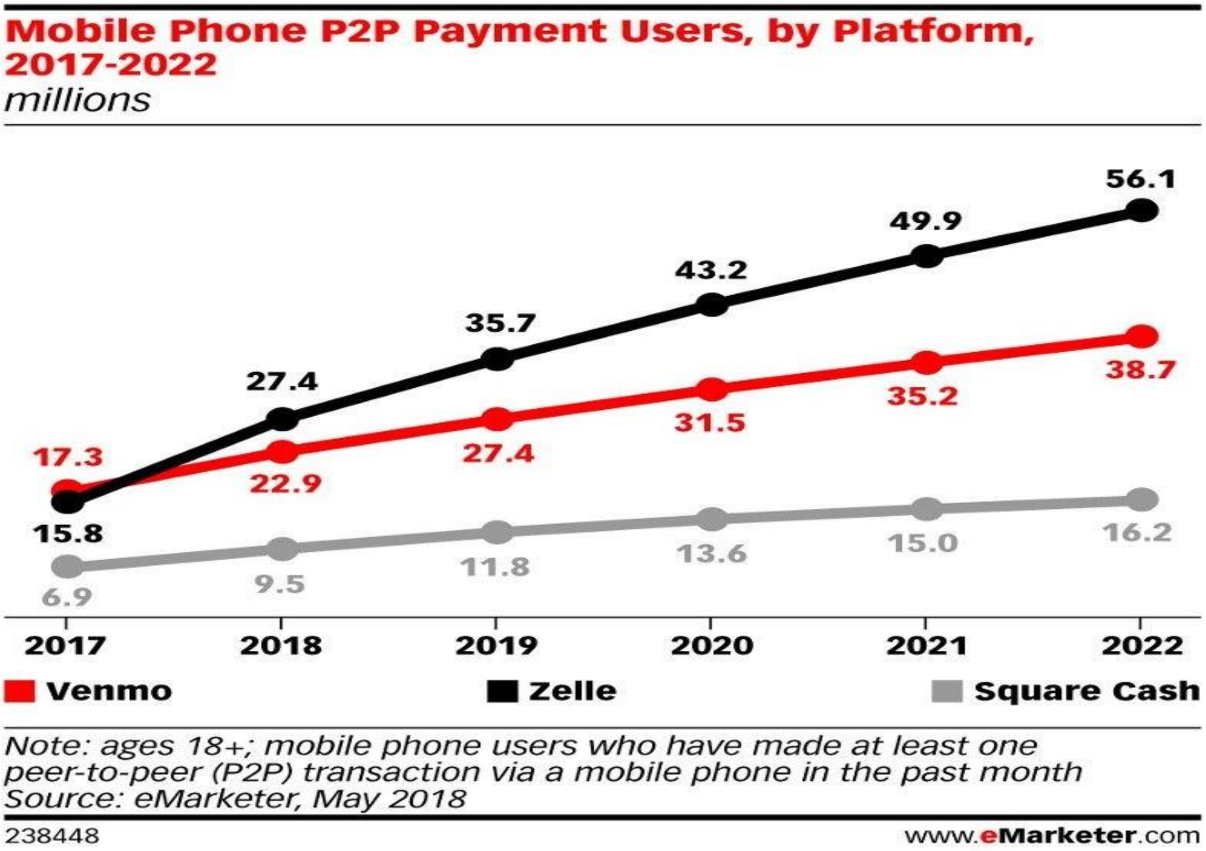
1.1 Introduction

“Mobile Payments (which encompass mobile wallets and mobile transfers) are regulated transactions that take place through our mobile device. That is, instead of paying for stuff with cash, cheques, or physical credit cards, mobile payments allows us to do so digitally”. It is the natural successor to electronic commerce. Although the concept of using non-coin-based currency system has a long history, it is only in the 21st century that technology has helped in the upliftment of the system. The capability to pay electronically coupled with a website in the form of an app is the engine behind electronic commerce. A mobile payment can be defined as any payment system where a device is used to initiate, authorize and confirm an exchange of financial value in return for goods and services.These may include PDAs, mobile phones, wireless tablets and any other device that connect to telecommunication network and make it possible for the payments to be made. The evolution of mobile payments will pave new and unforeseen ways of payments and commerce. It can become a complement to cash, cheques, credit and debit cards. The adoption of “digital payment system” is picking up speed. This is where the mobile payment apps come into play. The technology is changing at a great rate and hence the evolution of the digital payments is improving. There are various ways for doing a digital transaction with each having their own requirements. Hence, to do a transaction we have to enter the account number, IFSC Code etc.

This is where QR Codes comes into play. QR Code completely overcomes the issue of entering wrong essentials for doing a complete transaction i.e by scanning a merchant’s QR Code we can get his bank account details and without entering any of the credentials we can do the transaction. One of the key characteristics of using QR Code is that it saves time and eliminates the scenario of entering wrong details. This makes the transaction safe and secure.

1.2 History and Evolution

The technology behind mobile payments continues to improve at a rapid pace.The advance in technology means that the consumer's experience is continuing to be even more convenient, faster and efficient. However, India has a vast non-banking population, most of whom reside in the rural areas.The traditional banking industry finds it difficult to cater to the needs of India's large rural populace. Earlier, cash-less methods of payment evolved, but still required a physical manifestation of money. “Cheques were introduced by the Bank of Hindustan, one of India's first banking institutions, in 1770. The Negotiable Instruments Act of 1881 formalised non-cash modes of paper payments. While internet service and mobile phones came to India in the 1990s, internet usage and smart-phone ownership boomed only in the 21st century”. The rise of e-commerce markets also contributed to its development in the recent years. The progress of Information Technology has enhanced the advancement of payment processing and helped in creating some advanced payment systems.In fact, mobile pay volume is predicted to increase to $503 billion dollars by 2020. Mobile payments have changed the way businesses are thinking about payments processing. On one end, we’re heading towards more secure, authenticated ways to process payments. And on the other, we’re heading towards swift, and convenient ways to pay.



Mobile Phone P2P Payment Users, by Platform, 2017-2022

millions

Note: ages 18+; mobile phone users who have made at least one peer-to-peer (P2P) transaction via a mobile phone in the past month
Source: eMarketer, May 2018

238448 www.eMarketer.com

1.3 Types of Mobile Payment System in India

1.3.1 IMPS

IMPS stands for Immediate Payment Service. This digital payment system was launched by the Reserve Bank of India(RBI) in collaboration with the National Payments Corporation of India in November 2010. It was established keeping in mind the major challenges faced by the banking industry for real time and 24X7X365 interbank transaction. “IMPS provides robust and real time fund transfer which offers an instant, 24X7, interbank electronic fund transfer service that could be accessed on multiple channels like Mobile, Internet, ATM, SMS, Branch and USSD(*99#)”. It allows the transfer of funds instantly within banks across India which is safe as well as economical. “Currently on IMPS, 243 members are live which includes banks and PPIs”.

Objectives of IMPS

- To enable customers of various banks to use digital system as a medium for accessing their bank accounts and transfer funds.
- To sub-serve the goal of “Reserve Bank of India(RBI)” in electronification of person to person payments.
- To enhance and promote mobile payment system and making payment process much simpler.

Participants for IMPS is as listed below

- Remitter or Sender
- Beneficiary or Receiver
- Banks
- National Financial Switch-NPCI

Fund Transfer/Remittance

- Using Mobile Number & MMID(P2P)
- Using Account Number and IFSC Code
- Using Aadhaar Number

Process

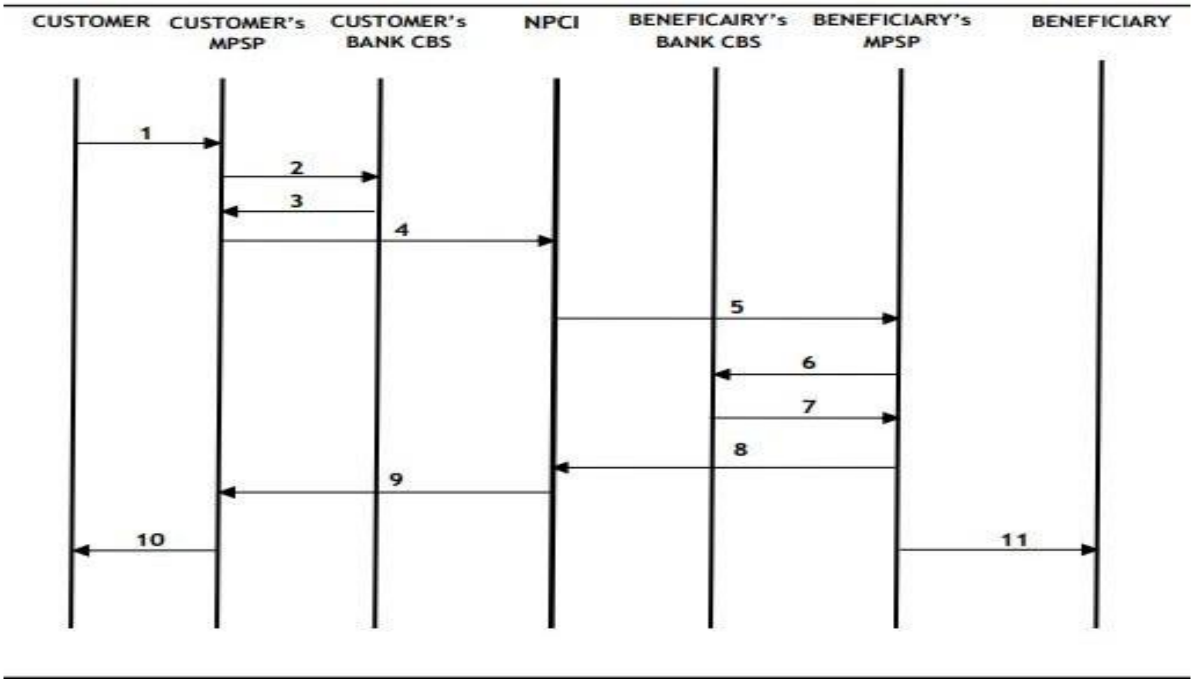


Fig 1.1 Entities involved in an IMPS Transaction

Steps Involved:

1. Customer inputs the beneficiary or receiver's mobile number along with other credentials that are required for the completion of transaction.
2. Customer's MPSP(Mobile Payment Service Provider) receives the request for remittance from customer and the request is then sent to Customer's bank for debiting the account.
3. Customer's bank after receiving the request from the previous entity debits the customer's account and sends confirmation back to the MPSP.
4. After receiving the confirmation MPSP then sends the transaction along with the details to NPCI.
5. After the resolution of destination point, NPCI forwards the transaction to beneficiary's MPSP.
6. Beneficiary's MPSP will forward the transaction to beneficiary's bank for crediting the account.
7. After the beneficiary's account is credited, receiver's bank sends a confirmation to the previous entity.

