Canteen Management system using E-Wallet

Submitted in partial fulfilment of the requirements

of the degree of

Bachelor of Engineering

by

Kalpesh Juvekar

Akash Katkar

Nitin Rohira

under the guidance of

Mrs. Smita Jangale

Department of Information Technology

Vivekanand Education Society's Institute of Technology

2017-18

Vivekanand Education Society's
Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Kaplesh Juvekar, Mr. Akash Katkar** and **Mr. Nitin Rohira** of Fourth Year Information Technology studying under the University of Mumbai have satisfactorily presented the project entitled **Canteen Management System E-wallet** as a part of the PROJECT-II for Semester-VIII under the guidance of **Mrs. Smita Jangale** in the year 2017-2018.

Date:

(Name and sign)

External Supervisor

(Name and sign)
Head of Department

(Name and sign)
Supervisor/Guide

Principal

Vivekanand Education Society's
Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

Project Report Approval for B. E.

This project entitled **Canteen Management System using E-Wallet** by **Mr. Kaplesh Juvekar, Mr. Akash Katkar** and **Mr. Nitin Rohira** is approved for the degree **Bachelor of Engineering** in **Information Technology**.

Examiners

1 _____

2 _____

Supervisor

1 _____

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words and where ideas have been included. I have adequately referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not been fabricated or falsified any idea/data/fact in my submission. I understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been cited or from who proper permission has not been taken when needed.

_____

(Signature)

Kalpesh Juvekar (Roll No 30)

_____

(Signature)

Akash Katkar (Roll No 34)

_____

(Signature)

Nitin Rohira (Roll No 57)

Date:

Abstract

A large number of people visit the the college canteen everyday. During breaks there is a huge crowd in the canteen. Starting from the queue at the coupon counter to the serving counter a lot of time is spent waiting due to which the students get late for their lectures also.They often wish to have a way to considerably reduce or get rid of this waiting time.One solution to this problem is to have a computer system by which the order gets placed directly in the kitchen by displaying it on a monitor.This would avoid the time wasted at the serving counter when a server takes time to deliver previous orders before taking a new coupon and placing it in the kitchen. Also one can have a facility of placing orders in advance so that his/her order is kept ready for the particular break. This would almost remove the waiting time.

Keywords: e-wallet, canteen management system, PHP,javascript, HTML, Bootstrap

Table of Contents

Chapter 1

Introduction

1.1 Introduction

In the college canteen a lot of time is wasted in queues. The application is mainly beneficial for reducing the time wasted waiting in the queue by sending the orders directly to the kitchen, placing orders in advance, the prepaid wallet facility also saves time wasted in tendering change. This time can be used for any other good purpose. Wallet provided in the application is more secure and faster than many other applications.

1.2 Problem Statement

The aim of our project is to develop a system which can take orders at the counter and display them in the kitchen. We aim to accomplish this task by creating a web application for managing the canteen menu and orders. The web application would make use of HTML5, Javascript and BootStrap for frontend and PHP for backend.Appropriate security features shall be implemented to prevent attacks. For placing orders in advance we will create an android application as well as provide the same facility via web browser. The orders placed in advance will have an order id which shall be used to get the order directly at the serving counter. Payments can be made via Cash or Online.

1.3 Objectives

The chief objective of this system is to save time which is very precious in today's world.The functionalities like online ordering , prepaid wallet and order processing through displays fitted in the kitchen are vital in achieving this objective.The idea of replacing the traditional coupon system with a system which not only saves time but also is easy to use can be fulfilled by this system.So the goal of our project is to develop a fast and user friendly system which can be implemented in our own college canteen and then in other institutes as well.

Chapter 2

Literature Survey

2.1 Techniques Studied

2.1.1 RSA Asymmetric Encryption Scheme

RSA is an algorithm for public-key cryptography and is considered as one of the great advances in the field of public key cryptography. RSA security lies in the difficulty of factoring large number into prime factors. The inventor of RSA Algorithm suggests prime number that is used to generate the keys have more than 100 digits' length for security reasons[1].

2.1.2 ElGamal Asymmetric Encryption Scheme

Elgamal algorithm also is one of the public key cryptography algorithms. The security of this algorithm lies in the difficulty of calculating discrete logarithm.[1]

2.2 Journal Papers

Title	**Key generation algorithm design combination of RSA and ElGamal algorithm**
Summary	In this paper, the author proposes a key generation algorithm that is considered safe from the combination of the RSA and Elgamal algorithm. Based on the experiment that has been done, the computing time required for the proposed algorithm is relatively short, compared to the original RSA algorithm[1].

