A

Project - I Report

On

POORNIMA PAYMENTS

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Information Technology

(Session 2018-2019)

Guided By -
Mr. SHIRISH NAGAR
Asst. Professor, Deptt. of IT

Submitted by -
Aditya Laxkar (PCE15IT003)
Tushar Gupta (PCE15IT061)
Semester - VII

DEPARTMENT OF INFORMATION TECHNOLOGY
POORNIMA COLLEGE OF ENGINEERING, JAIPUR
RAJASTHAN TECHNICAL UNIVERSITY, KOTA

October- 2018

Candidate's Declaration

I/We hereby declare that the work, which is being presented in the **Project - I Report**, titled **POORNIMA PAYMENT** in partial fulfillment for the award of degree of **Bachelor of Technology** in **Information Technology**, and submitted to the Department of **Information Technology, Poornima College of Engineering**, **Jaipur** is a record of my own work/investigations carried under the guidance of **Mr. Shirish Nagar**, **Asst. Professor Information Technology, Poornima College of Engineering.**

I/We have not submitted the matter presented in this Project-I Report anywhere for the award of any other Degree.

(1) Aditya Laxkar (PCE/IT/15/003)
(2) Tushar Gupta (PCE/IT/15/061)

Mr. Shirish Nagar
Asst. Professor
Information Technology

Date: **31/10/2018**
Place: **Jaipur**

POORNIMA
COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

Date: **31/10/2018**

CERTIFICATE

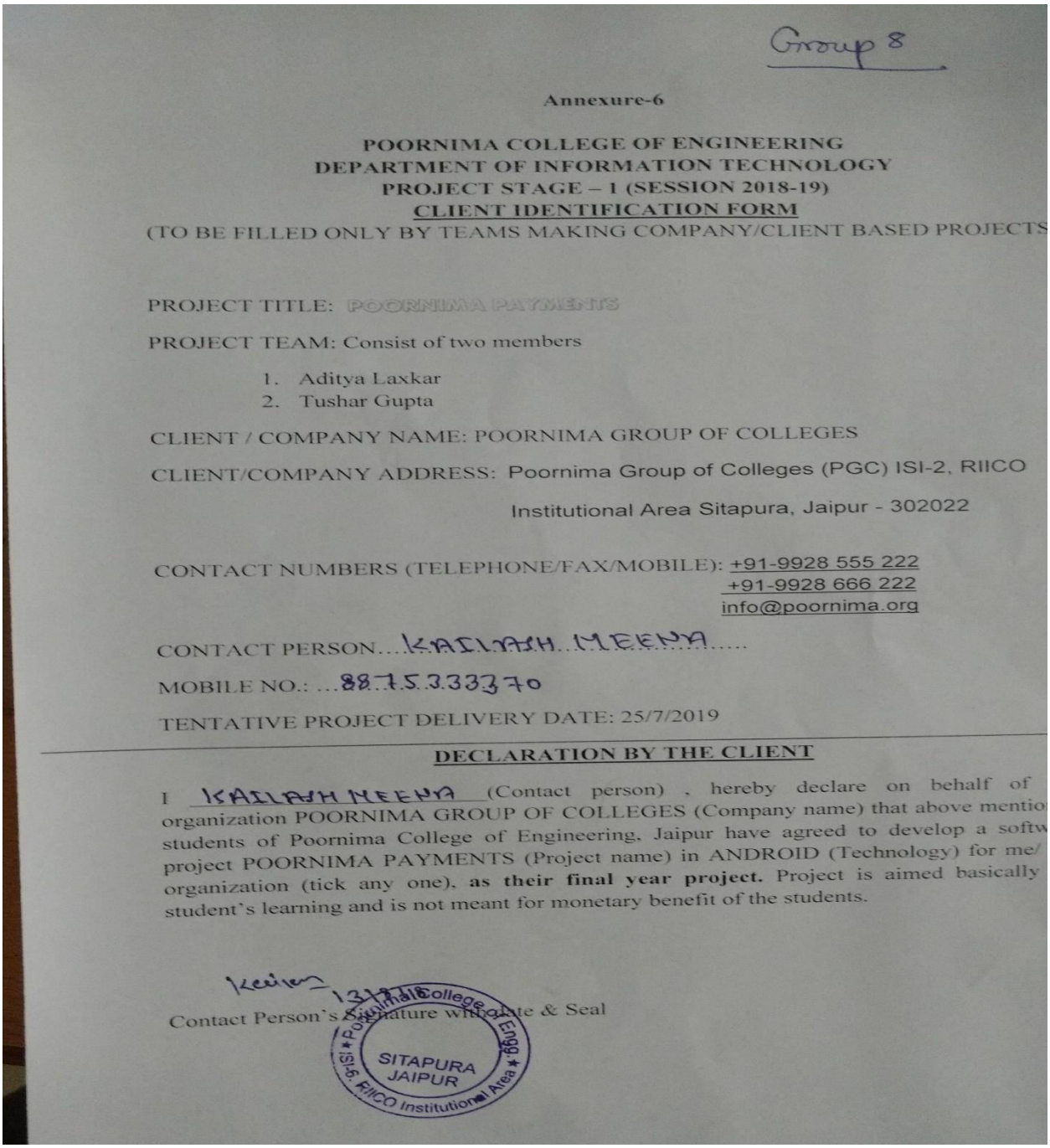
This is to certify that **Project - I** report titled **POORNIMA PAYMENTS** has been submitted by **Aditya Laxkar (PCE/IT/15/003), Tushar Gupta (PCE/IT/15/061)** in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **Information Technology** during the session 2018-19, Odd Semester.

The project work is found satisfactory and approved for submission.

(Mr. Amol Saxena)
HOD - IT

(Mr. Shirish Nagar)
Project Coordinator – IT

Certificate by Client



Group 8

Annexure-6

POORNIMA COLLEGE OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
PROJECT STAGE – 1 (SESSION 2018-19)
CLIENT IDENTIFICATION FORM
(TO BE FILLED ONLY BY TEAMS MAKING COMPANY/CLIENT BASED PROJECTS

PROJECT TITLE: POORNIMA PAYMENTS

PROJECT TEAM: Consist of two members

1. Aditya Laxkar
2. Tushar Gupta

CLIENT / COMPANY NAME: POORNIMA GROUP OF COLLEGES

CLIENT/COMPANY ADDRESS: Poornima Group of Colleges (PGC) ISI-2, RIICO Institutional Area Sitapura, Jaipur - 302022

CONTACT NUMBERS (TELEPHONE/FAX/MOBILE): +91-9928 555 222
+91-9928 666 222
info@poornima.org

CONTACT PERSON... KAILASH MEENA .....

MOBILE NO.: ...8875333370

TENTATIVE PROJECT DELIVERY DATE: 25/7/2019

DECLARATION BY THE CLIENT

I KAILASH MEENA (Contact person) , hereby declare on behalf of organization POORNIMA GROUP OF COLLEGES (Company name) that above mentio students of Poornima College of Engineering, Jaipur have agreed to develop a softw project POORNIMA PAYMENTS (Project name) in ANDROID (Technology) for me/ organization (tick any one), **as their final year project.** Project is aimed basically student's learning and is not meant for monetary benefit of the students.

Contact Person's Signature with Date & Seal

13/7/19

Poornima College of Engg. ★ ISI-2, RIICO Institutional Area ★ SITAPURA JAIPUR

Acknowledgement

I would like to thank God for showing me the way of my success. My deepest gratitude goes to my project supervisor **Mr. Shirish Nagar, Asst. Professor** who has given his best to help me to complete my project. His guidance is excellent, and without his assistance, it would be quite difficult to complete my project. It is an honor to thank the **Head of Department Mr. Amol Saxena** who is always supportive to us. It is my pleasure to thank all the teachers and staffs of **Poornima College of Engineering of Information Technology Department** who has made a great support for my whole study period in this project.

Table of Contents

List Of Tables

List Of Abbreviations

S. No.	Abbreviations	Full Forms
1.	ADT	Android Development Kit
2.	XML	Extensible Markup Language
3.	SQL	Structured Query Language
4.	ADB	Android Debug Bridge
5.	JIT	Just in Time
6.	UML	Unified Modeling Language

List Of Figures

ABSTRACT

E-commerce provides the capability of buying and selling products, information and services on the Internet and other online environments. In an e-commerce environment, payment stake the form of money exchange in an electronic form, and are therefore called Electronic Payment. E-Payment system is secure there should be no threat to the user credit card number, smart card or other personal detail, payment can be carried out without involvement of third party, it makes E payment at anytime through the internet directly to the transfer settlement and form E-business environment. Studied have been carried out on E-Payment system. E-Payment system an integral part of electronic commerce.

An efficient payments system reduces the cost of exchanging goods and services, and is indispensable to the functioning of the interbank, money, and capital markets. Questions are related to E-Payment system in which given options are Agree, Disagree, strongly disagree, strongly agree, Neutral. After analysis and comparison of various modes of electronic payment systems, it is revealed that it is quite difficult, if not impossible, to suggest that which payment system is best.

Some systems are quite similar, and differ only in some minor details. Thus, there are number of factors which affect the usage of e-commerce payment systems. Among all these user bases is most important success of e-commerce payment systems also depends on consumer preferences, ease of use, cost, industry agreement, authorization, security, authentication, non-refutability, accessibility and reliability and anonymity and public policy.

Chapter -1

Introduction to Project

1.1 ABOUT THE PROJECT

The most popular definition of E-Commerce is based on the online perspective of the conducted business. E-commerce provides the capability of buying and selling products, information and services on the Internet and other online environments. As for any trading activity, the issue of safe and reliable money exchange between transacting parties is essential. In an e-commerce environment, payments take the form of money exchange in an electronic form, and are therefore called POORNIMA PAYMENT.