8. On receiving the confirmation, it is forwarded to NPCI.
9. NPCI then forwards the confirmation to MPSP of the customer.
10. A confirmation message is sent to the customer's bank.
11. A confirmatory SMS is sent to the customer about the completion of the given transaction.
12. Beneficiary's MPSP sends a confirmation SMS to customer stating account being credited with certain amount.

1.3.2 UPI

"UPI stands for Unified Payments Interface. It was launched by the Reserve Bank of India on April 2016 in collaboration with National Payments Corporation of India. It was launched mainly to power multiple bank accounts into a single mobile application, merging several bank features, seamless fund routing and merchant payments into one hood. It is built over the IMPS infrastructure and allows transfer of money between two parties bank accounts".

Objectives of UPI

- To facilitate immediate money transfer through electronic device round the clock 24*7 and 365 days.
- Single mobile application for accessing different bank accounts and doing the inter-bank transaction.
- Merchant Payments with Single Application or In-App Payments.
  .

Participants in UPI is as listed below

- Payer PSP
- Payee PSP
- Remitter Bank
- Beneficiary Bank
- NPCI
- Bank Account Holders
- Merchants

Process

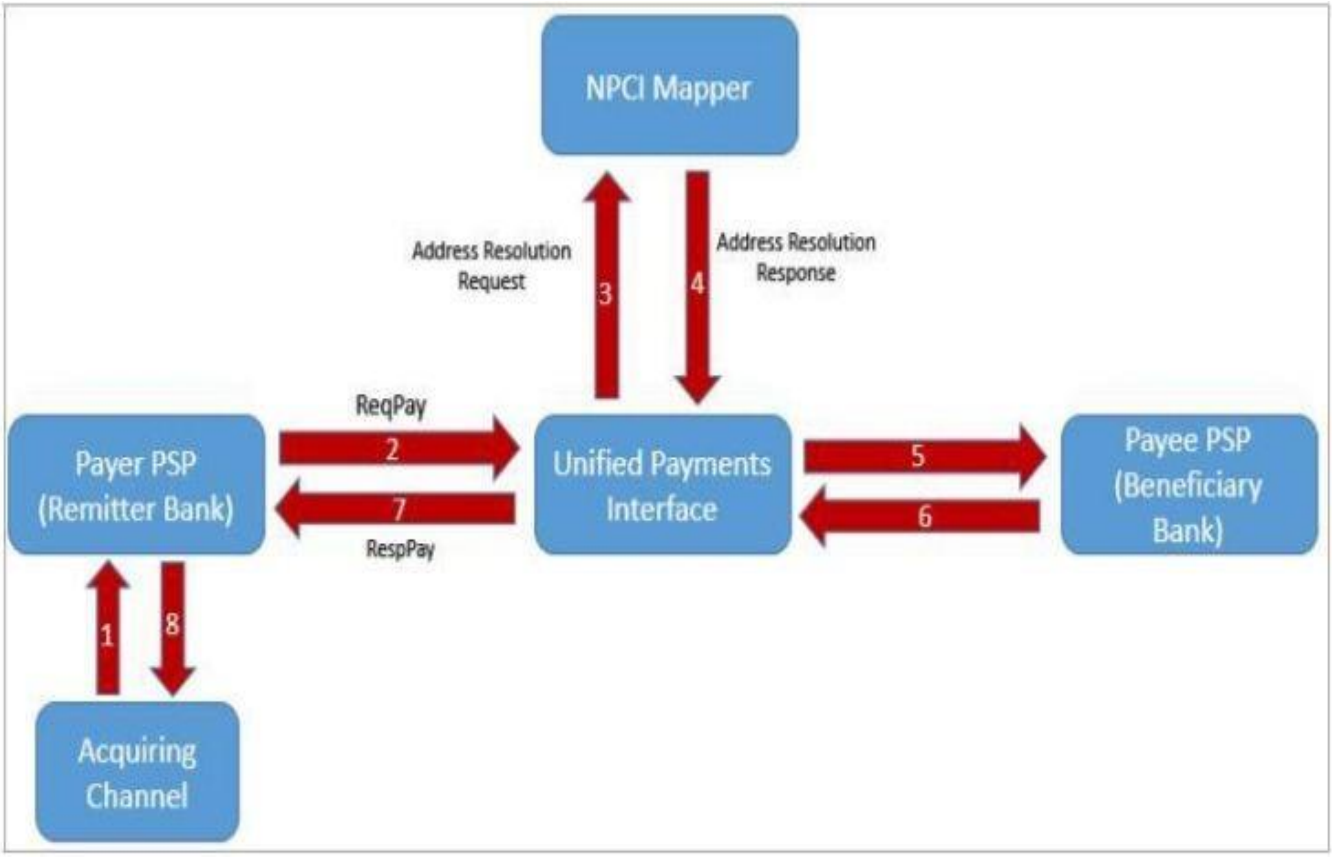

NPCI Mapper
Address Resolution Request
3
4
Address Resolution Response
ReqPay
2
Payer PSP (Remitter Bank)
Unified Payments Interface
5
Payee PSP (Beneficiary Bank)
7
RespPay
6
1
8
Acquiring Channel

Fig 1.2 Entities involved in UPI Transaction

Steps Involved:

1. The UPI Id of the Payee is entered in the app and the payment is authorized by entering the PIN.
2. Sender's bank then debits the sender's account and the transaction is sent to UPI along with the necessary details.
3. A query is sent to the NPCI Mapper for fetching the details of the payee for address transaction.
4. After that Mapper responds with the relevant Account information associated wit the transaction.
5. A credit request is initiated to the Receiver's bank by UPI.
6. Beneficiary's bank credits the payee account and responds back to UPI.
7. UPI forwards the message to the payer PSP for successful transaction.
8. Payer PSP confirms the same to the customer.

1.3.3 AePS

“AePS stand for Aadhaar Enabled Payment System. It is a bank led model which allows online interoperable financial transaction at the PoS(Point of Sale/ Micro ATM) through the Business Correspondent(BC)/ Bank Mitra of any bank using the Aadhaar authentication. The main background to design this model was to further speed track financial inclusion in the country. Two working groups were constituted by RBI with members representing RBI, Unique Identification Authority of India, NPCI, Institute for Development and Research in Banking Technology and some special invitees representing banks and research institutions”. The only inputs required for a customer to do a transaction under this scenario are:-

- IIN of the entities
- Aadhaar Number of both the entities
- Fingerprint of the entities captured during their enrollment.

Objectives of AePS

- To sub-serve the goal of Government of India (GoI) and Reserve Bank of India (RBI) in furthering financial institution and electronification of payments.
- To enhance inter- operability across banks in a safe and secure manner.

Steps Involved:

1. The Aadhaar Number and Location Code of the Payee is entered and the payment is authorized by entering the PIN.
2. Remitter bank then debits the customer account and the transaction is sent to AePS along with the payer account credentials and global address of the payee.
3. AePS sends a query to the NPCI Mapper and UIDAI for the identification of the payer.
4. UIDAI then responds whether the payer is authenticated or not.
5. NPCI Mapper gives the relevant account information of the entities associated with the transaction.
6. A credit request is initiated to the Beneficiary bank.
7. The payee account is credited and NPCI is given the response.
8. The payer is responded by NPCI for the successful transaction.

Process

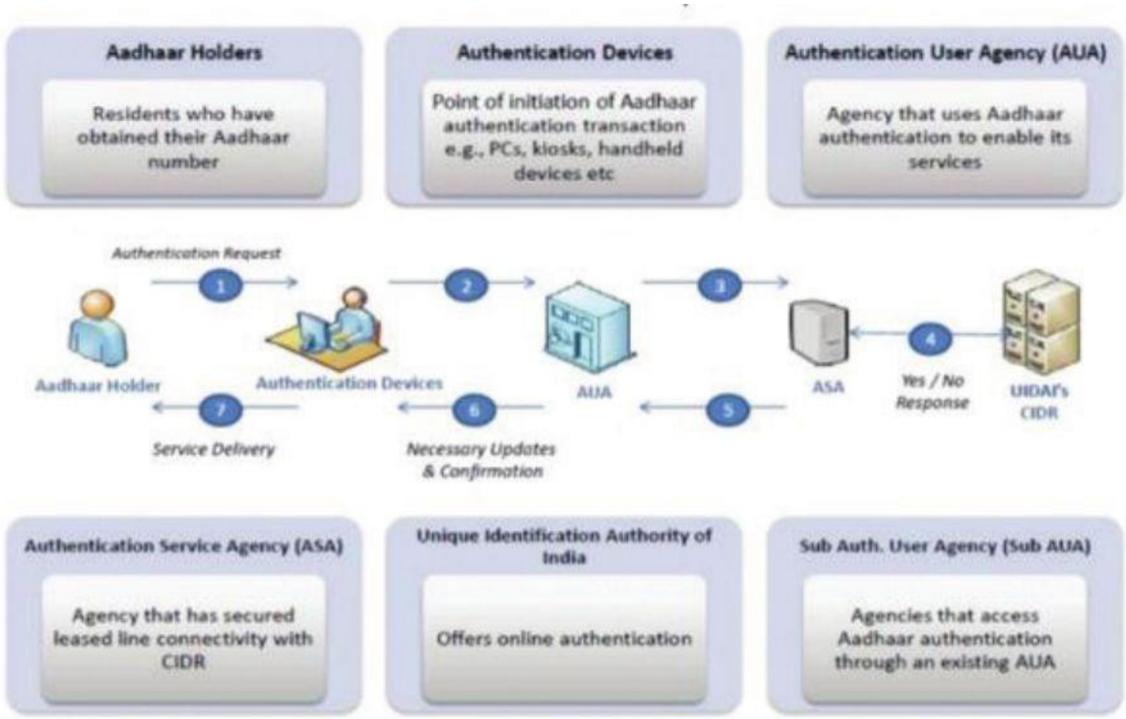

Aadhaar Holders
Residents who have obtained their Aadhaar number
Authentication Devices
Point of initiation of Aadhaar authentication transaction e.g., PCs, kiosks, handheld devices etc
Authentication User Agency (AUA)
Agency that uses Aadhaar authentication to enable its services
Authentication Request
1
2
3
4
5
6
7
Aadhaar Holder
Authentication Devices
AUA
ASA
Yes / No Response
UIDAI's CIDR
Service Delivery
Necessary Updates & Confirmation
Authentication Service Agency (ASA)
Agency that has secured leased line connectivity with CIDR
Unique Identification Authority of India
Offers online authentication
Sub Auth. User Agency (Sub AUA)
Agencies that access Aadhaar authentication through an existing AUA

Fig 1.3 Entities involved in AePS Authentication

1.3.4 NEFT

“NEFT stands for National Electronic Funds Transfer. It is an electronic funds transfer system that is maintained by the Reserve Bank of India. It was started in November 2005. It enables bank customers in India to transfer funds between any two NEFT-enabled bank accounts on a one-to-one basis. Unlike RTGS, funds are transferred in half-hourly batches and not in real time basis.”