Title	**Computational Resources for mobile E-wallet System with observers**
Summary	This paper proposes an effective realization of e-wallet system, providing e-cash divisibility, off-line payment and deposit options as well as purchaser's anonymity against the vendor and satisfying general electronic payment security requirements of double spending prevention, unforgeability as well as unclonability of purchaser's identity. The latter is achieved through the implementation of bank's tamper resistant hardware agent, known as observer, provided with physically unclonable function, capable of yielding a unique unclonable code, in user's mobile device. The estimation of the necessary resources in the usage of the proposed mobile e- wallet system is presented in order to determine whether the proposal is worth the computation cost[2].

Title	**Canteen Food Ordering Android System**
Summary	To overcome of less-time issue, the proposed system in this paper gives an android app for food ordering from canteen in house, which will save the time by avoiding long queue, waiting for long time to get their order and inconvenience will vanish.

Title	**Cloud Based Canteen Management System**
Summary	This paper proposes a system to make the whole process easier for both the canteen and for the customers by automating the process on Cloud. Hosting on cloud is cost effective than owning individual components. This proposed system bridges the gap between canteen and its usage. Also, to reduce the queues another method to order and pay is through mobile

	based app where deductions are carried out directly from the customer's account. Account can be recharged through online transfer and e-wallets. Both the web and mobile applications will be hosted on cloud.

Chapter 3

Requirements and Analysis

3.1 Functional requirements

- **Place Order:**

There are two ways of placing an order, via online ordering feature or through the canteen counter. The online ordering feature shall be available to users who login only and have a valid balance in their E-wallet. The counter ordering facility shall be available only to administrator through administrator login.

- **Make Payment:**

There are two modes of payment, via E-wallet and Cash. Online orders can be paid only through E-wallet. The E-wallet payment can be used at the counter also. Cash payment option is also present at counter. Since the E-wallet is prepaid it needs to be recharged at the counter by paying cash to be able to use. Recharge function is available in administrator login**.**

- **Display Order:**

The items of a placed order shall be displayed on the screens in the kitchen which indicate the cooks to prepare the items. When the order is delivered its status is updated .When the status of the order is updated to “COMPLETE” it goes off the screen.

3.2 Non-functional requirements

- **Reliability:**

The developed application should be such that it can be implemented in real life and facilitate a smooth functioning of the canteen along with order placing and processing.

- **Security:**

The E-wallet payment should be secure enough to facilitate safe payment. Necessary encryption algorithms shall be deployed to build a secure e-wallet.

3.3 Hardware and Software requirements

3.3.1 Hardware requirements

- **Personal Computer**

  Specifications:

1. Processor : 2 GHz
2. RAM : 2 GB
3. Keyboard
4. Mouse

- **Computer Displays - 2**

  For the purpose of displaying order in kitchen.

- **Internet Connection**

  Speed : 1 MBPS

3.3.2 Software Requirements

- **For Development**

1. Java
2. MySQL Database server
3. Apache Web Server
4. Angular JS
5. PHP
6. Bootstrap for Website

- **For User**

1. Google Chrome Browser to access web application from Desktop or Mobile.

3.4 Analysis

We have analysed two encryption techniques for providing security to e-wallet, RSA asymmetric key encryption and ElGamal asymmetric key encryption. A comparative study between the two led to selection of ElGamal asymmetric key encryption technique. The advantages of using ElGamal asymmetric key encryption algorithm are :

1. One of the strength of ElGamal is its non-determinism-encrypting the same plaintext multiple times will result in different ciphertexts, since a random k is chosen each time.
2. El-Gamal encryption is used in the free GNU privacy Guard Software,recent versions of PGP,and other cryptosystems.