The merchant sells the goods to customer and customer pay the price with the help of E-Payment system. In offline world the payment are made with cash or through cheque

In online sales accepting payment is a curial aspect of the transaction.

The spread of e-payment usage varies unevenly between countries partly due to differences in factors such as quality of regulatory framework and readiness of telecommunication infrastructure. New payment services based on the Internet and mobile phones proliferate in the advanced economies. E-Payment system is secure. There should be no threat to the user credit card number, smart card or other personal detail, payment can be carried out without involvement of third party, it makes E payment at any time through the internet directly to the transfer settlement and form E-business environment.

1.2 PURPOSE AND SCOPE

Purpose of creating this application is to take one step further towards digital marketing as by doing digital transaction we can reduce the baggage of carrying paper currency instead of using digital way of payment.

Also, it will be helpful for users to keep track of their monthly expenses. This application will also provide some benefits like coupons, offers, %off on purchase, extra benefits.

1.3 ABOUT ANDROID

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code. The unveiling of the Android platform on 5th November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.

The Open Handset Alliance, a consortium of several companies which include Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Sprint Nextel and NVIDIA, was unveiled with the goal to develop open standards for mobile devices. Along with the formation of the Open Handset Alliance, the OHA also unveiled their first product, Android, an open source mobile device platform based on the Linux operating system.

Android application development is the development of the dynamic & static android application. It includes Mobile application UI development, Java programming. As firebase has been launched by Google recently application, APIs integration and application maintenance can be done on firebase. Training included UI development of android application, programming & its maintenance through firebase.

.

1.4 Architecture

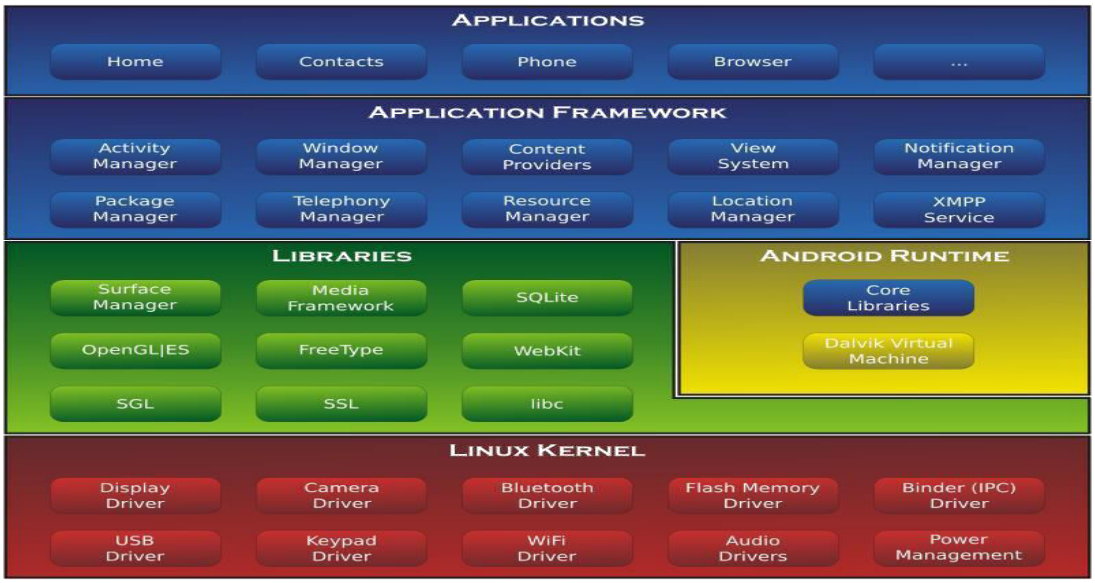


Fig 1.1: Architecture of Android

[Courtesy: Techotopia]

1.4.1 Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, and network stack & driver model. The kernel also acts as an abstraction layer between the hardware & the rest of the software stack. It helps to manage security, memory management, process management, network stack & other important issues. Therefore, the user should bring Linux in his mobile device as the main operating system and install all the drivers required to run it.

1.4.2 Android Runtime

At the same level there is Android Runtime, where the main component Dalvik Virtual Machine is located. It was designed specifically for android running in limited environment, where the limited battery, CPU, memory, data storage is the main issues. Android gives an integrated tool which converts generated byte code from jar to decks file. After this byte code becomes more efficient to run on the small process.

1.4.3 Application Framework

After that, there is Application Framework, written in Java language. It is a toolkit that all applications use, ones which come with mobile device like Contacts or SMS box, or applications written by Google and any Android developer. It has several components. The Activity Manager manages the life circle of the applications and providesa common navigation back stack for applications, which are running in different processes. The Package Manager keeps track of the applications, which are installed in the device. The Windows Manager is Java programming language abstraction on the top of lower level services that are provided by the Surface Manager.

1.4.4 Application Layer

At the top of Android Architecture, we have all the applications, which are used by the final user. By installing different applications, the user can turn his mobile phone into the unique, optimized and smart mobile phone. All applications are written using the Java programming language.

1.5 Android SDK

Android development starts with the Android SDK. While there are many different programming languages and a host of IDEs (Integrated Development Environments) you can use to create an app, the SDK is a constant.

SDK stands for 'Software Development Kit', and essentially provides a selection of tools required to build Android apps or to ensure the process goes as smoothly as possible. Whether you end up creating an app with Java, Kotlin or C#, you need the SDK to get it to run on an Android device and access unique features of the OS.

The Android SDK can be broken down into several components. These include:

- **Platform-tools**: - The Platform-tools are more specifically suited to the version of Android that you want to target. Generally, it is best to install the latest Platform-tools which are also what will be installed by default. After first install though, you need to keep your Platform-tools constantly updated. The tools should be backwards compatible, meaning that you will still be able to support older versions of Android.

- **Build-tools**: - The Build-tools were once categorized under the same heading as the Platform-tools but have since been decoupled so that they can be updated separately. As the name suggests, these are also needed to build your Android apps. This includes the zip align tool for instance, which optimizes the app to use minimal memory when running prior to generating the final APK, and the APK signer which signs the APK (surprise!) for subsequent verification.
- **SDK-tools**: - Arguably the most important parts of this package are in the SDK-tools. You will need these tools regardless of which version of Android you are targeting and these are what will actually create the APK – turning your Java program into an Android app that can be launched on a phone. These include a number of build tools, debugging tools and V*image tools. An example is DDMS, which is what lets us use the Android Device Monitor to check the status of an Android device.
- **The Android Debug Bridge (ADB):** - The Android Debug Bridge (ADB) is a program that allows you to communicate with any Android device. It relies on Platform-tools in order to understand the Android version that is being used on said device and hence it is included in the Platform-tools package. You can use ADB to access shell tools such as log cat, to query your device ID or even to install apps.
- **Android Emulator**: - The Android emulator is what lets you test and monitor apps on a PC, without necessarily needing to have a device available. To use this, you also get an Android system image designed to run on PC hardware. You'll use the 'Android Virtual Device' manager in order to choose which version of Android you want to emulate, along with the device specifications (screen size, performance etc.)

1.5.1 Android Version

Platform Version	API Level	VERSION_CODE
Android 6.0	23	MARSHMALLOW
Android 5.0	21	LOLLIPOP

Android 4.4	19	KITKAT
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.0.x	11	HONEYCOMB
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.0	1	BASE

1.6 FIREBASE

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

Firebase Services are:

1.6.1 Real Time Database

The Firebase Real-time Database is a cloud-hosted No SQL database that lets you store and sync between your users in real time. The Real-time Database is really just one big JSON object that the developers can manage in real time. Real-time syncing makes it easy for your users to access their data from any device, be it web or mobile. Real-time Database also helps your users collaborate with one another. Another amazing benefit of Real-time Database is that it ships with mobile and web SDKs, allowing you to build your apps without the need for servers. When your users go offline, the Real-time Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. The Real-time Database can also integrate with Firebase Authentication to provide a simple and intuitive authentication process.

1.6.2 Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. Normally, it would take you months to set up your own authentication system. And even after that, you would need to keep a dedicated team to maintain that system. But if you use Firebase, you can set up the entire system in under 10 lines of code that will handle everything for you, including complex operations like account merging.

Chapter -2

Client Survey / Result

2.1 Survey

This application is made for POORNIMA COLLEGE OF ENGINEERING as they are our client for this e-wallet application. This application will be beneficial for client as it is an android based mobile application for online payment as user is no longer required to carry paper currency whenever they want to make any payment for any purchase user made.

In POORNIMA COLLEGE OF ENGINEERING payment method for any purchase is based upon paper currency. So, we suggested them to change their payment method to e-wallet as this payment will be beneficial for them as following:

User will be able to keep track of all the payments made through this application in their college canteen and departmental stores.

User will be able to keep all payment transaction history records.

Any cases of fraud payment or no entry of payment by staff. Any kind of theft by staff as product is sold and payment received but not shown in records all those problems can be removed.

College Staff will be able to know about all products availability and needs for their re-ordering.

So, this e-wallet will work like whenever user wants to make any payment then they have to simply scan the QR-CODE given for departmental stores or canteen by using the mobile application which will have all the details of user and all products available in college canteen and college departmental stores.

When user want to make payment, they need to have sufficient balance in their e-wallet and they can have balance in their e-wallet simply by linking their account to any valid (authorized form RBI) bank credit cards. And then transfer require amount into this application which will allow user to make payments through this application.

Sending money through QR Code:

you can transfer money to another user or college canteen and departmental stores using their QR code. Read below to know how:

- Click on ‘Send’
- A page will open with scan code option
- Scan the QR Code with your smartphone of the canteen / departmental store or other user you have to pay
- Click on ‘Send’

You can view your QR code by swiping for the menu from the left and selecting ‘My QR code’ from the list.

We have all seen barcodes in our day-to-day lives: at the supermarket, clothing store, parking garage – you name it. But in recent years, a different type of barcode has become increasingly popular thanks to the mobile revolution:

QR, or quick-response, codes.