Objectives of NEFT

- To have a low cost fund transfer system and an almost instant system.
- To have a flexible payment system such that one can schedule a payment as well for online transfer of funds.
- For the electronification of the payment system.

Participants for NEFT will be as follows

- Sender or Remitter
- Beneficiary or Receiver
- Sender Bank
- Receiver Bank
- Pooling Center
- Clearing Center

Process

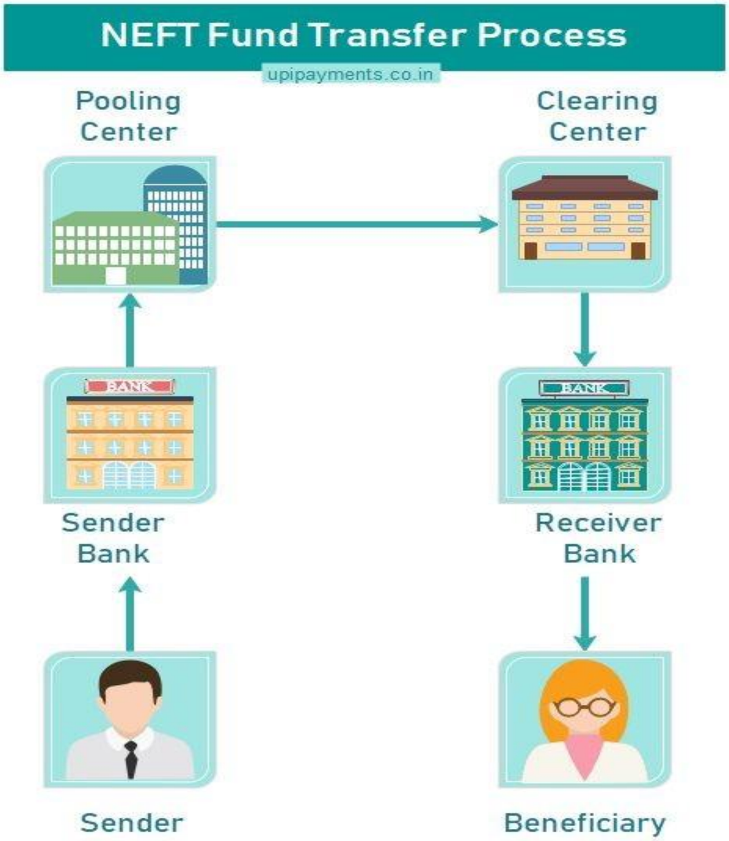


Fig 1.4 Entities involved in NEFT Transaction

Steps Involved:

1. The customer fills an application form to provide all necessary credentials of the beneficiary ("like name, bank, branch name, IFSC, account type and account number")

2. A message on being prepared by the originating bank is sent to the pooling centre (“also called NEFT Service Centre”)
3. The pooling centre after receiving the message then forwards the message to the NEFT Clearing Centre (“operated by RBI and National Clearing Cell”) to be included for the next available batch.
4. The Clearing Centre sorts the funds transfer transactions destination bank-wise and prepares accounting entries to receive funds from the given originating banks and transfer the funds to the destination banks.
5. The destination bank on receiving the inward remittance messages from the Clearing Centre passes the credit to the customer’s accounts.

1.4 QR Codes

1.4.1 Introduction

“QR Code stands for Quick Response Code. It is a two-dimensional square barcode which can store encoded data. It was created in 1994 by Denso Wave, a Japanese subsidiary in the Toyota Group for the automotive industry. It became popular due to its fast readability and greater storage capacity compared to standard UPC barcodes. Mostly, it consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as camera etc”. It has found its way into mobile marketing with the widespread adoption in the mobile payment system.

1.4.2 Specifications

“There are about 40 versions of the QR Code. Each version has a different module configuration or number of modules (The module refers to the black and white dots that make up the QR Code). Module Configuration refers to the number of modules contained in a symbol, commencing from version 1(21*21 modules) to version 40(177*177 modules). Each higher version of QR Code comprises 4 additional modules per side. The size of the QR Code is determined by the version and size of the module. The actual size is determined by multiplying the number of modules by the printable size of the module and margins are added to determine the required space for the QR Code. According to the encoding standard of QR code symbols, after masking processing, the proportion of dark module to light module is 1:1. This means that when QR Code (except for blank spaces) occupies the whole image then the number of black squares and white squares that form module are equal in number or in equal proportion”.

Smallest Symbol Size= 21x21 modules
Largest Symbol Size= 177x177 modules

Maximum Data Capacity

- Numeric = 7089 characters
- Alphanumeric = 4296 characters
- Binary = 2953 characters

1.4.3 Scanning Process

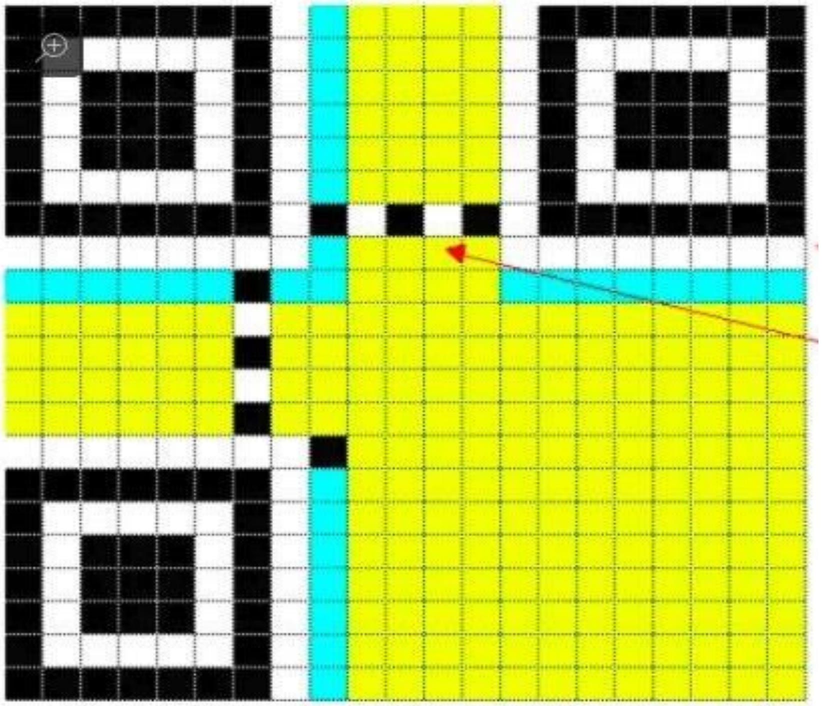


Fig 1.5 QR Code Scanning Image

The smart phone’s image sensor reads the block of smaller black and white squares, which is then interpreted by the system processor. The three large squares act as alignment targets, while the smaller square in the remaining corner acts to normalize the size and angle of the shot. The strips near the alignment squares contain formatting information and the remaining yellow area is the actual data that’s converted into binary codes and are checked for errors before being displayed. An empty white border that makes it possible to isolate the code from among other printed information.

1.4.4 Error Tolerance Pattern of QR Codes

“QR Code have an error correction capability to restore the data if the code is dirty or damaged. It depends on the amount of data to be corrected. There are four error correction levels that are available for users to choose according to the operating environment. The higher the error correction level, the less is the storage capacity since raising this level also increases the amount of data QR Code size”.
For Ex: “If there are 100 codewords of QR Code that is to be encoded, 50 of which needs to be corrected, 100 words of Reed Solomon Code is required as Reed Solomon Code is twice the amount of codewords to be corrected”.

Total number of codewords = 200
Words need to be corrected = 50
Error correction level =(50/200) x 100
= 25%

1.4.5 Shapes of the QR Code

The QR Codes generally have square shape as it follows a standard and is governed by rules and specifications. The eyes or position makers or the square within squares at three corners help the QR Code scanners to determine the orientation of the QR Code and these corners cannot be changed. QR Code also needs to have a quiet zone i.e whitespace around the QR Code to help the scanners identify the eyes. So even if we try to transform the square shape into circular design or any other design by adding black and white blocks(called modules) around the square, it won’t work. The width of the quiet zone needs to be at least 4 modules.

iQR Code: “iQR Code is a matrix type 2D code that allows easy reading of its position and size. iQR Code was designed to allow a wide size range of codes from ones smaller than the traditional QR Code to large ones that can store more data than these. It can be printed as a rectangular code, turned-over code, black-and-white inversion code or dot pattern code allowing a wide range of applications in various areas. With iQR there is higher restoration capability than with traditional QR Code.”

Features:

1. High rate of Information packing
2. Reduced size
3. Use of rectangular modules
4. High data capacity
5. High restoration capability

1.4.6 International Development

The usage of QR Codes is increasing at the international markets. “In Japan, the number of restaurants, convenience stores and other shops accepting payments with smartphones using QR Code is rising”. Major IT firms are working in cohesion to promote digital payment system and banks are also joining this drive. “In Japan, the payments are carried out through various apps such as Rakuten Pay, Line Pay operated by Rakuten Inc. or Line Corp”. “QR Code payments have seen a huge rise in China, which is the world’s largest mobile payment market with an estimated $5.5 trillion in mobile payments in 2018”. According to Chinese news reports the country had nearly 520 million users in 2018, about a third of China’s population, with 82% of transaction being executed using mobile devices. QR Codes became hugely popular as a part of the digital payment method in some european countries as well although they have not widely been used in the United States. Hence, with the evolution of technologies there is a broadening industry focus on enhancing and managing the use of QR codes in payment services.