Chapter 4

System Design

4.1 Architecture

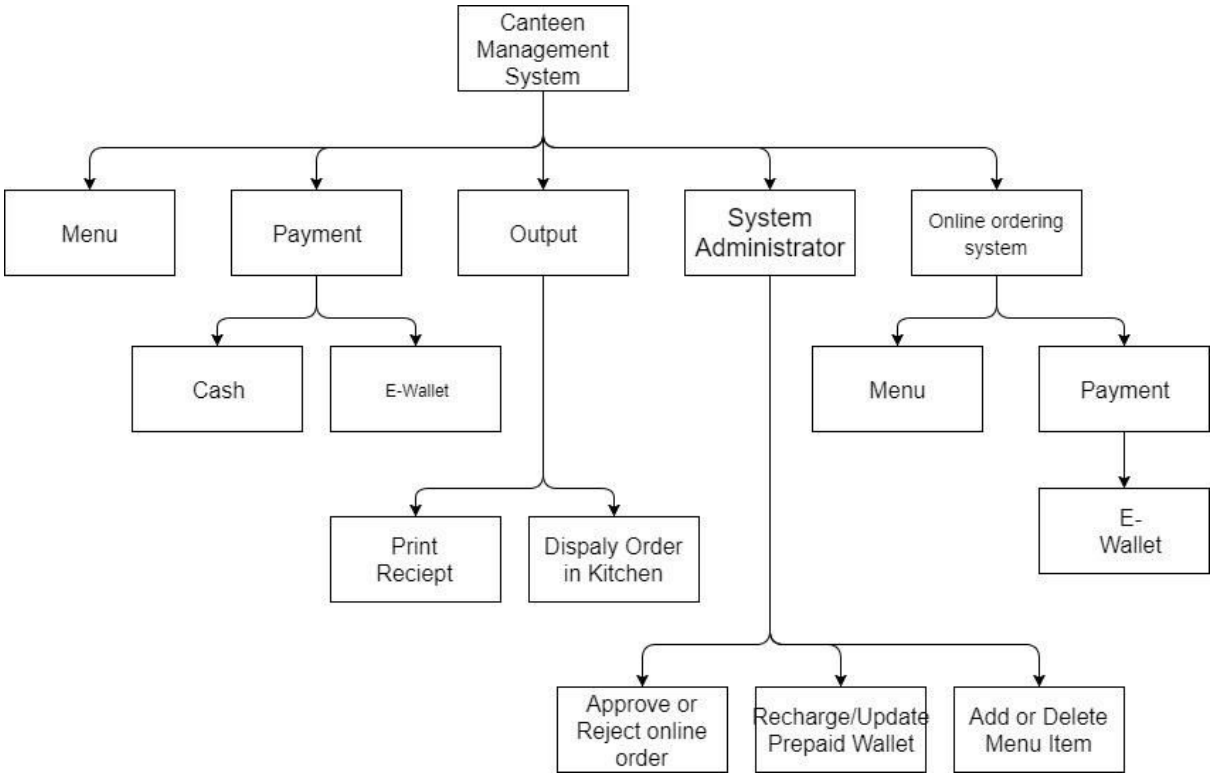


Fig. 4.1 Architecture

4.2 Detailed Design

4.2.1 Flowcharts

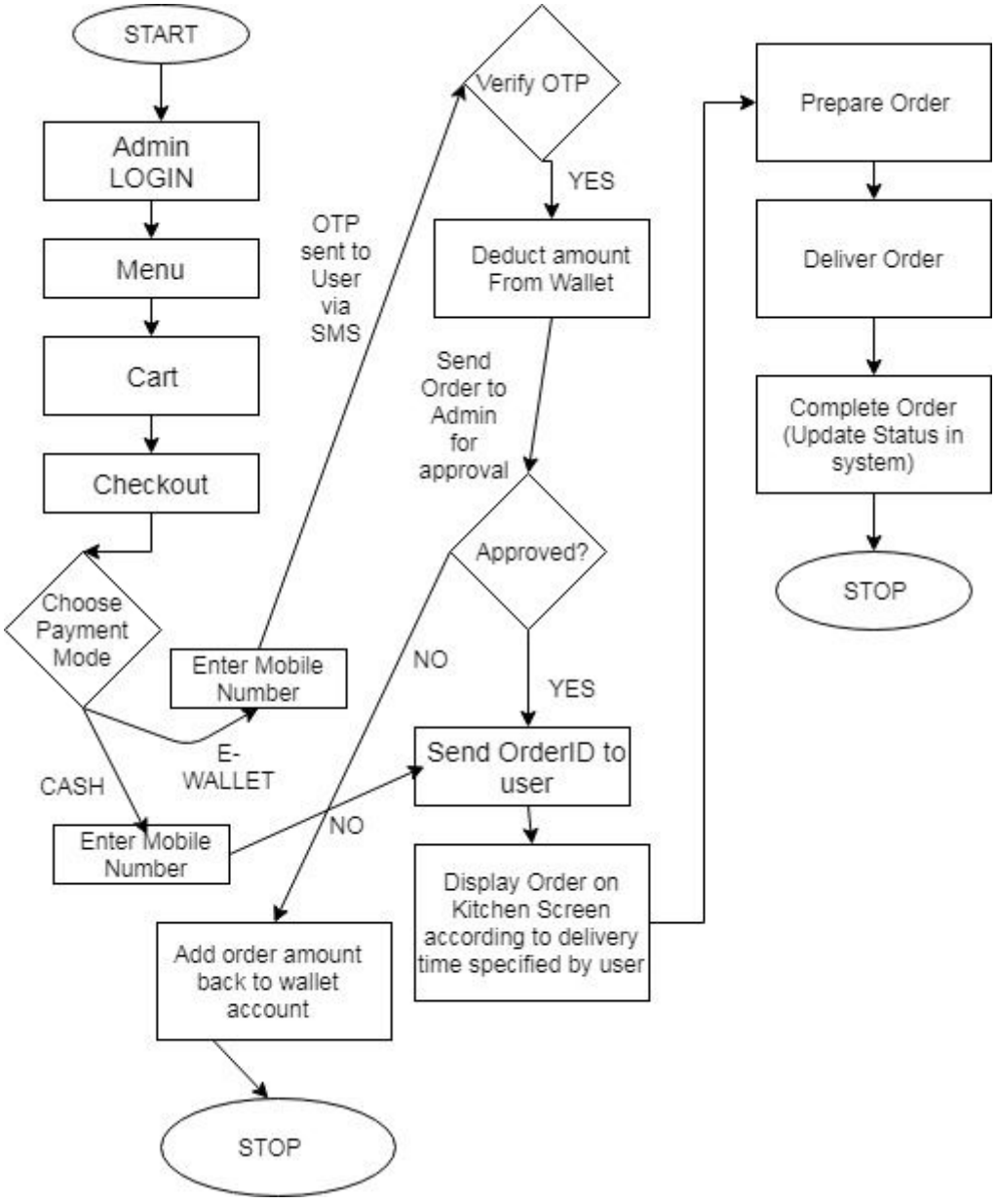


Figure 4.2.1.1 : Counter ordering system's working

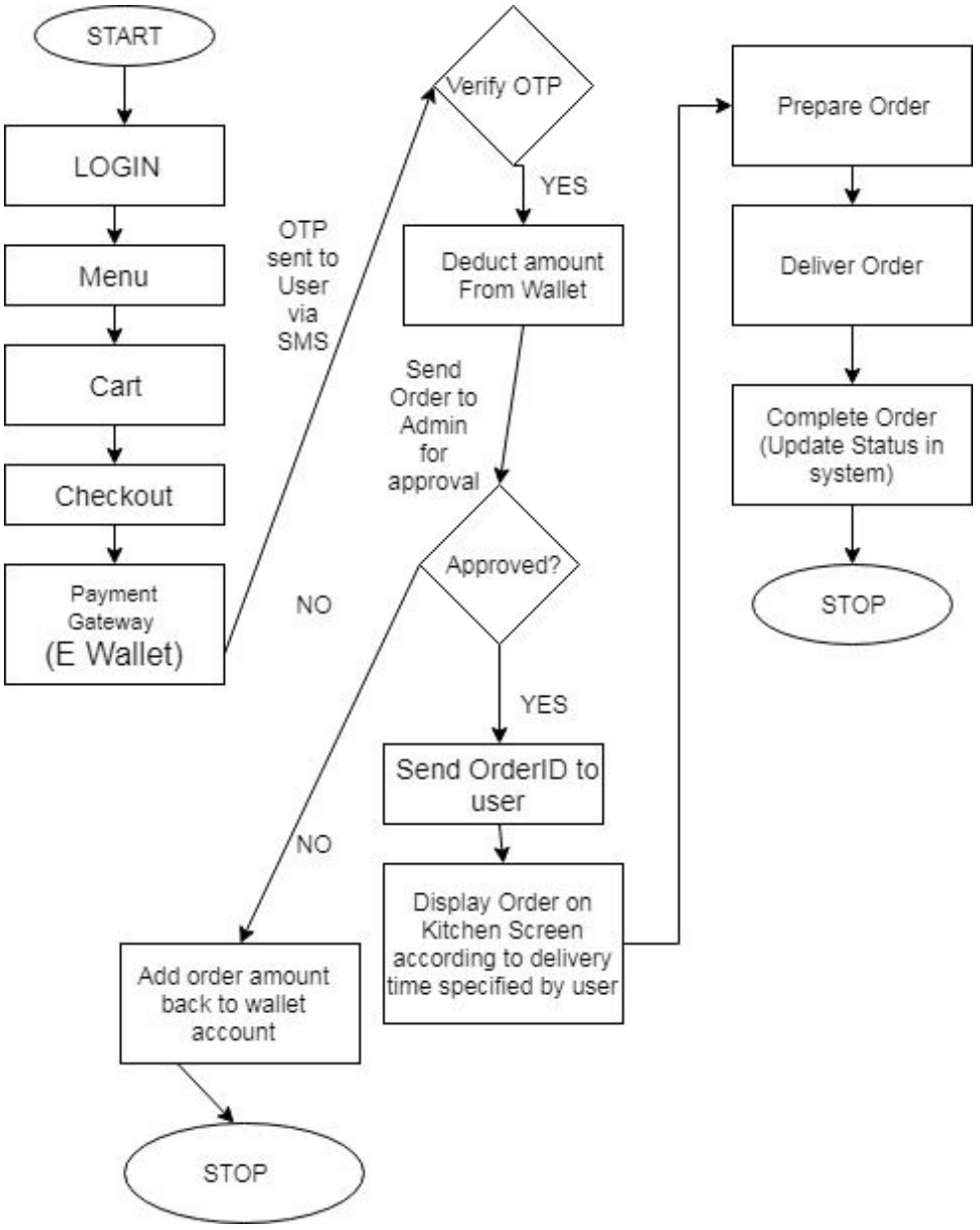


Figure 4.2.1.2 : Online ordering system's working

4.2.2 Activity Diagram

Online ordering

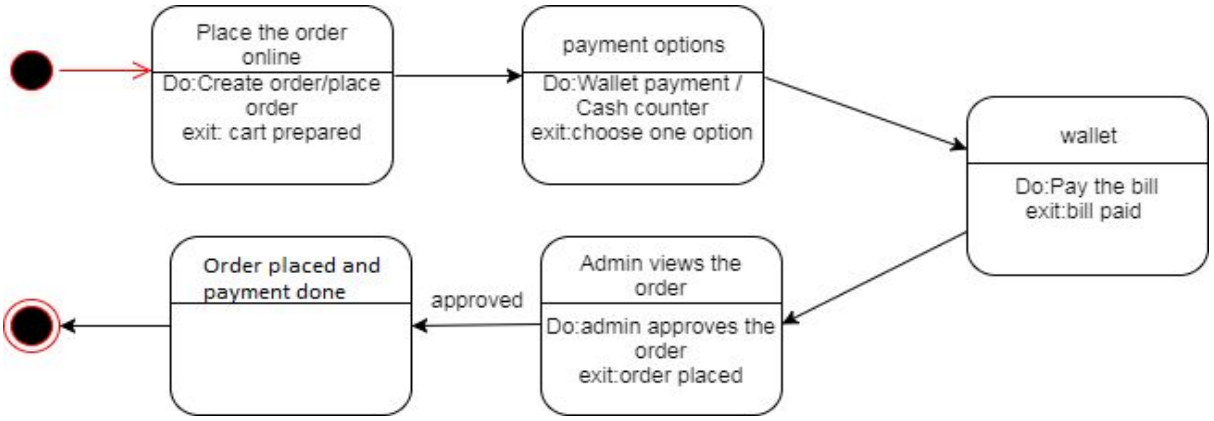


Fig. 4.2.2.1 Online Ordering activity diagram

Counter ordering

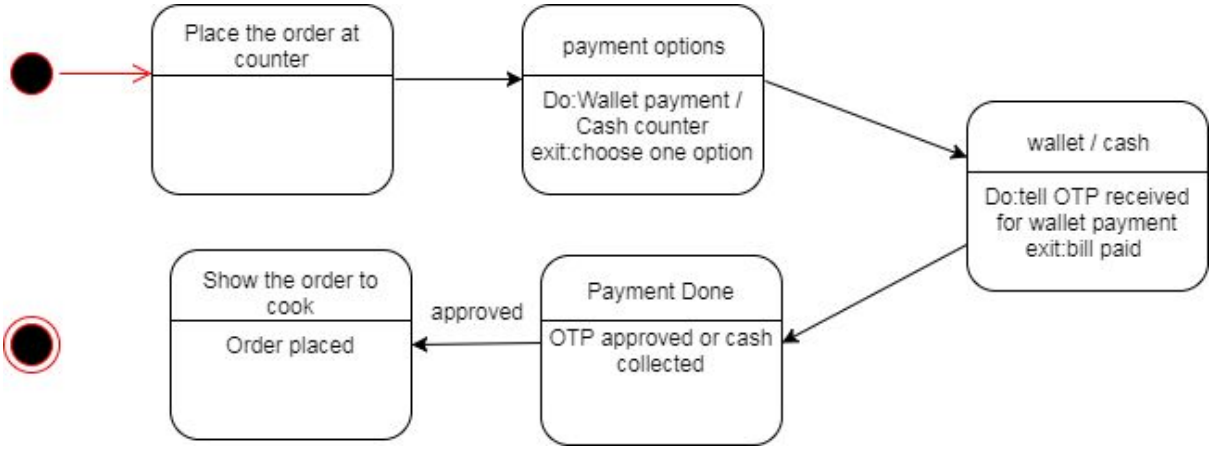


Fig. 4.2.2.2 Counter Ordering activity diagram

Wallet working

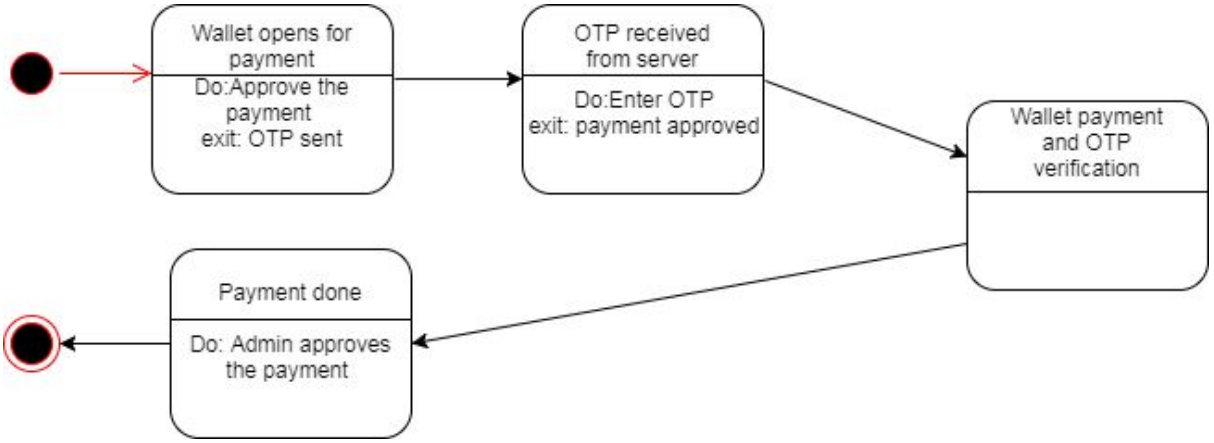
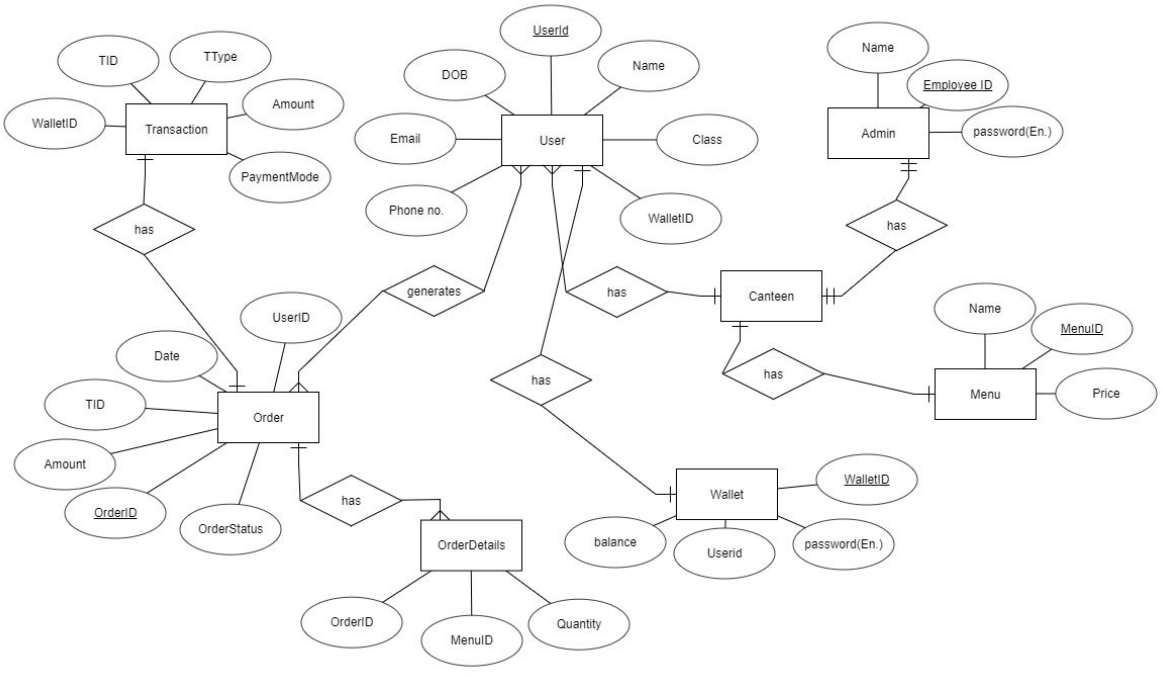


Fig. 4.2.2.3 Wallet working activity diagram

4.2.3 E-R Diagram



Chapter 5

Implementation

5.1 Application Code