The difference between a QR code and a regular barcode is that QR codes store information in two dimensions, so they appear as pixelated black and white squares, each containing a unique pattern. Also, they are designed to be read by a mobile phone camera, whereas a regular barcode uses a scanner.

Figure 2.1 Barcode and QR Code

QR codes have many different uses, but one of the most popular uses is to make payments using a mobile device.

How can QR codes be used to make payments?

Mobile shopping is on the rise, and many retailers are looking for new ways to let people pay in-store using their mobile phones. In fact, the number of people in the US making in-store mobile payments is expected to reach 150 million by the end of 2020!

One way of making mobile payments in-store is by using QR codes. All you need is a mobile phone with a camera, and a mobile app that can scan, store, and share QR codes.

Here are the three main payment types you can make using QR codes have many different uses, but one of the most popular uses is to make payments using a mobile device. QR codes, and how they work:

1. **Paying retailers with QR scanners:**

   At the check-out, the cashier will enter the amount to pay. You will then open your QR scanning app and display your unique QR code to the retailer. The retailer will scan the QR code to identify you and deduct the money from your mobile wallet, using a compatible mobile payment app.

2. **Paying retailers without QR scanners:**

   In this scenario, the retailer will display a QR code and you will scan it using the QR scanning app on your mobile device. The app will identify the retailer. Then you can enter the payable amount and complete the payment. One example of this in action is the system used by US department store, Kohl's. Kohl's Charge cardholders can now make payments via their mobile app by scanning a QR code at the check-out.

3. **Paying individuals (such as self-employed professionals):**

   Maybe you will need to pay your taxi fare upon drop-off, rent to your landlord, or even pay back a friend you borrowed cash from in the past. In this case, both you and the recipient open the QR scanning payment app. You will scan the recipient's unique QR code, add the amount to pay and complete the transaction.

What are the benefits of using QR codes?

There are many benefits of using QR codes to make payments:

1. No special equipment needed:

   One of the challenges of mobile payments is finding a solution that works with different devices. QR codes help overcome this challenge, as all you need to scan a QR code is a camera on your mobile phone.

2. Quick and easy to use:

   You don't need to enter the details of the person you are paying, just scanning the QR code will identify them instantly.

3. More convenience:

   You can make payments using QR codes from any location with only a few taps on the mobile device.

4. Increased safety:

   Storing your payment details in your mobile phone and carrying it around is much safer than bringing your entire wallet full of cash and credit cards everywhere you go. The chances of theft and fraudulent purchases using your payment details are much less with QR codes.

QR codes are a promising trend for the future of mobile payments, and are an exciting, easy and convenient way to pay using just your mobile phone.

As a result, POORNIMA COLLEGE OF ENGINEERING seems happy with this product proposal and they are read to invest in this project and want us to build an application for payments in their departmental stores and college canteen. After application response in their campus they will look forward for future changes and up-gradation in their mobile based e-wallet application. But till now they want us to make application for canteen and departmental store payments by user.

Chapter –3

Software Requirement Specification (SRS)

3.1 Introduction

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated.

3.2 Purpose

Purpose of creating this application is to take one step further towards digital marketing as by doing digital transaction we can reduce the baggage of carrying paper currency instead of using digital way of payment.

Also, it will be helpful for users to keep track of their monthly expenses. This application will also provide some benefits like coupons, offers, %off on purchase, extra benefits.

3.3 Feasibility

1.) Reliability: The solution should operate with minimum fault in normal condition as well as withstand unexpected circumstances. This criterion is very important due to its link to finance and the distrust of the customers for the new payment method.

2.) Security: The solution relates to the protection of payment details and customers' identity besides preventing business fraud to happen. In short, it should cover the issues of anonymity, privacy and non-repudiation

3.) Speed of payment process: The speed of payment process has to be very fast as the customer could not wait patiently for a micro payment transaction and get frustrated with delay.

4.) Social acceptability: The new payment process could highly be affected by the market acceptance which mostly comes from public's prior knowledge and previous experiences with existing mobile payment method.

3.4 Functional / Non-Functional Requirements

1 Functional Requirements

1.1 User Registration

- User needs to login to application
- Valid ID should be updated in order to use application
- Email must be used in order to login

1.2 Add Money to Wallet

- User must be able to add money to wallet using debit or credit card
- Card must be authenticated from bank by admin before allowing user to transact payments.

1.3 Transfer money

- User must be able to transfer money to another user.
- User must be able to make payments using this application.

1.4 Add cards to user profile

- User must be able to add their debit/credit cards to application with bank authentication
- If any user tries to fraud application by invalid cards then application must be able to detect these frauds.

1.5 Maintain customer profile

- The system shall allow user to create profile and set his credential.
- The system shall authenticate user credentials to view the profile.
- The system shall allow user to update the profile information

1.6 Provide user Support

- The system shall provide online help, FAQ's customer support, and sitemap options for customer support.
- The system shall allow user to enter the customer information for the support.

- The system shall display the online help upon request.

2 Nonfunctional Requirements

2.1 Performance Requirements

- You will be sign in within 26 sec.
- Credit card validate within 5 sec.
- Web support 250 customers logged at the same time

2.2 Security Requirements

- Web app will validate credit cards against fraud.
- Web app keep the information of the customers and merchants safe.
- Web app obeys all the security laws of INDIA relating to cyber traffic.

2.3 Software Quality Attributes

- The web app well is easy to use.
- The web app uses simple English so that user cannot confuse with terms.
- The web app should be easy to upgrade

2.4 Graphical User Interface

- The system shall provide a uniform look and feel between all the application pages.
- The system shall provide use of icons and toolbars.

2.5 Performance

- The product shall be based on web and has to be run from a web server.
- The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run.

- The performance shall depend upon hardware components of the client/customer.

3.5 Security

1 Data Transfer

- The system shall use secure sockets in all transactions that include any confidential customer information.
- The system shall confirm all transactions with the customer's mobile application

2 Data Storage

- The customer's mobile application shall never display a customer's password. It shall always be echoed with special characters representing typed characters.
- The customer's mobile application shall never display a customer's credit card number after retrieving from the database. It shall always be shown with just the last 4 digits of the credit card number.
- The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.
- The system's back-end servers shall only be accessible to authenticated administrators.

3.6 Technical Requirments (Hardware /Software)

1 Interfaces

- There are many types of interfaces as such supported by the Poornima Payment software system namely; User Interface, Software Interface and Hardware Interface.
- There shall be logical address of the system in IPv4 format.

2 User Interfaces

- The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or chrome by which user can access to the application.
- The user interface shall be implemented using any tool or software package like Android Studio.

3 Hardware Interfaces

- Since the Poornima Payment application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. internet connectivity with device is must.

4 Software Interfaces

- The Poornima Payment system shall communicate with the Configuration to identify all the available components to configure the product.
- The Poornima Payment system shall communicate with bill generation system to identify available payment methods, validate the payments and process payment.

5 Communications Interfaces

- The Poornima Payment system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

3.7 Other Requirements

1 Supportability

1.1 Configuration Management Tool

- The source code developed for this system shall be maintained in configuration management tool.

1.2 On-line User Documentation and Help System Requirements

As the product is Poornima Payment, On-line help system becomes a critical component of the system which shall provide –

- It shall provide specific guidelines to a user for using the Poornima Payment system and within the system.
- To implement online user help, link and search fields shall be provided.

1.3 Licensing Requirements

- Not Applicable

1.4 Legal, Copyright, and Other Notices

- Poornima Payment should display the disclaimers, copyright, word mark, trademark and product warranties of the Marvel electronics and home entertainment

3.8 System Feature

Module 1: User

1.1 Payer (Customer)

- Payer register on application if new user?
- If payer is an existing user then will login using user id and password.
- Make payment request to Payee via application.
- Payment can be made either using mobile no. or QR-CODE.
- Payer will add money in wallet if balance is low or payment amount exceeds wallet balance.
- Payer can add money to wallet using any valid credit card or debit card.
- If money added to wallet successfully then payer can make payment otherwise payment will be declined.

- Payer can check all transaction history made by payer in Account passbook.

Other features for Payer

- Can edit personal profile (customization of profile allowed).
- Can transfer money to other user.
- Ask for change password if user forgets current password.

1.2 Payee (service provider)

- Payer register on application if new user?
- If payer is an existing user then will login using user id and password.
- Make payment request to Payer via application.
- Payment can be made either using mobile no. or QR-CODE.
- Money will be added to wallet once transaction from payer side completed.
- Payee can check all transaction history made by payer in Account passbook.
- Other features for Payer
- Can edit personal profile (customization of profile allowed).
- Can transfer money to another user.
- Ask for change password if user forgets current password.

Module 2: Bank Management

- Will check all validation to credit card or debit card information given by users.
- Authentication to admin about user's transaction will be done by bank.
- Can ask admin to deactivate user account or block card if found faulty.

Module 3: Admin Panel

- Admin can check all details of all users.
- Admin can deactivate any user account if found breaching code of conduct.
- Admin will take care of all managing all details of all users and payments.
- Admin can also add money to any users account if required.
- Admin also require to add money to wallet using credit card or debit card. For making any payments.
- Admin can reset users account password if requested by user.
- Admin can keep track on user's activity in application.
- Admin will have access to database of the application
- Admin can check all transactions history done via Poornima Payment application
- Admin can add coupons, offers and discount deals to the application.
- Admin can update all canteen items their prices, departmental store items and their prices.

Module 4: System Working

4.1 User Registration & Welcome

- Only appears once (the first time the application is run)
- Enables the user to customize his/her account settings and preferences

4.2 Group Creation & Management

- Streamlines the process of creating and organizing groups
- Provides support for multiple groups
- Allows the user to add group members manually or from contacts list

4.3 Member-to-Member Transaction

- Enables group members to simulate transfers of debt, payments made, etc.
- Adjusts member balances accordingly
- Records relevant information (amount paid, members involved, etc.)