1.4.7 Bharat QR Codes

“Bharat QR Code is a person to merchant(P2M) mobile payment solution. Once they are deployed on merchant’s locations, users can pay the utility bills without sharing their bank credentials with the merchant. NPCI jointly worked with International card Schemes to develop a common standard QR Code specifications. Bharat QR Codes contain lots of additional information unlike other QR Codes like merchant name, merchant address, merchant bank information etc. It is much more secure and more widely acceptable in comparison to other QR Codes. It works as an alternate channel of payment, where a user has to download his/her bank’s Bharat QR enabled mobile banking app. User on scanning the Bharat QR Code will get the beneficiary information and can select the card to make the payment”.

Advantages:

1. Open Payment Interface
2. Integration of UPI
3. Lower Transaction cost of Bharat QR
4. Dynamism in making payments

Types of Bharat QR Codes:

1. **Static Bharat QR Code**: In order to send money using Static Bharat QR Code, a sender has to scan that QR Code using his/her bank app or UPI app QR Code scanner. Then the amount of money is entered which the user has to send.

2. **Dynamic Bharat QR Code**: In this system the receiver generates the QR Code for a specific amount. That means the sender can’t enter his/her desired amount to transfer. The amount will be fixed for each Dynamic QR Code to sell fixed rate products.

1.5 API

“API stands for Application Programming Interface”. It helps a user’s product or service to communicate with other product or service like dynamically posting content from one application to another, extracting data from a database than a regular UI might allow. Basically it is a part of the server that receives requests and send responses.

Types of APIs:

1. Web API: A Web API is an extensively used API and can be easily accessed using the HTTP requests. Being an open source interface it can be used by large number of clients through their phones, tablets or PCs.
2. Local API: This type of APIs offer OS or middleware services to application programs. TAPI(Telephony Application Programming Interface) and .NET are the common example of APIs.
3. Remote API: It makes a remote program appear to be local by making use of RPC(Remote Procedure Calls)

Data Format:

1. XML: “XML stands for eXtensible Markup Language. It is a software and hardware independent data format for storing and transforming the data. Although, it is a markup language like HTML yet it doesn’t have predefined tags. It also defines a rule for encoding documents in a format that is both machine readable and human readable”.

2. JSON: “JSON stands for Javascript Object Notation. It is a lightweight data-interchange format and is completely language independent. Based on Javascript programming language, it is often used when data is sent from a server to a webpage”.

HTTP Methods:

1. GET: The HTTP GET method is used to request a representation of the specified resource. Requests using GET is used only to retrieve data.

2. POST: “The HTTP POST method sends data to the server. A POST request is sent via an HTML form and results in a change on the server.”

3. PUT: “The HTTP PUT request method creates a new resource or replaces a representation of the target resource with the target payload.”

4. DELETE: “The HTTP DELETE request method deletes the specified response.”

Chapter 2: Design and Development of

APIs for QR Code based payments

2.1 Introduction

In this chapter we will discuss about the design and the development of APIs for each mode of payments, inputs that are being given to each entity in JSON and the request/response involved in each mode of payments.

2.2 Modes of Digital Payments

A. IMPS:

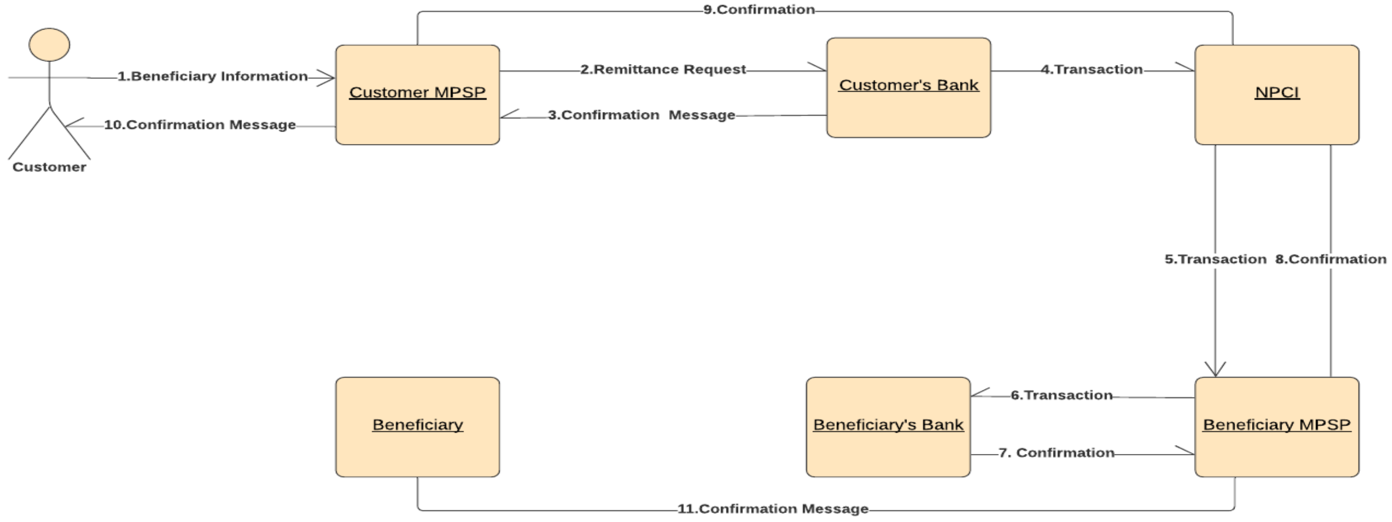


Fig 2.1 Entities in IMPS

The main objective of the API is to carry the request to the next entity and give a specified response. The data format used for transferring the data from one entity to another entity is JSON. The inputs given in each step are being specified below.

1. In this step the user on scanning the QR Code gets the following information which are being used as the input. The information is passed from customer to customer’s MPSP.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    ]}
```

2. The customer MPSP gets the data and initiates a remittance request to customer’s bank and hence the transaction request is forwarded.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    ]}
```

3. The customer’s bank gives the response for the remittance request and debits the transaction from the customer’s bank account and hence a confirmatory message is being forwarded.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    ]}
```

4. The transaction is received by the NPCI and it contains all the information regarding the transaction and hence the API further carries the request to the beneficiary’s entity.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},]}
```

5. The transaction is then forwarded to the beneficiary’s MPSP and the API carries the request to the MPSP and data is transferred in JSON.

```
{"payload": [
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},
    {"amount": "1"},
    ]}
```

6. The transaction is then forwarded to the beneficiary’s bank and credit request is being initiated by the API.

```
{"payload": [
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},
    {"amount": "1"},
    ]}
```

7. A response is carried by the API that the transaction is being done and a confirmatory message is forwarded to the MPSP.

```
{"payload": [
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},
    {"amount": "1"},
    ]}
```

8. Confirmatory Response to the NPCI that the transaction is being completed by the beneficiary’s MPSP.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},
    ]}
```

9. Confirmatory response to the customer about the completion of the transaction.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"}
    {"customer_name": "ABC"},
    {"customer_account": "4589763512"},
    {"customer_ifsc_code": "SBI2569852"},
    {"customer_mobile": "8796563515"},
    ]}
```

Request and Response

1.Successful Transaction

REQUEST

```
import requests
url = "”
payload = {
          'beneficiar_name': 'Ashish T',
          'beneficiary_account_number': '1008012513',
          'beneficiary_ifsc_code': 'ICIC1008567',
          'beneficiary_amount': '1',
          'beneficiary_mobile': '8796565201'
         }

files = {}
headers = {
  'api_key': 'API_KEY'
}

response = requests.request('POST', url, headers = headers, data = payload, files =
files, allow_redirects = false, timeout = undefined)
print(response.text)
```

RESPONSE

200-OK

```
{
  "message": "Transaction Successful",
  "government_commission": 0,
  "other_commision": 0,
  "requestNo": "IMPSI447I166366158967IU",
  "sent_amount": 1,
  "status": "success"
}
```

2. Insufficient Balance in your account

REQUEST:

```
import requests
url = ''
payload = {
          'beneficiar_name': 'Ashish T',
          'beneficiary_account_number': '1008012513',
          'beneficiary_ifsc_code': 'ICIC1008567',
          'beneficiary_amount': '1',
          'beneficiary_mobile': '8796565201'
         }

files = {}
headers = {
  'api_key': 'API_KEY'
}

response = requests.request('POST', url, headers = headers, data = payload, files =
files, allow_redirects = false, timeout = undefined)
print(response.text)
```

RESPONSE

400- Bad Request

```
{
"message": "Sorry! The money you are trying to send is more than the sum of your
balance amount and the charges that we deduct per transaction. Please add more
balance and try again. Thanks!",
"status": "failed"}
```

B. UPI:

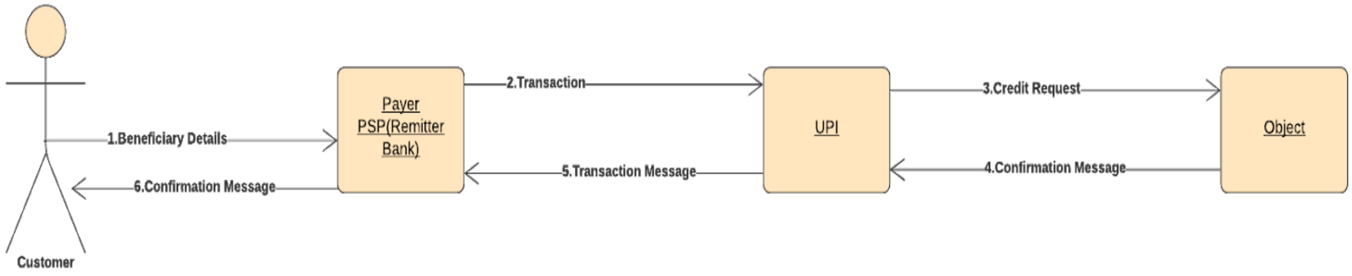


Fig 2.2 Entities in UPI

The main objective of the API is to carry the request to the next entity and give a specified response. The data format used for transferring the data from one entity to another entity is JSON. The inputs given in each step are being specified below.