The front-end of this project is developed using HTML5, CSS, Javascript and Bootstrap framework. The back-end is developed using JSP. The important code snippets are mentioned below.

5.1.1 Elgamal Encryption/Decryption Scheme

KeyStore.java

```
package CMS;

import java.math.*;
import java.util.*;
import java.security.*;

public class KeyStore
        {
        public static BigInteger p, g, a, x;
      static Random sc = new SecureRandom();
      KeyStore()
      {
        generator();
      }
        private void generator()
                {
                        x = new BigInteger("123456789012345678901234567890123456");//private Key
                                                p                =                new
BigInteger("188956112148988884942120127948847824448089575134143599249183801612542247860046258048096398654193536033360783245243591716443840475069172667318567692757104400402598303926718667543261056129481020301996914579635918384177705616755904553804324922020078768666729684399379280888913448931351714347558413966942112305056348620173175173097525264314921534364137977623312008890 5
```

```
4046137021852175096285215533445462681316890075252003721626077404546509694277535358565054387220108652722893393973963754873066759960248402159955546466853923518255203339702729778384384857848867143145305816376862070576983900234031338178839367493708157818891");
                                                    g = new BigInteger("9558086680230558267451310284035787725170836817641592255651102069890145642982705224398688200208535944171935437408484762228571663798895409680964747403027164889397125142534491121443974872952187532875757237005899841002580118249731221786849979660319454885380281815049897092175118910293469293190096906258338976054942808272711340174786651851459889021392623575753524302308711373208910538875903391104475158332872924305795871669945808592209348450107162187015109972251467193514317213525337496920694253303793490134689588531550592640041506005303693280654269665139568018459399892795692206829384817883663307281807067947086571658873");
                                a = g.modPow(x, p);
                        }
          public BigInteger getPublicKey()
             {
                return p;
             }
          public BigInteger geta()
             {
                return a;
             }
       public BigInteger getg()
             {
                return g;
             }
          public BigInteger getPrivateKey()
             {
                return x;
             }
          }
```