4.4 Final Debt Resolution

- Calculates the most efficient method of sorting out debts
- Notifies group members of unresolved debts, credits, etc.
- Offers the option to disband a group once all payments are made

4.5 Show All Debts

- Enumerates all of a user's unresolved debts across each group he/she is a part of
- Provides easy access to relevant information (past transactions, group info, etc.)
- Offers the option to resolve a debt (or debts) immediately

4.6 Help Menu

- Displays a list of topics covering the different components of SplitPay
- Offers detailed information on each feature, menu, etc.
- Can be accessed at any time via the Settings menu

3.9 Analysis Diagrams

1. Use Case Diagram:

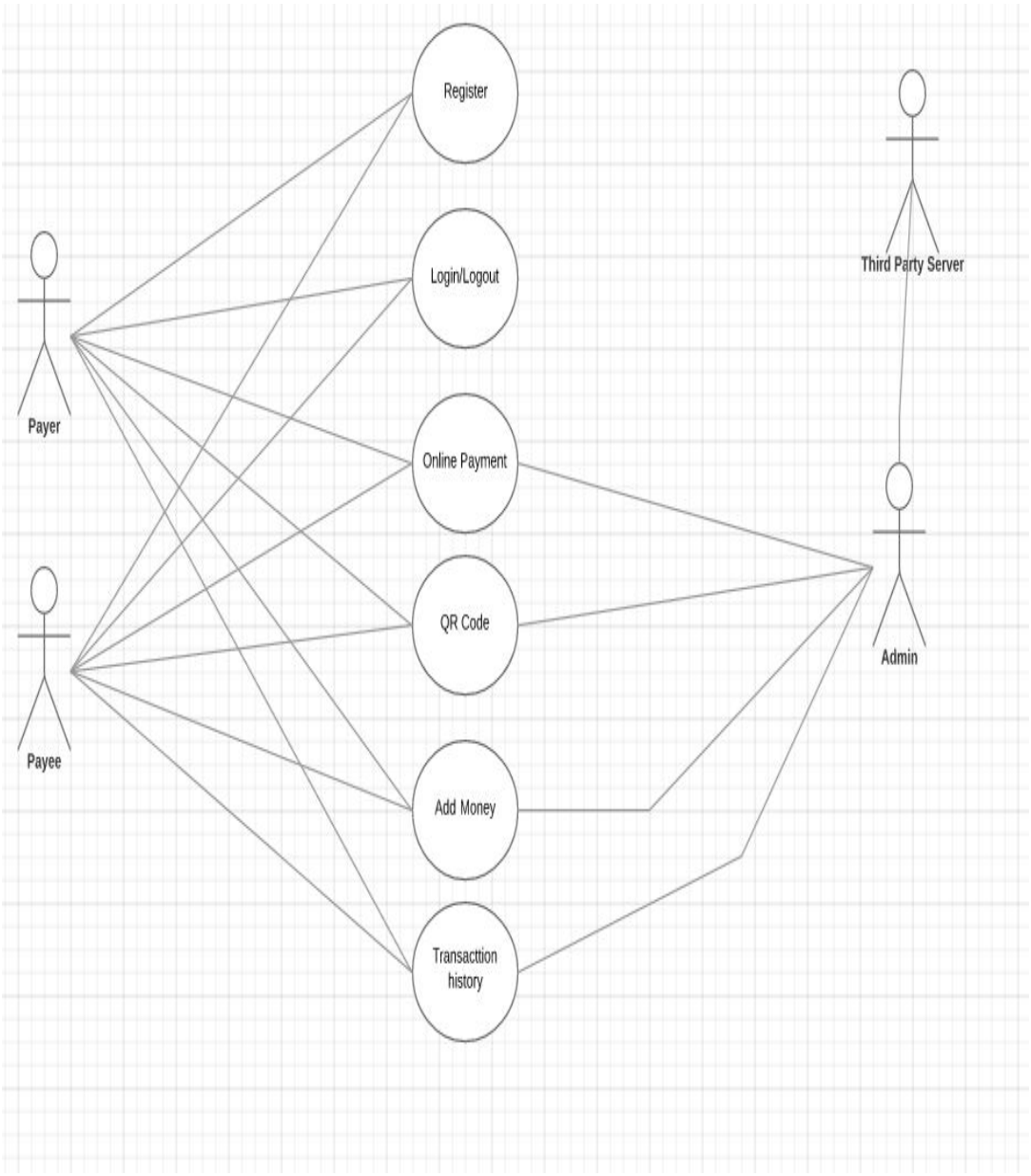


Figure 3.1: Use Case Diagram

2. Sequence Diagram:

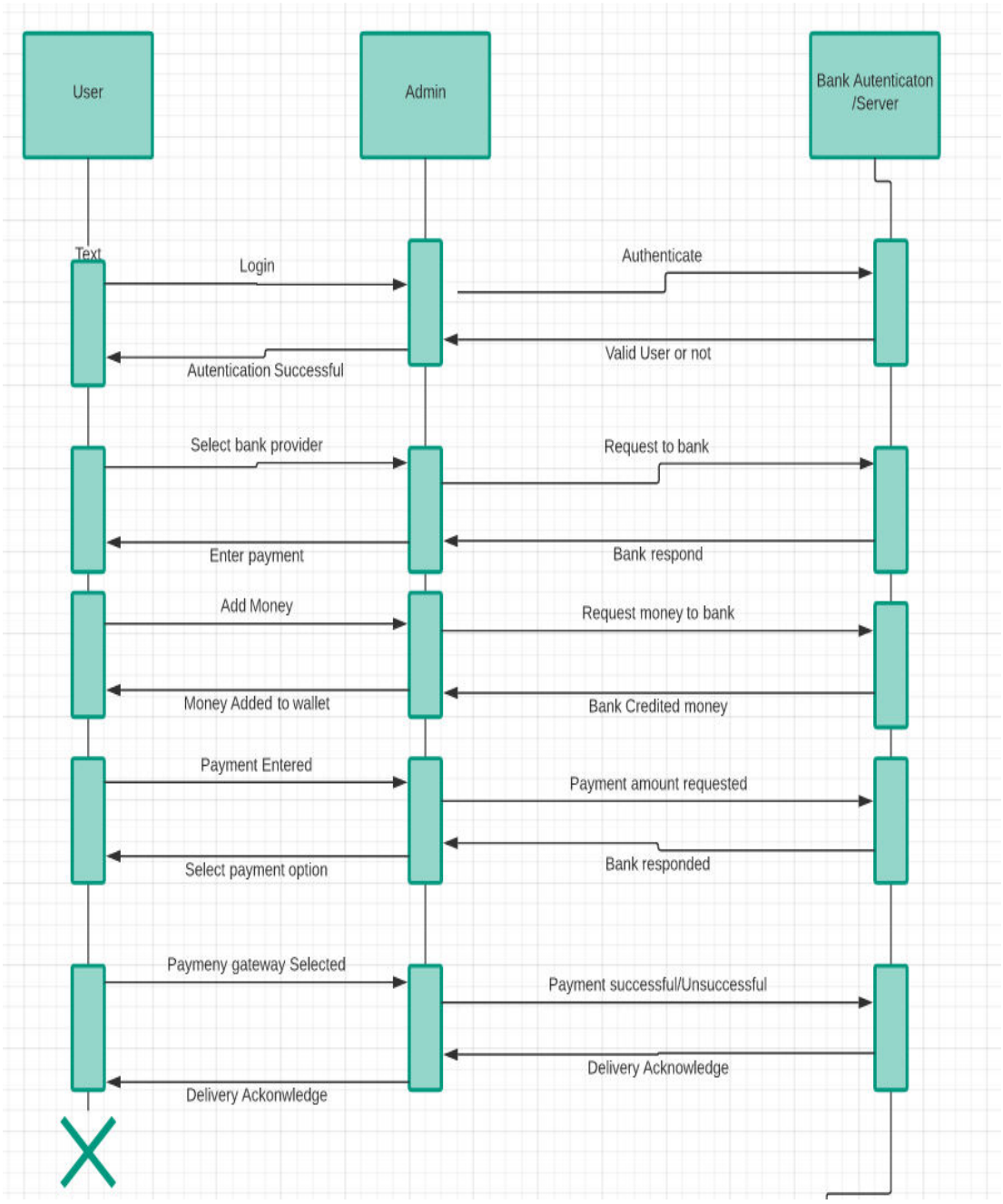


Figure 3.2: Sequence Diagram

3. Component diagram

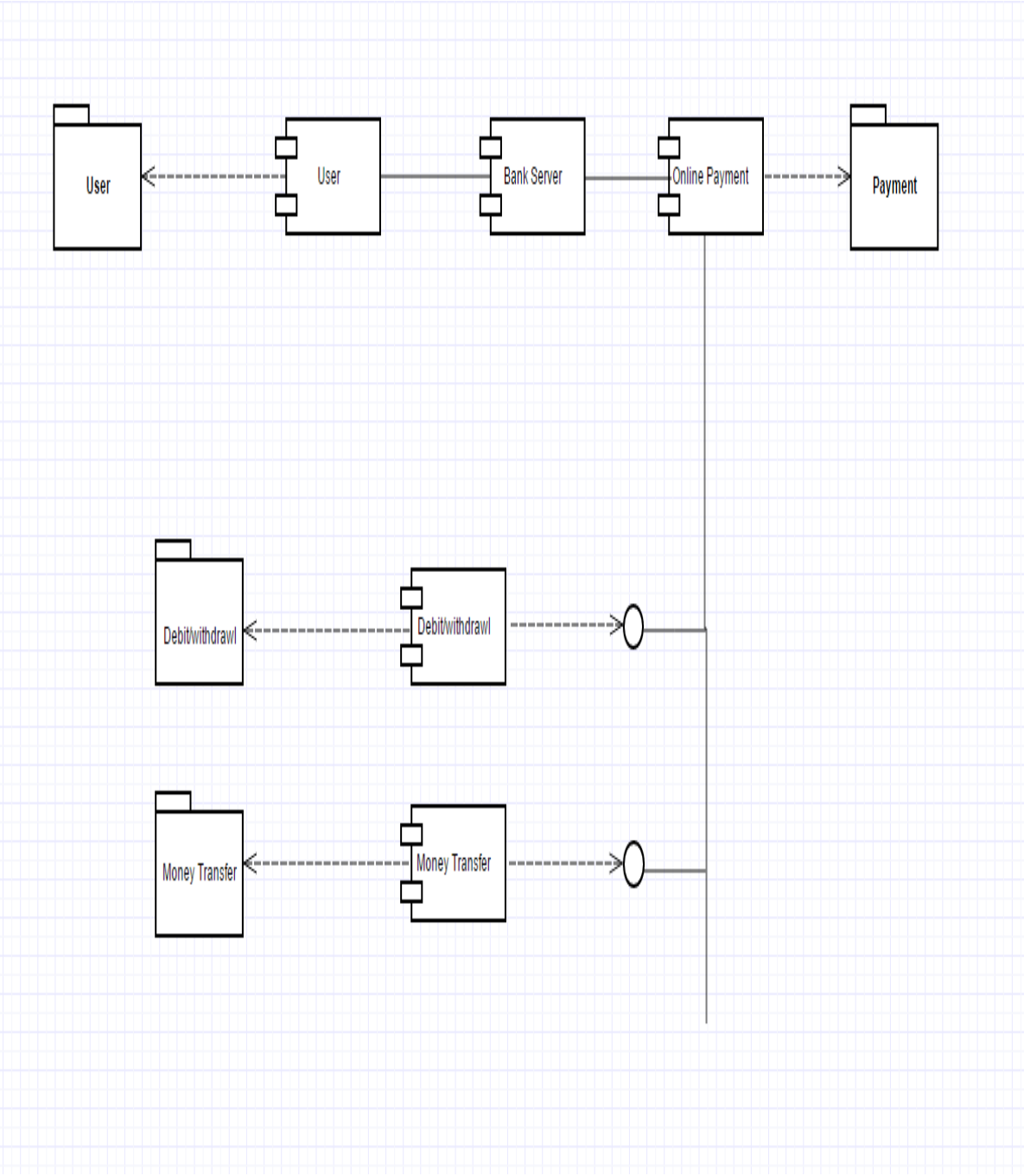


Figure 3.3: Component Diagram

4. Data Flow Diagram:

Level-1

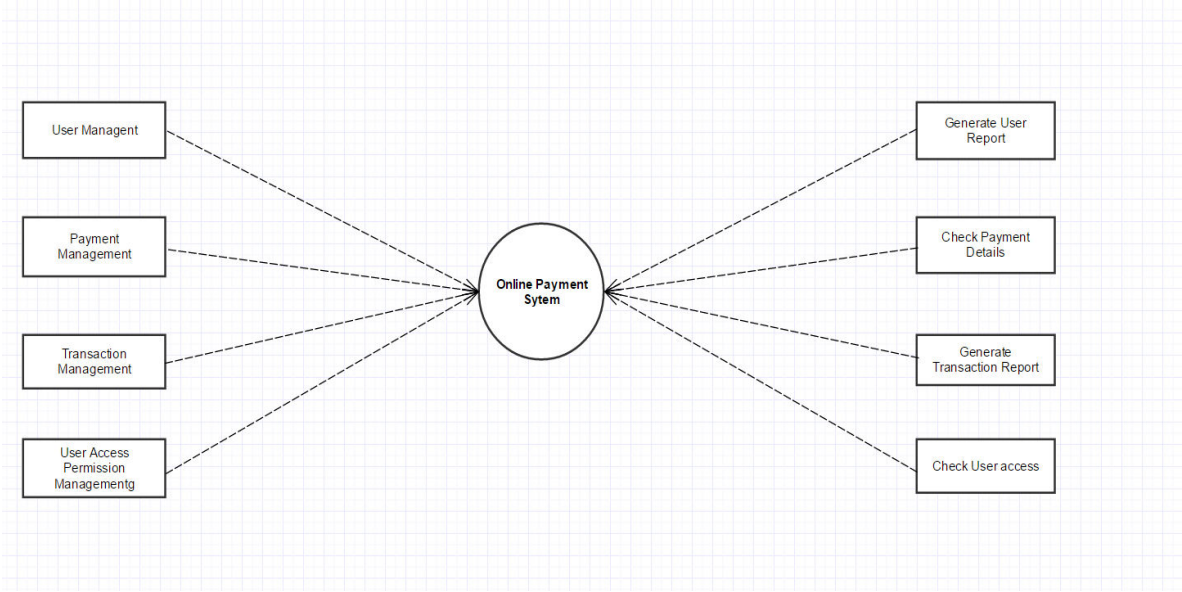