1. In this step the user on scanning the QR Code gets the necessary credentials of the beneficiary which are then passed as input. The API carries the request to the next entity and the data is transferred to the remitter bank in JSON format.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_virtual_address": "xyz@upi"},
    {"beneficiary_amount": "1"},
    ]}
```

2. The transaction request is then forwarded to the UPI and credit request is initiated and all the credentials of the transaction is then forwarded to the UPI.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_virtual_address": "xyz@upi"},
    {"beneficiary_amount": "1"},
    {"customer_name": "ABC"},
    {"customer_virtual_address": "abc@upi"},
    {"customer_account_number": "1587965423"},
    {"customer_ifsc_code": "SBI5897658"}
    ]}
```

4. The API carries the response of the credit request initiated and a confirmatory message is forwarded regarding the completion of the transaction.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_account_number": "1008565479"},
    {"beneficiary_ifsc_code": "YES1010789"},
    {"beneficiary_amount": "1"},
    {"beneficiary_mobile": "8762516977"},
    {"beneficiary_virtual_address": "xyz@upi"}
    ]}
```

5. The transaction message is forwarded to the PSP regarding the completion of the transaction along with the customer details.

```
{"payload": [
    {"customer_name": "ABC"},
    {"customer_virtual_address": "abc@upi"},
    {"customer_account_number": "1587965423"},
    {"customer_ifsc_code": "SBI5897658"},
    {"amount" : "1"}
    ]}
```

6. The confirmatory response is forwarded to the customer about the completion of the transaction.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_virtual_address": "xyz@upi"},
    {"amount": "1"},
    ]}
```

Request and Response

1. Successful Transaction

REQUEST:

```
import requests
url = ''
payload = {
          'beneficiary_virtual_address': 'ashishsamrat12@oksbi',
          'beneficiary_name': 'Ashish T',
          'beneficiary_amount': '1'
         }

headers = {
  'api_key': 'API_KEY'
}
response = requests.request('POST', url, headers=headers, data=payload,
allow_redirects=False, timeout=undefined)
print(response.text)
```

RESPONSE

200-OK

```
{
 "message": "Transaction Successful",
 "other_commission": 0,
 "order_id": "UPI24456794323457IU",
 "sent_amount": 1,
 "status": "success"
}
```

2. Insufficient Balance in the account

REQUEST

```
import requests
url = ''
payload = {
          'beneficiary_virtual_address': 'ashishsamrat12@oksbi',
           'beneficiary_name': 'Ashish T',
           'beneficiary_amount': '1'
          }

headers = {
  'api_key': 'API_KEY'
}
response = requests.request('POST', url, headers=headers, data=payload,
allow_redirects=False, timeout=undefined)
print(response.text)
```

RESPONSE

```
400- BAD REQUEST
{
 "message": "Sorry! The money you are trying to send plus charges deducted per
transaction has exceeded your account balance. Please add more balance and then try
again. Thank you!"
 "status": "failed"
}
```

C. AePS:

The main objective of the API is to carry the request to the next entity and give a specified response. The data format used for transferring the data from one entity to another entity is JSON. The inputs given in each step are being specified below.

1. In this step the user on scanning the QR Code gets the necessary credentials required for the transaction using AePS mode and then \

```
{"payload": [
    {"pin_number": "1234"},
    {"location_code": "067"},
    {"aadhar_number": "1234567890"}
    ]}
```

2. The transaction request is then forwarded to the AePS and credit request is initiated and all the credentials of the transaction is then forwarded to the AePS.

```
{"payload": [
    {"merchant_aadhar_number": "123456788"},
    {"merchant_location_code": "098"},
    {"amount": "1"}
    ]}
```

3. The aadhar number is then verified by UIDAI and then the response is given back to the NPCI to forward the transaction.

```
{"payload": [
    {"customer_aadhar_number": "1234567890"},
    {"customer_location_code": "067"},
    {"merchant_aadhar_number": "123456788"},
    {"merchant_location_code": "098"}
    ]}
```

4. The transaction is then forwarded to the beneficiary’s bank and credit request is initiated and forwarded to the beneficiary’s MPSP.

```
{"payload":[
     {"customer_aadhar_number": "1234567890"},
     {"customer_location_code": "067"},
     {"customer_account_number": "1579654112"},
     {"customer_ifsc_code": "SBI8796521"},
     {"amount": "1"}
     ]}
```

5. The transaction message is forwarded to the PSP regarding the completion of the transaction along with the customer details.

```
{"payload":[
    {"merchant_aadhar_number": "123456788"},
    {"merchant_location_code": "098"},
    {"merchant_account_number": "123456788"},
    {"merchant_ifsc_code": "YES458796"},
    {"customer_aadhar_number": "1234567890"},
    {"amount": "1"}
    ]}
    200
```

6. The confirmatory response is forwarded to the customer about the completion of the transaction.

```
{"payload": [
    {"merchant_aadhar_number": "123456788"},
    {"merchant_location_code": "098"}
    ]}
```

Request and Response

1. Successful Transaction

REQUEST

```
import requests
url = ''
payload = {
        'beneficiary_aadhar_number': '1203456954',
        'beneficiary_location_code': '0647',
        'beneficiary_amount': '1'
        }

headers = {
  'api_key' = 'API_KEY'
}

response = requests.request('POST', url, headers=headers, data=payload,
allow_redirects=False, timeout=undefined)
print(response.text)
```

RESPONSE

```
200-OK
{
 "message": "successful transaction",
 "government_commission":0,
 "other_commission":0,
 "requestNo.": "AEPS15487965315662",
 "amount": 1
}
```

2. Insufficient Balance in the account

REQUEST

```
import requests
url = ''
payload = {
          'beneficiary_aadhar_number': '1203456954',
          'beneficiary_location_code': '0647',
          'beneficiary_amount': '1'
          }

headers = {
   'api_key' = 'API_KEY'
}

response = requests.request('POST', url, headers=headers, data=payload,
allow_redirects=False, timeout=undefined)
print(response.text)
```

RESPONSE:

```
400- BAD REQUEST
{
  "message": "Sorry! The money you are trying to send plus charges deducted per
transaction has exceeded your account balance. Please add more balance and then try
again. Thank you!"
 "status": "failed"
}
```

D. NEFT:

The main objective of the API is to carry the request to the next entity and give a specified response. The data format used for transferring the data from one entity to another entity is JSON. The inputs given in each step are being specified below.

1. In this step the user on scanning the QR Code gets the necessary credentials required for the transaction using NEFT mode.

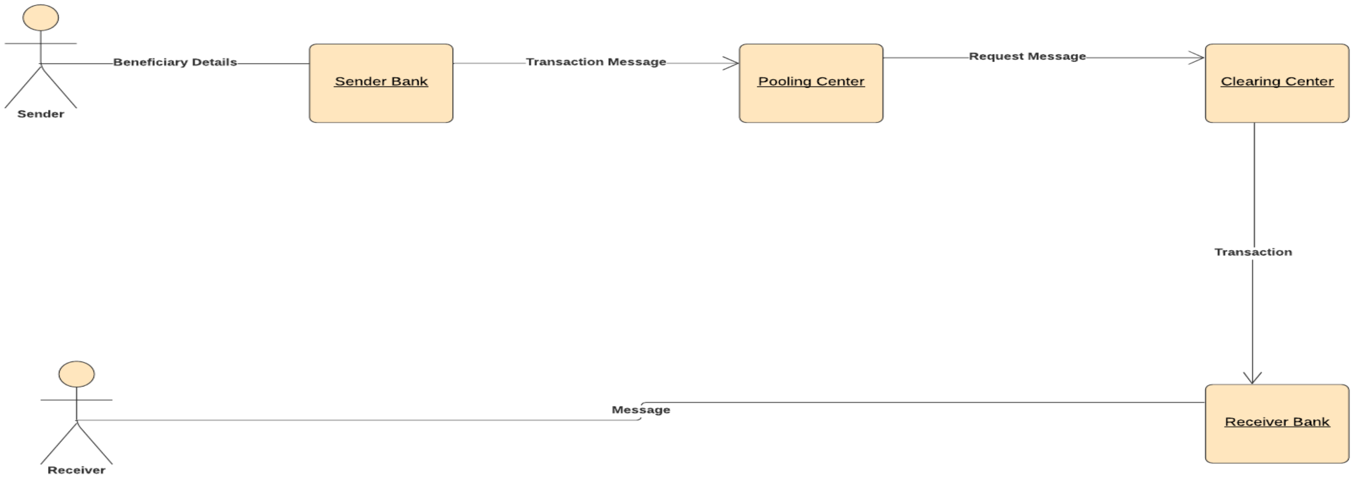
```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"beneficiary_bank": "SBI"},
    {"IFSC_Code": "SBI489958992212"},
    {"Account_number": "254814395101"},
    {"beneficiary_branch_name": "abcd SBI"},
    {"amount": "1"}
    ]}
```

2. The transaction request is then forwarded to the Pooling Center and credit request is initiated and all the credentials of the transaction is then forwarded to the Pooling center.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"IFSC_Code": "SBI489958992212"},
    {"Account_Number": "254814395101"},
    {"beneficiary_bank": "SBI"},
    {"amount": "1"}
    ]}
```

3. The transaction is then forwarded to the Clearing center and the credit request is then initiated for the transaction to be completed.