Encrypt.java

```
package CMS;

import java.math.*;
import java.util.*;
import java.security.*;

public class Encrypt
{
         public static BigInteger a,p,g,k,X,E,D;
           public void encrypt(String s)
           {
              final long serialVersionUID;
```

```
            serialVersionUID = 1L;
            Random sc=new SecureRandom();
            KeyStore key=new KeyStore();
            p = key.getPublicKey();
            a = key.geta();
            g = key.getg();
            X = new BigInteger(s);
            k = new BigInteger(2048, sc);
            E = X.multiply(a.modPow(k, p)).mod(p);
            D = g.modPow(k, p);

        }
        public BigInteger getE()
        {
       return E;
        }
        public BigInteger getD()
        {
       return D;
        }
}
```

Decrypt.java

```
package CMS;

import java.math.*;

public class Decrypt
{
      public static BigInteger a,p,g,k,x;

   /**
    *
    * @param D
    * @param E
    * @return
    */
   public BigInteger decrypt(BigInteger D,BigInteger E)
      {
      KeyStore key=new KeyStore();
      x=key.getPrivateKey();
      p=key.getPublicKey();
      BigInteger crmodp = D.modPow(x, p);
      BigInteger d = crmodp.modInverse(p);
      BigInteger ad = d.multiply(E).mod(p);

      return ad;
```

```
    }
}
```

5.1.2 SMS GATEWAY HUB API Implementation

sendSMS.java

```
package CMS;

import java.net.*;

public class sendSMS {
 static String mn;
 static int otp;
 static String requestUrl;
 public sendSMS(String a,int b){
    mn=a.trim();
    otp=b;
 }
public static void sendOTP() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen App. Your OTP is "+otp+".";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=Y
OURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20th
is%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();
```

```
System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}

public static void sendBalUpdate() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen App. Amount of Rs."+otp+" deducted
from your E-wallet.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=Y
OURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20th
is%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
```

```
System.out.println(ex.getMessage());

}
}
public static void sendOrder() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen. Your ORDERID "+otp+" is successfully placed.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=YOURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20this%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}
 public static void sendConfirmation() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
```

```
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Welcome to V.E.S. Canteen App "+mn+". You have been succesfully registered.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=YOURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20this%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
 }
 public static void sendAddConfirmation(String b) {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Amount of Rs. "+otp+" added to your account.Updated Balance is Rs."+b+" .";
```

```
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=Y
OURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20th
is%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}
 public static void sendRefund(String b) {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Amount of Rs. "+otp+" refunded to your account.Updated Balance is
Rs."+b+" .";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=Y
OURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20th
is%20is%20testing%20message&route=
```

```
requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}
 public static void AcceptOrder() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen. Your ORDERID "+otp+" is accepted.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=YOURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20this%20is%20testing%20message&route=



requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
```

```
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}
 public static void DeclineOrder() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen. Sorry!Your ORDERID "+otp+" is declined due to some issues.Refund will be processed soon.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=YOURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20this%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();
```

```
System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());

}
}
  public static void CompleteOrder() {
try {
String apikey = "sUpr4MdfDUedA9LW7GIcJA";
String senderid = "TESTIN";
String channel = "2";
String DCS = "0";
String flashsms = "0";
String mobile = "91"+mn;
String message = "Thank you for using V.E.S. Canteen.Your order with ORDERID "+otp+"
is deivered.";
String route = "13";

https://www.smsgatewayhub.com/api/mt/SendSMS?APIKey=YOURAPIKEY&senderid=Y
OURSENDERID&channel=2&DCS=0&flashsms=0&number=91XXXXX&text=hello%20th
is%20is%20testing%20message&route=


requestUrl = "https://www.smsgatewayhub.com/api/mt/SendSMS?" +
"APIKey=" + URLEncoder.encode(apikey, "UTF-8") +
"&senderid=" + URLEncoder.encode(senderid, "UTF-8") +
"&channel=" + URLEncoder.encode(channel, "UTF-8") +
"&DCS=" + URLEncoder.encode(DCS, "UTF-8") +
"&flashsms=" + URLEncoder.encode(flashsms, "UTF-8") +
"&number=" + URLEncoder.encode(mobile, "UTF-8") +
"&text=" + URLEncoder.encode(message, "UTF-8") +
"&route=" + URLEncoder.encode(route, "UTF-8");



URL url = new URL(requestUrl);
HttpURLConnection uc = (HttpURLConnection)url.openConnection();

System.out.println(uc.getResponseMessage());

uc.disconnect();

} catch(Exception ex) {
System.out.println(ex.getMessage());
```

```
}
}
}
```