Figure 3.4(a): Data Flow Diagram Level -1

Level-2

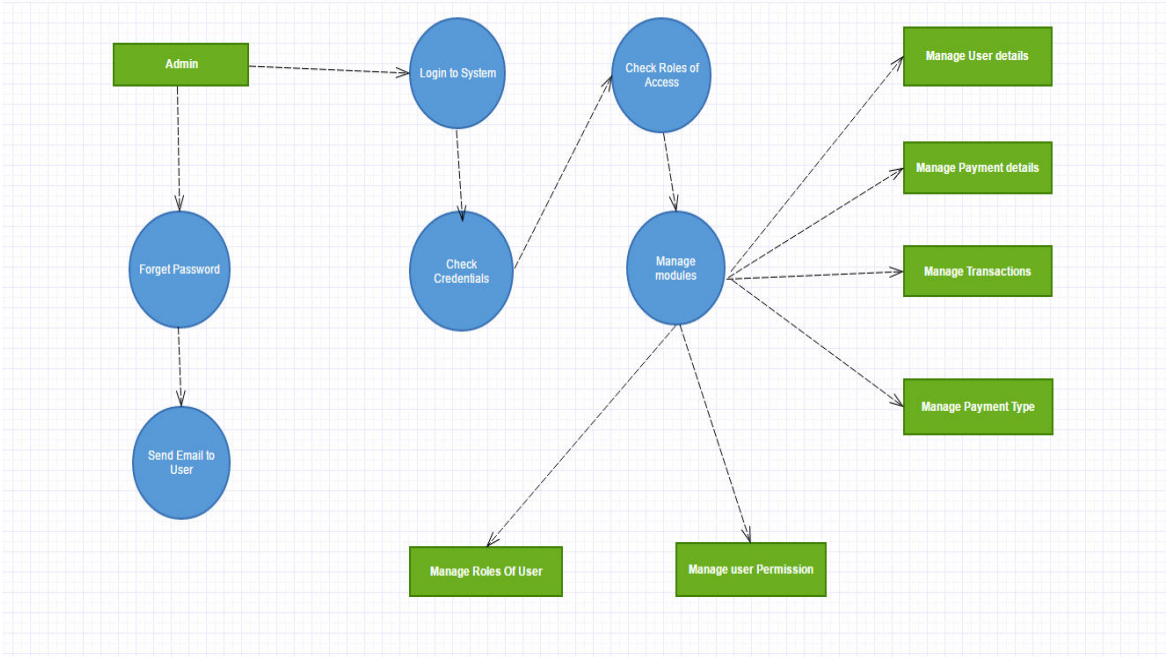


Figure 3.4(b): Data Flow Diagram Level- 2

5. ER Diagram

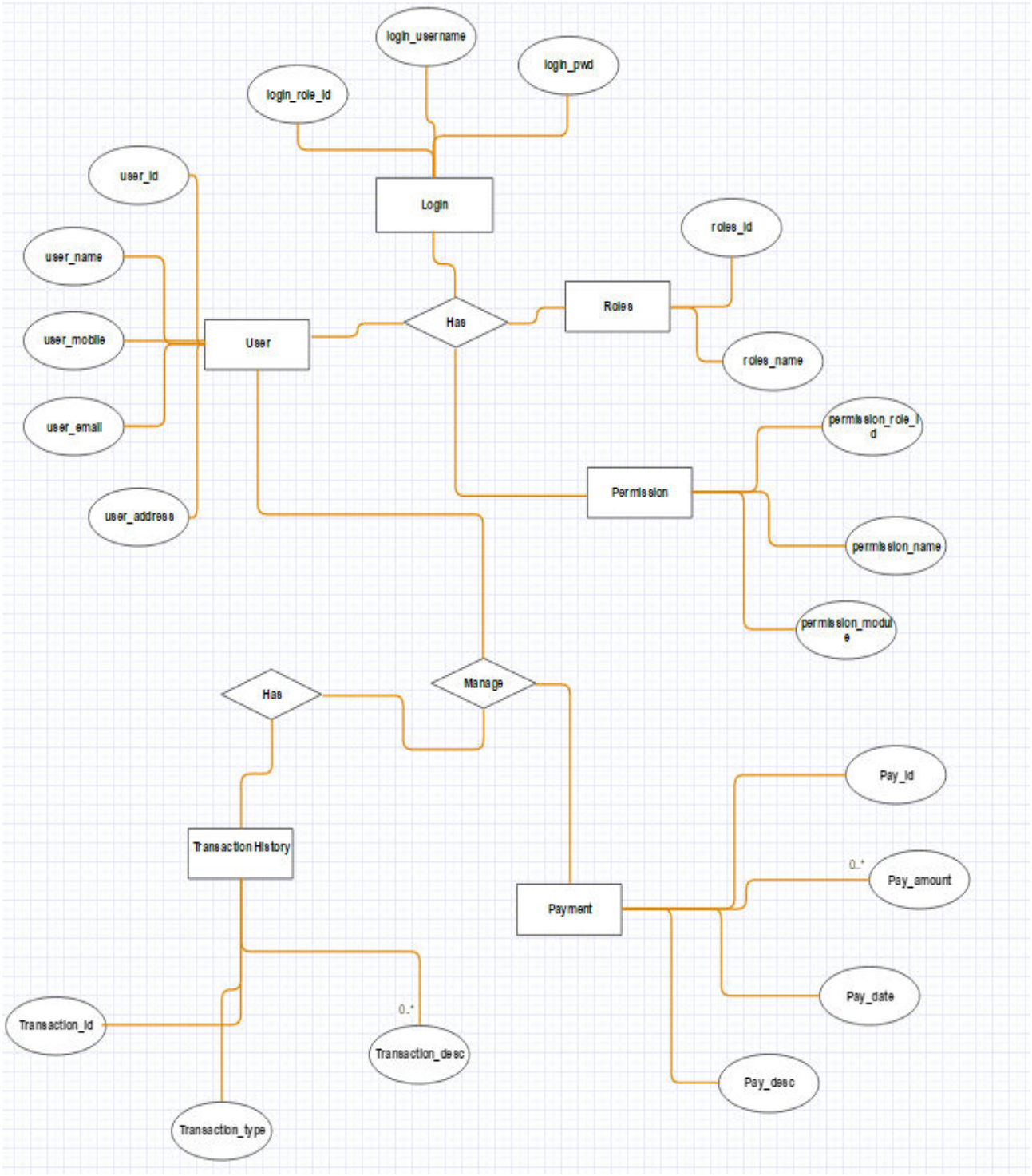


Figure 3.5: ER Diagram

6. Activity Diagram:

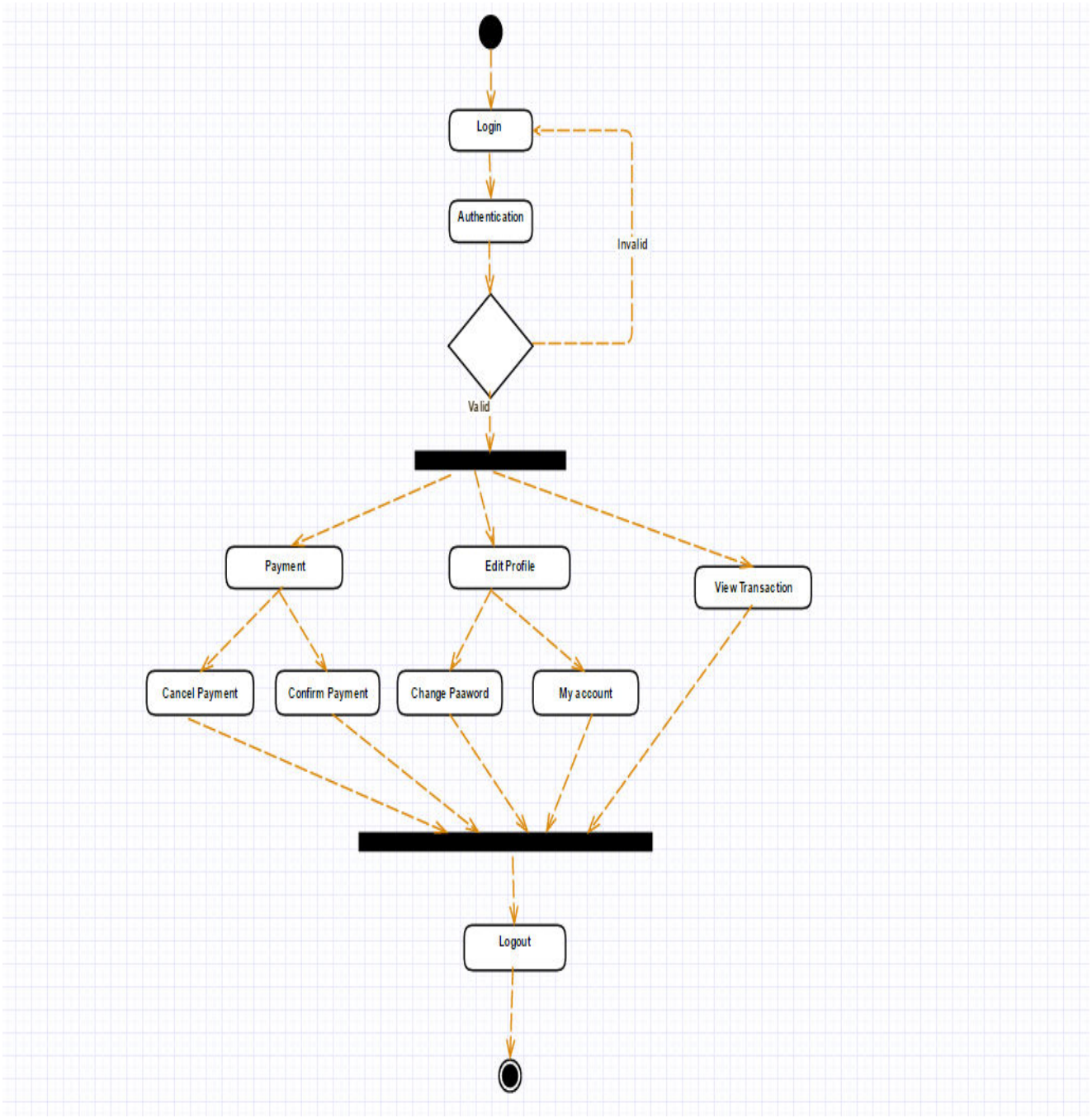


Figure 3.6: Activity Diagram

7. Project Schedule Diagram:

POORNIMA COLLEGE OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
SESSION 2017-18: ODD SEMESTER
PROJECT - I (7ITPR) - GANTT CHART

PROJECT - I SCHEDULE			DURATION (days)
START DATE	END DATE	DESCRIPTION	
3-Jul-2018	10-Jul-2018	Inaugural Presentation	2
17-Jul-2018	19-Jul-2018	Group Finalization	3
19-Jul-2018	21-Jul-2018	Selection of Category of project	3
21-Jul-2018	23-Jul-2018	Problem identification and definition	3
23-Jul-2018	24-Jul-2018	Guide's Acceptance and Project Approval	2
24-Jul-2018	26-Jul-2018	Literature Survey of Problem	3
26-Jul-2018	27-Jul-2018	Synopsis Submission	2
27-Jul-2018	14-Aug-2018	Synopsis Presentation	19
31-Jul-2018	13-Aug-2018	ProjectPresentation	14