```
{"payload": [
    {"beneficiary_name": "XYZ"},
    {"IFSC_Code": "SBI489958992212"},
    {"Account_Number": "254814395101"},
    {"beneficiary_bank": "SBI"},
    {"amount": "1"}
    ]}
```



2.3 Conclusion

In this chapter we have designed the APIs for each mode of transaction and the data format in which data is transferred between different entities. The request and response in different scenarios for each mode of payments with appropriate messages was also designed.

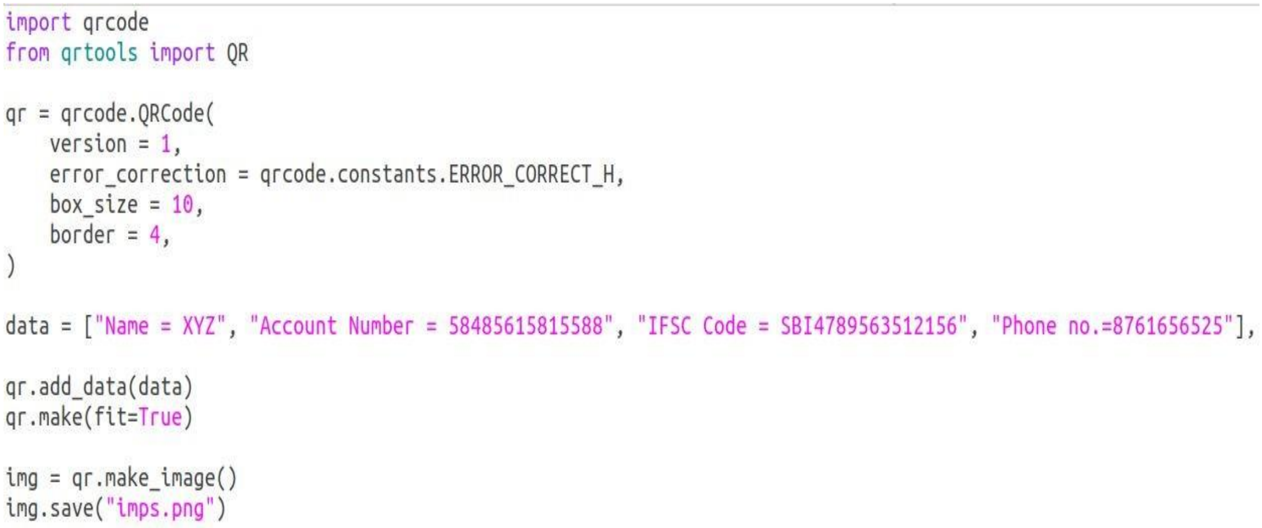
Chapter 3: Code and Prototype Implementation

3.1 Introduction

In this chapter we will discuss about the codes required for the generation of QR Codes for various modes of payments. We will also combine the QR Codes of different payment modes into a single QR Code so that the credentials required for multiple transaction can be selected at once.

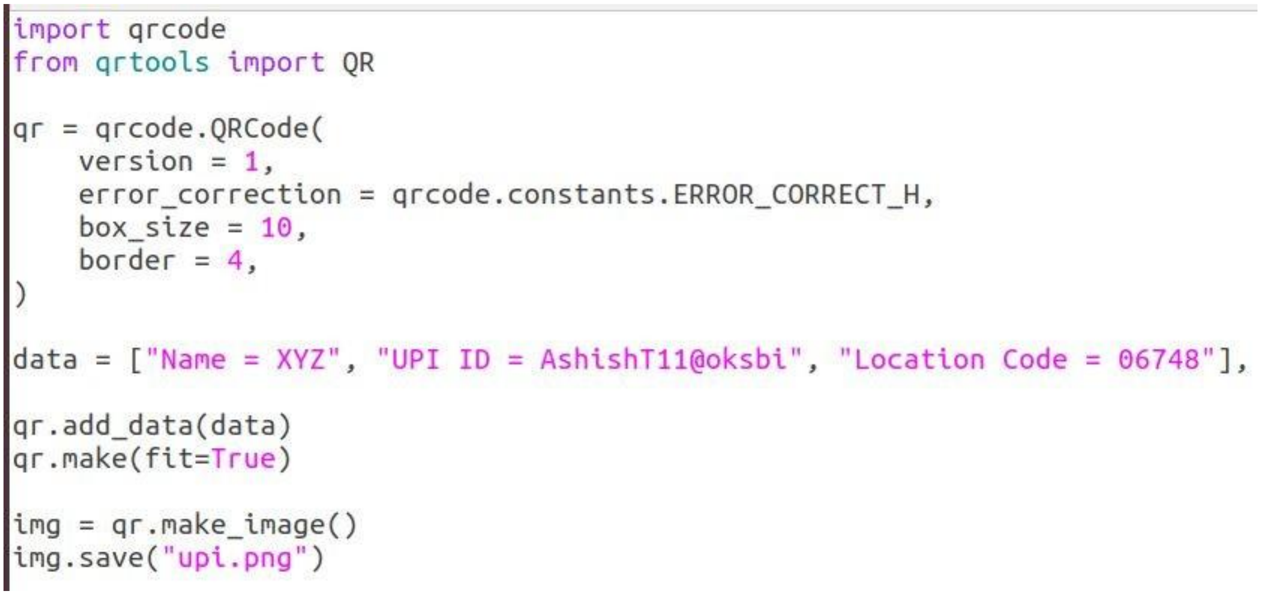
3.2 Codes for generating QR Codes for various payment modes

IMPS:



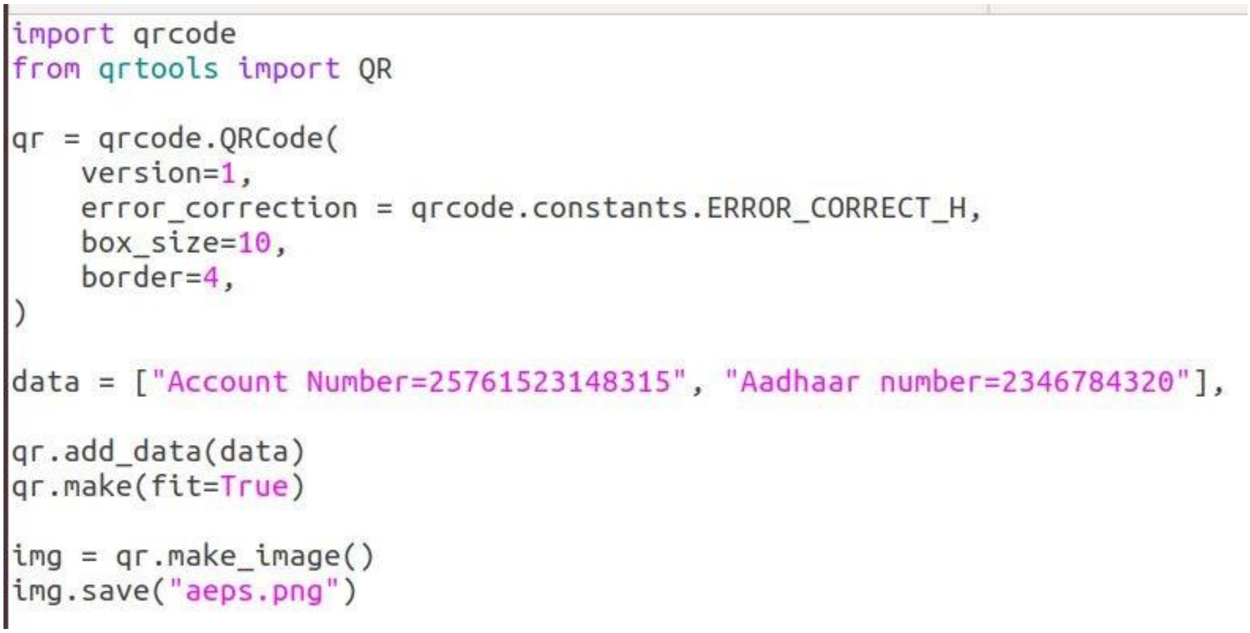
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "Account Number = 58485615815588", "IFSC Code = SBI4789563512156", "Phone no.=8761656525"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("imps.png")
```

UPI:



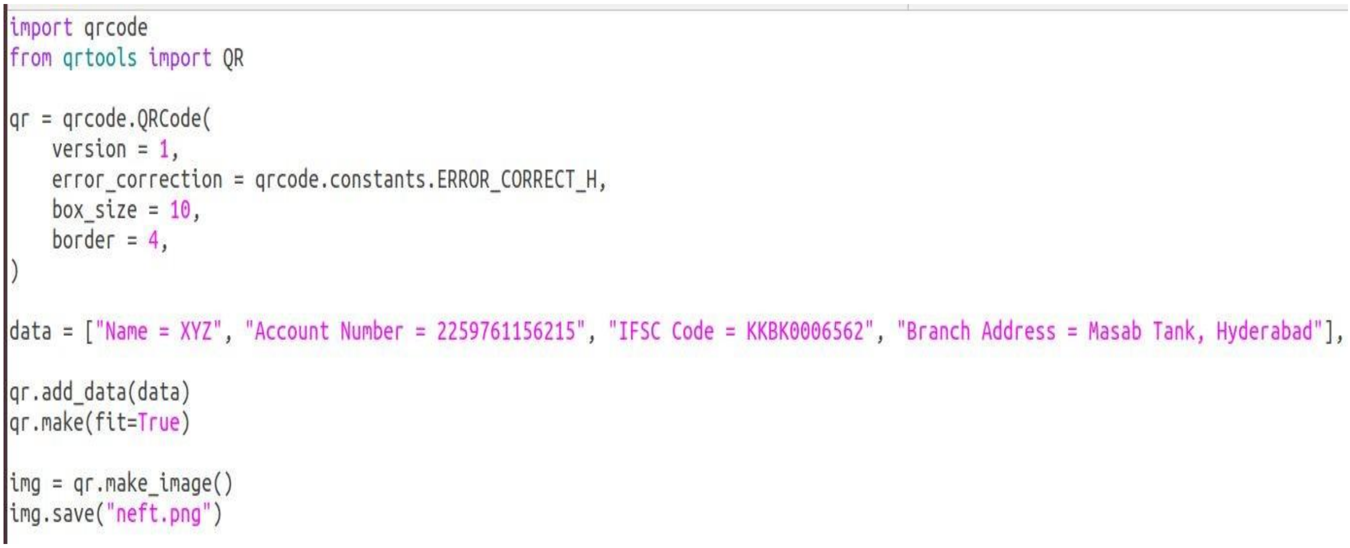
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "UPI ID = AshishT11@oksbi", "Location Code = 06748"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("upi.png")
```

AePS:



```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version=1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size=10,
    border=4,
)

data = ["Account Number=25761523148315", "Aadhaar number=2346784320"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("aeps.png")
```

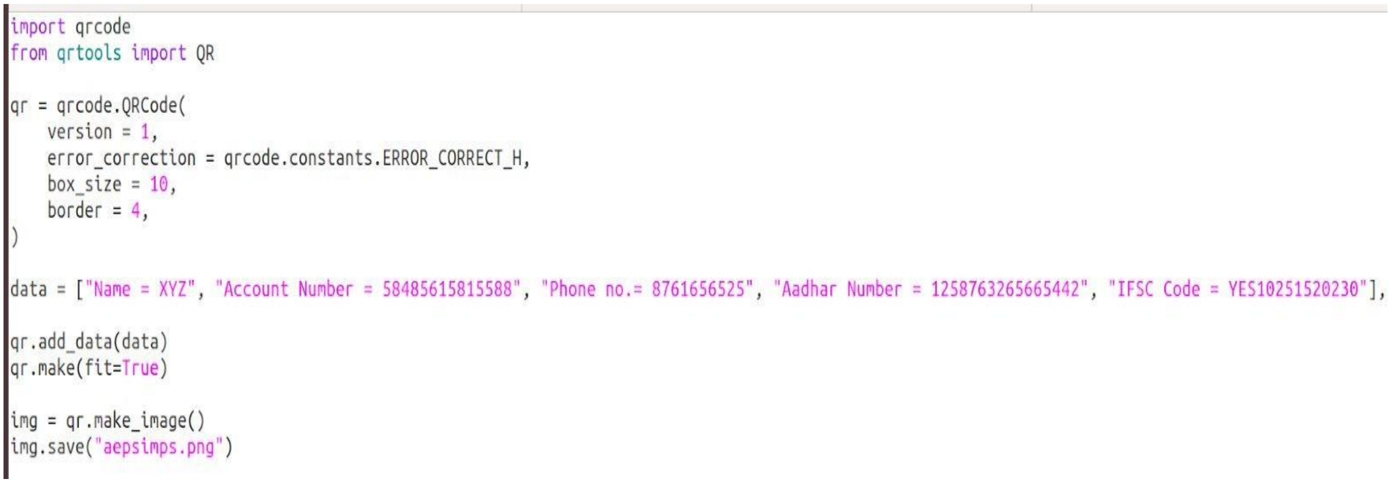
NEFT:



```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "Account Number = 2259761156215", "IFSC Code = KKBK0006562", "Branch Address = Masab Tank, Hyderabad"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("neft.png")
```

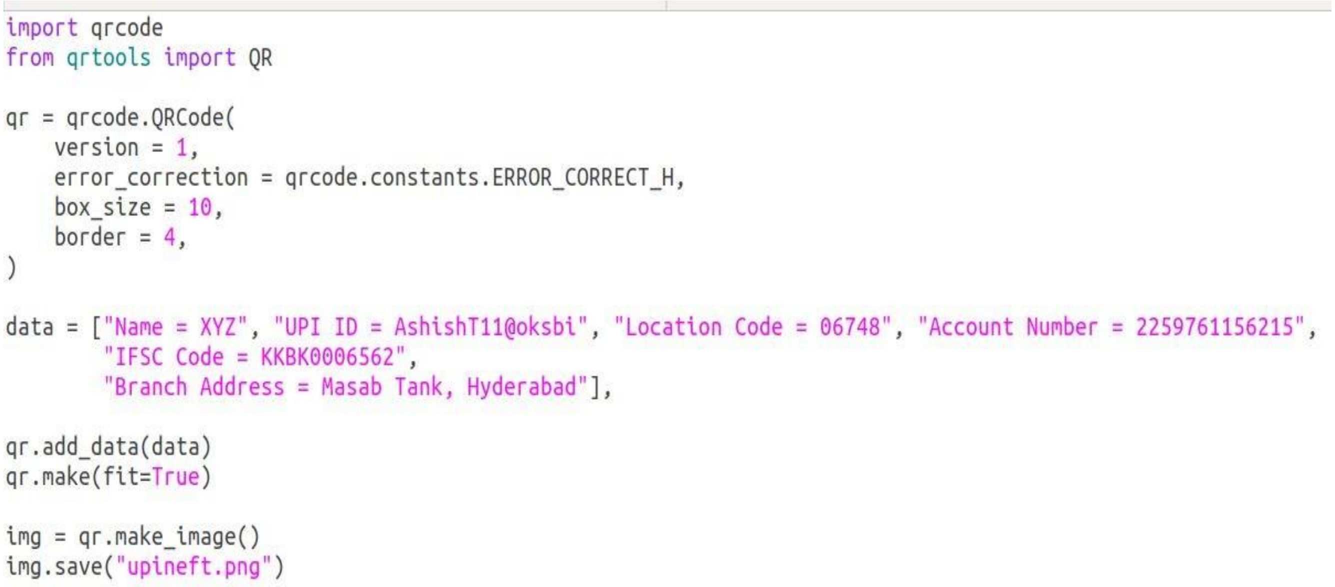
3.3 Combining various Payment System

IMPS With AePS:



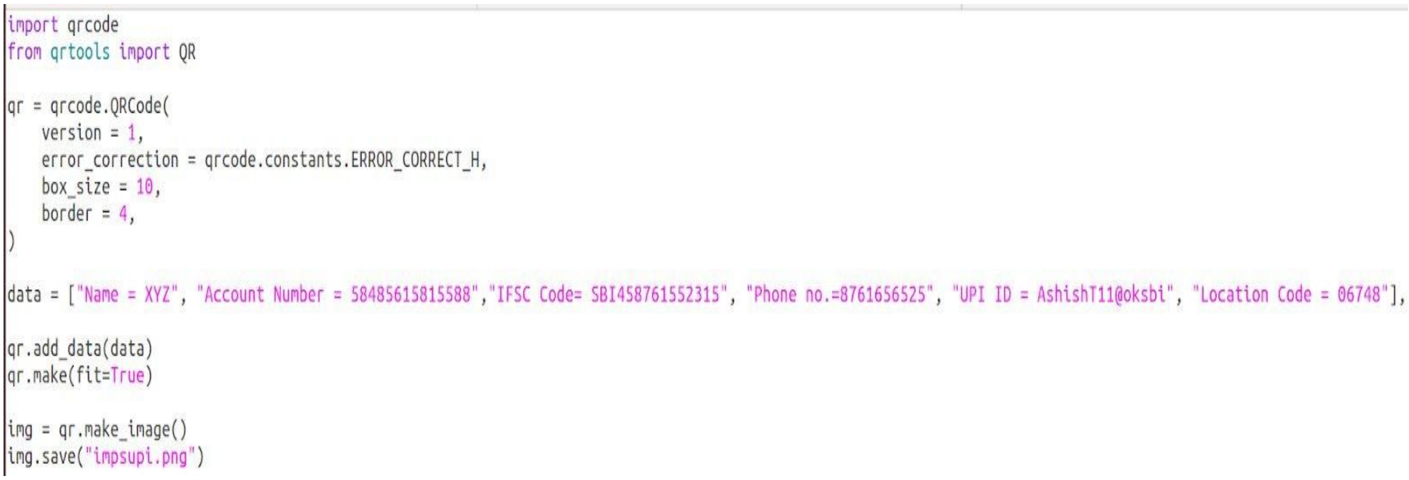
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "Account Number = 58485615815588", "Phone no.= 8761656525", "Aadhar Number = 1258763265665442", "IFSC Code = YES10251520230"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("aepsimps.png")
```

UPI With NEFT:



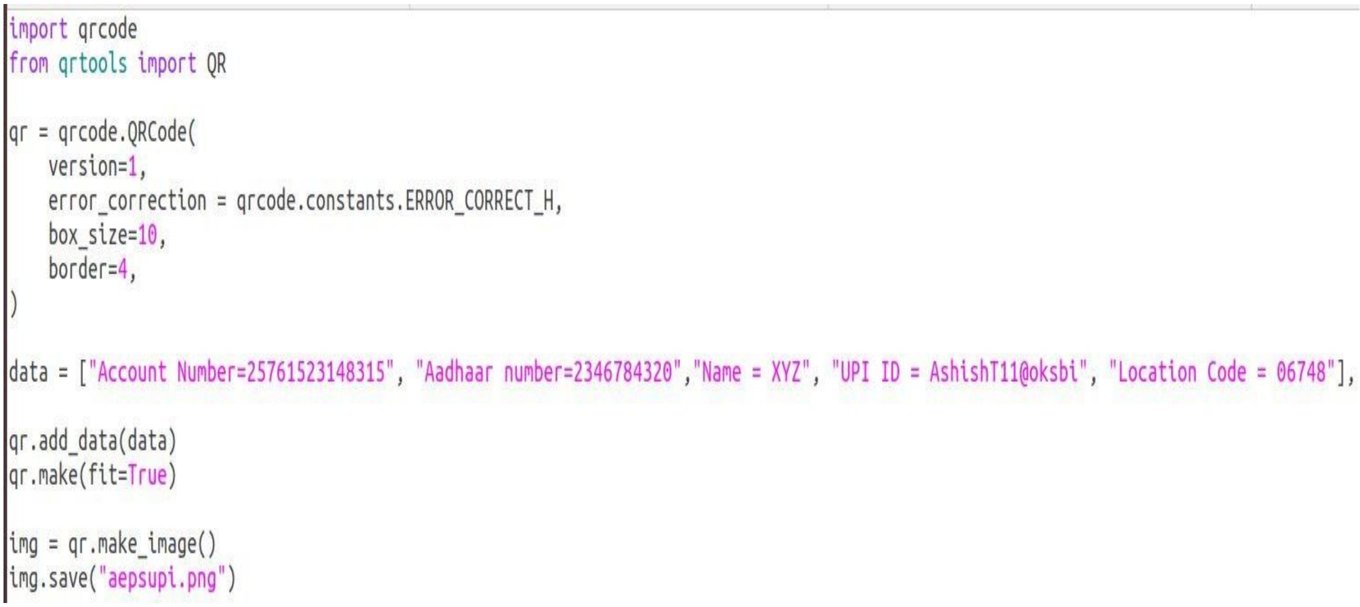
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "UPI ID = AshishT11@oksbi", "Location Code = 06748", "Account Number = 2259761156215",
        "IFSC Code = KKBK0006562",
        "Branch Address = Masab Tank, Hyderabad"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("upineft.png")
```

IMPS With UPI:



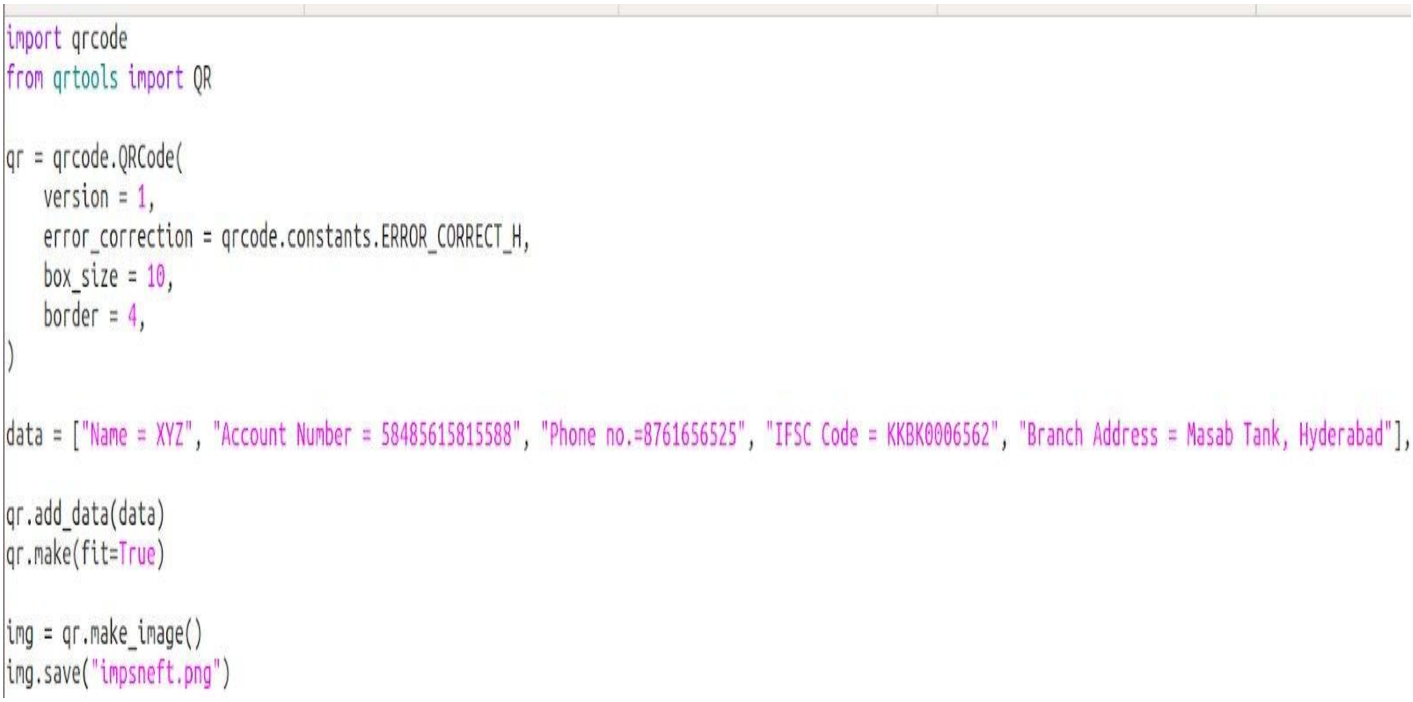
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "Account Number = 58485615815588","IFSC Code= SBI458761552315", "Phone no.=8761656525", "UPI ID = AshishT11@oksbi", "Location Code = 06748"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("impsupi.png")
```