8-Aug-2018	20-Aug-2018	Requirement Analysis & Planning Submission	15
20-Aug-2018	27-Aug-2018	System Design DocumentSubmission	8
27-Aug-2018	14-Sep-2018	Source code development (Phase - I)	19
14-Sep-2018	21-Sep-2018	Unit Testing (Phase - I)	8
28-Sep-2018	17-Oct-2018	Source code development (Phase - II)	20
18-Oct-2018	28-Oct-2018	Unit Test (Phase-II), Code Review& Integration Testing	11
29-Oct-2018	2-Nov-2018	ProjectReport Submission	5

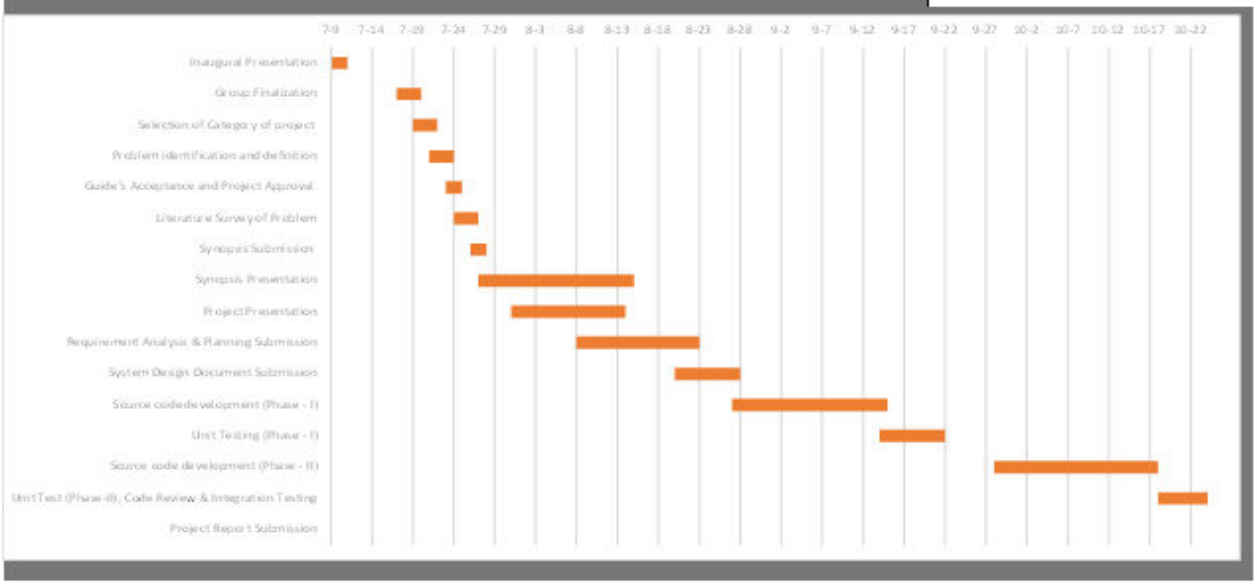


Figure 3.7: Project schedule Diagram

Chapter -4

Software Design Document

4.1 Introduction

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

4.2 Purpose

Purpose of creating this application is to take one step further towards digital marketing as by doing digital transaction we can reduce the baggage of carrying paper currency instead of using digital way of payment. Also, it will be helpful for users to keep track of their monthly expenses. This application will also provide some benefits like coupons, offers, %off on purchase, extra benefits.