AePS With UPI:



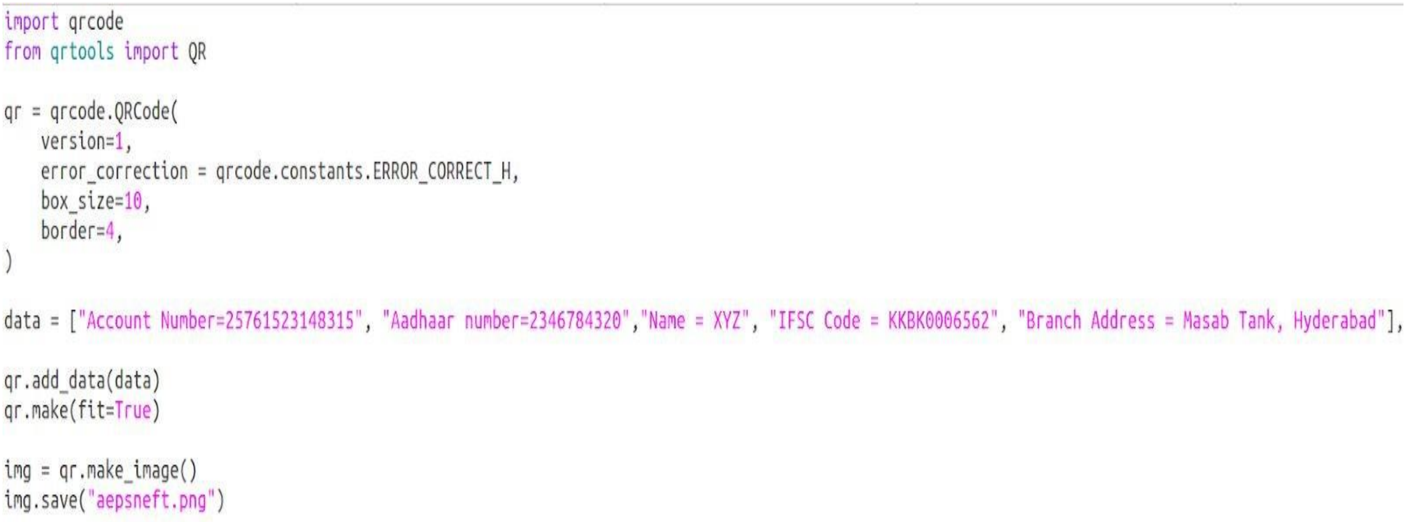
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version=1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size=10,
    border=4,
)

data = ["Account Number=25761523148315", "Aadhaar number=2346784320","Name = XYZ", "UPI ID = AshishT11@oksbi", "Location Code = 06748"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("aepsupi.png")
```

IMPS With NEFT:



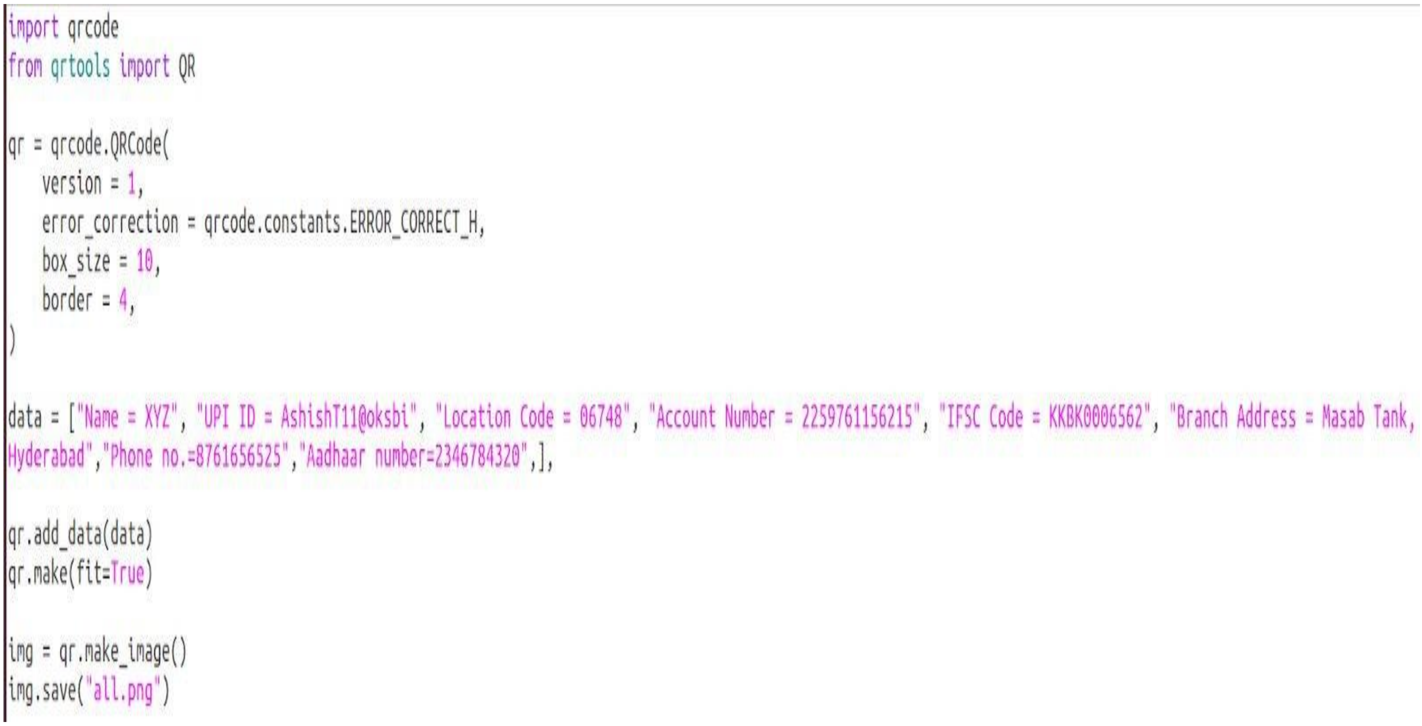
```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "Account Number = 58485615815588", "Phone no.=8761656525", "IFSC Code = KKBK0006562", "Branch Address = Masab Tank, Hyderabad"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("impsneft.png")
```

AePS With NEFT:



```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version=1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size=10,
    border=4,
)

data = ["Account Number=25761523148315", "Aadhaar number=2346784320","Name = XYZ", "IFSC Code = KKBK0006562", "Branch Address = Masab Tank, Hyderabad"],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("aepsneft.png")
```

Single QR Code for all payments:



```
import qrcode
from qrtools import QR

qr = qrcode.QRCode(
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 10,
    border = 4,
)

data = ["Name = XYZ", "UPI ID = AshishT11@oksbi", "Location Code = 06748", "Account Number = 2259761156215", "IFSC Code = KKBK0006562", "Branch Address = Masab Tank,
Hyderabad","Phone no.=8761656525","Aadhaar number=2346784320",],

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image()
img.save("all.png")
```

Chapter 4: Conclusion

This project report consists of three chapters. In Chapter 1 we have discussed about the history and the evolution of the mobile payments and about the different entities involved in each type of payment system. We also discussed about the QR Codes its specification, scanning process, shapes and Bharat QR Codes.

In Chapter 2 we have discussed about the design and development of APIs for different payment system. The entities involved, the request and the response of each payment mode.

In Chapter 3 we have the codes that are used for designing the QR Codes for various payment system and combining the QR Codes for different payment system.

Future Scope:

The result obtained from this report can be used in the development of a payment app which can use QR Codes as a medium to initiate the transaction. User on scanning the QR Code can choose any payment mode to do the transaction and proceed with it. All these features can be combined in a single app and can be released for the users. To generate the QR Code in Android we can use the Zxing library and hence proceed with further development.

Chapter 5: References

1. https://www.npci.org.in/

2.https://github.com/mypoolin/BharatQR

3.https://github.com/mypoolin/mypoolin-server

4.https://github.com/lord/slate

5.https://www.unitag.io/qrcode

6.https://www.qrcode.com/en/codes/iqr.html

7.https://rbi.org.in/

8. https://apidocs.imgur.com