4.3 Software Design

1. Architectural Design

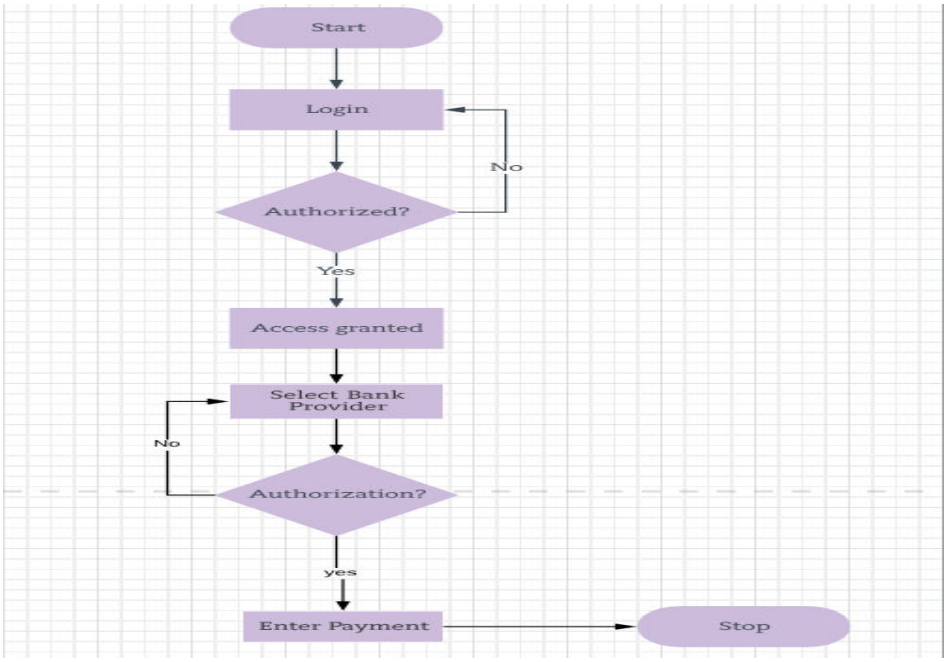


Figure 4.1: Architecture Diagram

2.Class Diagram

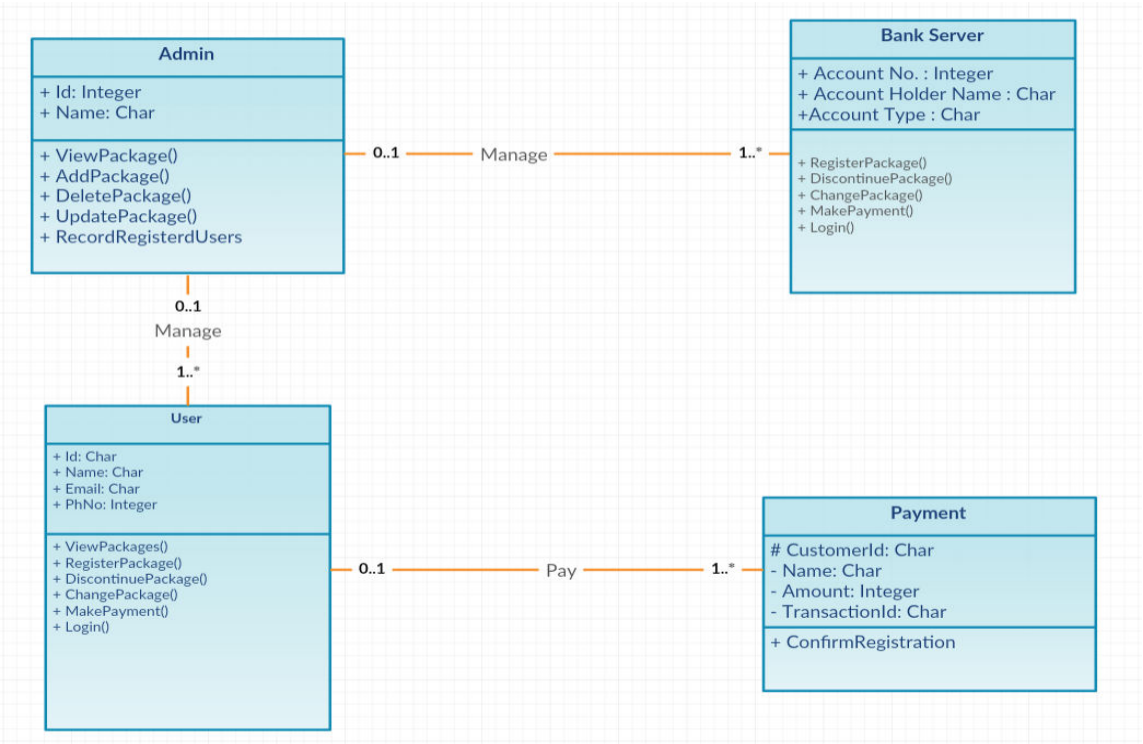


Figure 4.2: Class Diagram

3 Deployment Diagram:

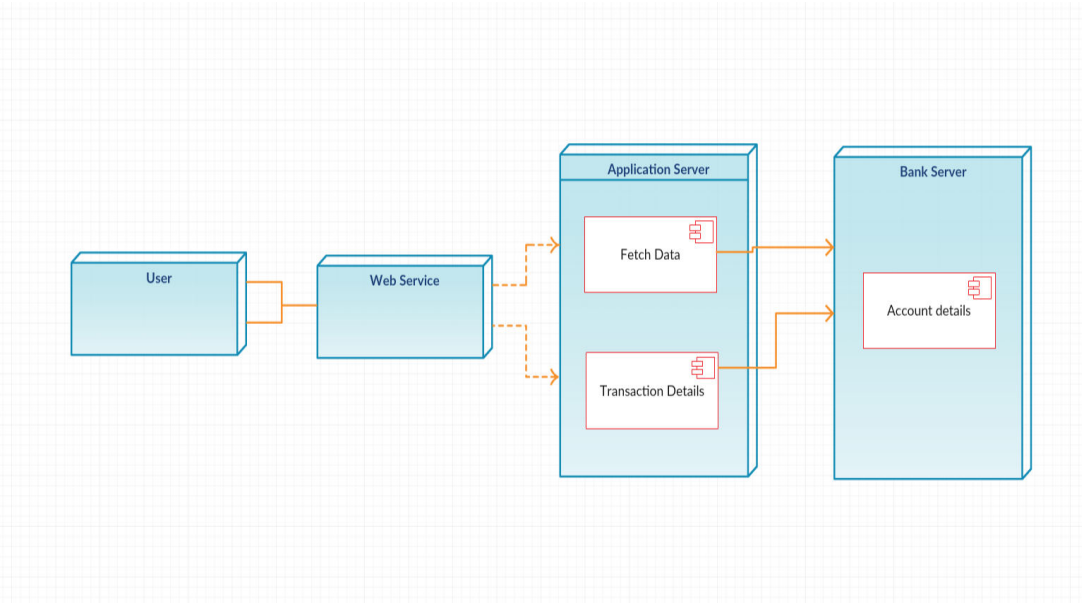


Figure 4.3: Deployment Diagram

4 Package Diagram:

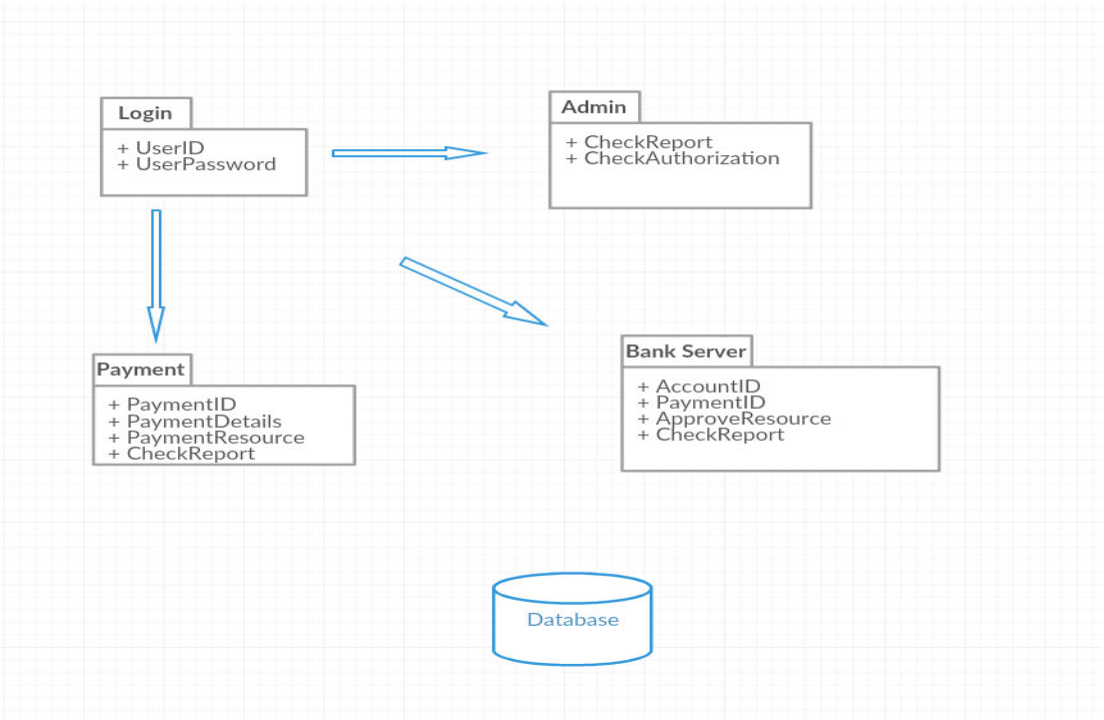


Figure 4.4: Package Diagram

5 Databases Diagram:

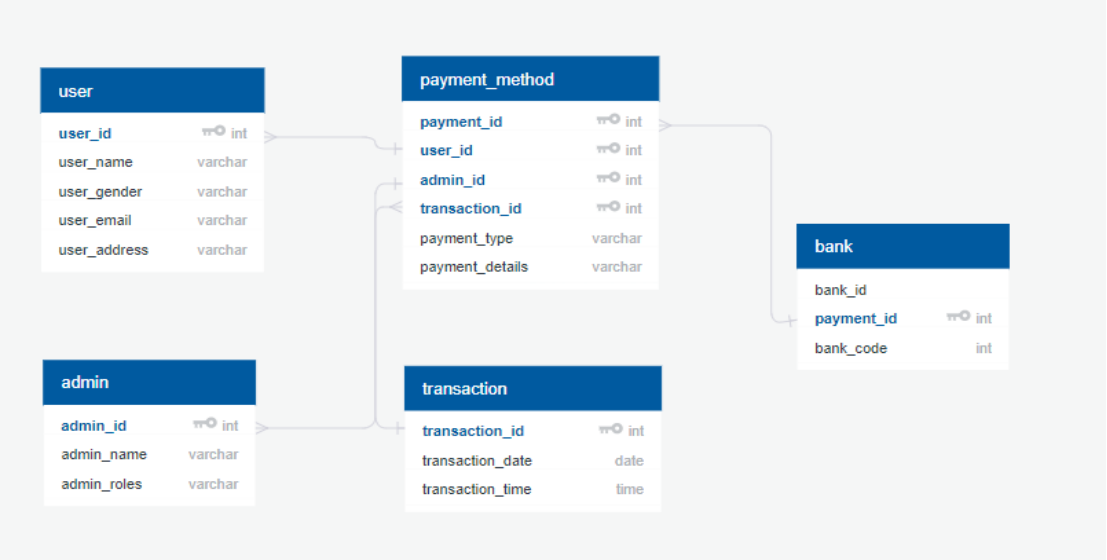


Figure 4.5: Database Diagram

6. Graphical User Interface (GUI):

1) Login Page: In this page user login to E-wallet application if user already have existing account in the application. Otherwise user will go to new user activity and then create new user account by fill all the required information.

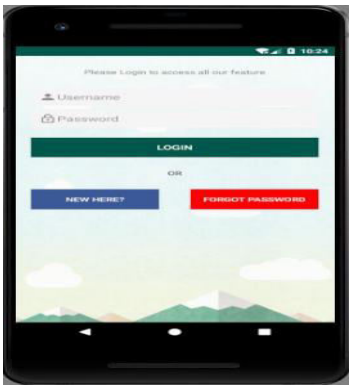


Fig 4.6(a): Login Page

Otherwise user have an existing account on the application but user forget their user login account password then user can go to forget password page and then by giving their username and can change their account password and can again login to the E-wallet application.

2) Sign up Page: In this page user create new user account by fill all the required information. Like name, mobile number, email address, password and confirming that password, user gender then finally clicks on sign up button and user account will be automatically created in the E-wallet application and next time when user want to use this E-wallet application then user have to only login the E-wallet application and no need of creating new account every time user wants to use this application.

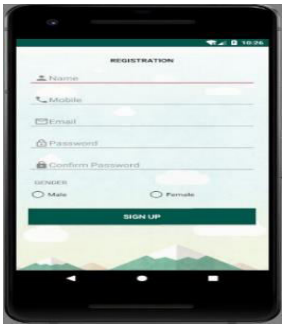


Fig 4.6(b): Sign Up Page

3) Forget password: In this page already, existing user can change their account password if they forget their account password and user wishes to continue using application using that same account then user can reset their account password if they

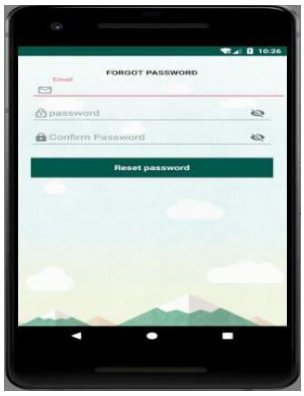


Fig 4.6(c): Forget Password

forget their account password.

4) User Profile Page: In this page user will be able to check its account profile details like its username QR-Code, contact number, Help, guidelines, terms and conditions of the E-wallet application.

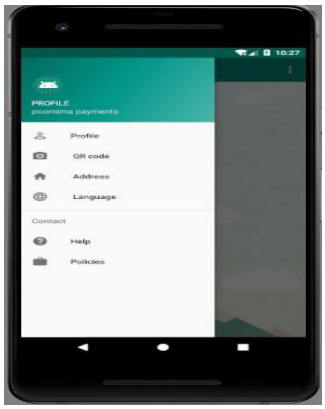


Fig 4.6(d): User Profile Page

7. API (Application Program Interface)

S. No	Class / Library	Function	Arguments (& Data types)	Description of the functionality
1	android.os. Bundle	osBundle	OnCreate (Bundle savedInstanceState)	Android using Bundle for sharing variables. Bundle is used to pass data between Activities.
2	android.app. ActivityManager	ActivityManager	ActivityManager.class	This class gives information about, and interacts with, activities, services, and the containing process.
3	android.app. ActionBar	getActionBar ()	ActionBar.class	The ActionBar, now known as the App Bar, is a consistent navigation element that is

				standard throughout modern Android applications.
4	android.view.Menu	ContextMenu	MenuItem.getMenuInfo()	By default, every Activity supports an options menu of actions or options. You can add items to this menu and handle clicks on your additions.

Table 1: API Specification

Chapter- 5

Test Cases Design

No	Test Objective	Test Step	Expected result	Result
1.	To ensure that the app is properly installed in the mobile.	i. Turn on Android studio. ii. Run the application on mobile through ADB. iii. Check if app shows expected result.	The app should work properly.	Pass
2.	To ensure that the app is able to login the page.	i. Run the app in phone. ii. Click on the login as user.	The login should be done properly.	Pass
3.	To ensure that the login details given are properly implemented.	i. Run the app in phone. ii. Click on the button. iii. Check if it is working properly.	The details should be stored and retrieved from database accordingly.	Pass

Table 2: Test Cases

Chapter -6

Report from Guide and Recommendation

Chapter -7

Conclusion and Future Scope of Work

7.1 INTRODUCTION

The project will have two modules- one for admin of application and other for user of application (College students and college faculty and other stuff members). Their working will be as given:

i) Admin: A system for an Admin who can have control over all activities of user and all activities like editing in menu items or payment methods. Admin can make update in menu items of departmental stores and college canteen.

ii) Users: College students and faculties can register the details to the Application. The facility to see available services in the Campus. Maintains the user records, so that user can check where they had spent their money and their previous purchase. Thus, using this application admin can keep an eye on all the payments made, purchase made from departmental stores and canteen.

The application is made such that all the people who are responsible for a payment to college departmental stores and canteen are regularly connected to each other when required.

7.2 FUTURE SCOPE

This application will be a payment method for all college staff, faculty and students who so ever wants to make any purchase from the college canteen or college departmental stores.

So, this application will never go out of use. As no user will stop purchasing from college canteen and departmental stores. And it is an e-wallet payment method so user will no longer require to carry paper currency only cards and sufficient balance in e-wallet will be enough for any purchase from college.

7.3 LESSONS LEARNED, SKILLS DEVELOPED

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

Chapter-8

FAQ About Projects

1) How is the Android software developed?

2) What is involved in releasing the source code for a new Android version?

3) What is the application of your project?

4 What are the functional and non-functional requirements of your project?

5) What is the future scope of your project?

6) What are the technologies used in your project?

References

1. Android Programming: The Big Nerd Ranch Guide (3rd Edition) by Bill Phillips, Chris Stewart, and Kristin Marsicano
2. Android Studio 3.0 Development Essential by Neil Smyth
3. Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers) Third Edition Edition by Ed Burnette
4. Programming Android: Java Programming for the New Generation of Mobile Devices (2nd Edition) by Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura
5. The Android Developers Cookbook: Building Applications with the Android SDK by James Steele and Nelson To

6. http://developer.android.com/reference/android/hardware/Camera.html
7. http://developer.android.com/guide/topics/location/index.html
8. http://code.google.com/apis/maps/documentation/places/
9. https://www.x.com/community/ppx/dev-tools
10. https://stackoverflow.com/questions/android-database-to-server-connectivity

APPENDICES

1. Android Version

We have used Android Studio 3.1 for making our application.

2. Firebase

Firebase can be divided into: -

a.) Develop: It includes:

i. Realtime Database

ii. Auth

iii. Cloud Functions

iv. Hosting

v. Perfomance Monitoring

b.) Grow: It includes:

i. Firebase Analytics

ii. Remote Configuration

iii. AdMob

iv. Dynamic Links

v. Cloud Messaging

3. API Specification

- android.os.Bundle: Android using Bundle for sharing variables. Bundle is used to pass data between Activities.
- android.support.v4.app.Fragment: android.support.v4.app.Fragment is the Fragment class in the android support library, which is a compatibility package that allows you to use some of the newer features of Android on older versions of Android.
- android.app.ActionBar: The ActionBar, now known as the App Bar, is a consistent navigation element that is standard throughout modern Android applications.
- android.app.ActivityManager: This class gives information about, and interacts with, activities, services, and the containing process.
- android.view.View: View is one of the most general class in Android. It holds references to single piece of UI. See link above. Passing View view as argument in methods in most cases gives you opportunity to call method associated with this view.
- android.widget.Button: Android Button represents a push-button. The android.widget.Button is subclass of TextView class and CompoundButton is the subclass of Button class.
- android.widget.TextView: A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.