5.1.3 Amazon MySQL DB connection using JDBC drivers

MyDb.java

```
package CMS;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class MyDb {
   Connection con;
   public Connection getCon() throws SQLException{
      try {
         Class.forName("com.mysql.jdbc.Driver");
                                                                    con = (Connection)
DriverManager.getConnection("jdbc:mysql://cms.caxkntezqqs2.us-west-2.rds.amazonaws.co
m:3306/cms","vescanteen","vescanteen");
      } catch (ClassNotFoundException ex) {
         Logger.getLogger(MyDb.class.getName()).log(Level.SEVERE, null, ex);
      }

      return con;
   }
}
```

5.1.4 API for client side encryption

KeyRequest.jsp

```
<%@page import="CMS.MyDb"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.math.BigInteger"%>
<%@page import="CMS.Keyclient"%>
<%@page import="java.util.Random"%>
<%@page import="java.sql.*"%>
<%
        BigInteger y1,y2;
        Keyclient key=new Keyclient();
        key.encrypt();
        y1=key.getkeyy1();
```

```
        y2=key.getkeyy2();
        Random rand = new Random();
        int n = rand.nextInt(9000) + 1;
        session.setAttribute("amountrand",n);
        MyDb db=new MyDb();
        Connection con=db.getCon();
        String q1="Select * from amount where amountid='"+n+"'";
        Statement st1=con.createStatement();
        ResultSet rs1=st1.executeQuery(q1);
        if(!rs1.next())
        {
           String updateQuery = "insert into amount(y1,amountid) values('"+y1+"','"+n+"')";
              PreparedStatement ps1 = con.prepareStatement(updateQuery);
              ps1.executeUpdate();
           }
        else
        {
        String updateQuery = "update amount set y1='"+y1+"' where amountid='"+n+"'";
              PreparedStatement ps1 = con.prepareStatement(updateQuery);
              ps1.executeUpdate();
           }
        response.setContentType("text/plain");
        response.setCharacterEncoding("UTF-8");
        response.getWriter().write(String.valueOf(y2));
%>
```

Encryption:

KeyClient.java

```
package CMS;
import java.math.*;
import java.util.*;
import java.security.*;
   public class Keyclient
{
        public static BigInteger a,p,g,k,X,y2,y1;
         public void encrypt()
         {
            final long serialVersionUID;
            serialVersionUID = 1L;
            Random sc=new SecureRandom();
            KeyStore key=new KeyStore();
            p = key.getPublicKey();
            a = key.geta();
            g = key.getg();
            k = new BigInteger(2048, sc);
            y2 = a.modPow(k, p).mod(p);
            y1 = g.modPow(k, p);
```

```
    }
    public BigInteger getkeyy2()
    {
    return y2;
    }
    public BigInteger getkeyy1()
    {
    return y1;
    }
}
```

5.1.5 Client side javaScript function for Ajax call and calculation of encrypted value

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
var big=""+this.responseText.trim()+"";
var y2=""+bigInt(big).multiply(total).toString()+"";
document.getElementById("amount").value = y2;
document.getElementById("submitorder").disabled=false;}};
xhttp.open("POST", "keyrequest.jsp", true);
xhttp.send();
```

5.2 Application UI

5.2.1 Client Application Screenshots

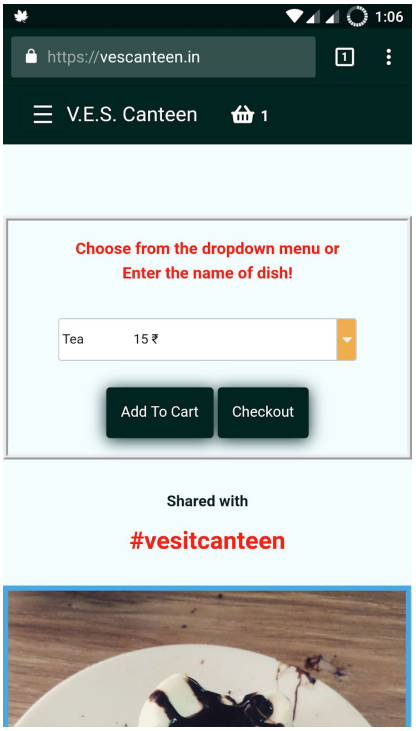


Figure 5.2.1.1 Client Home

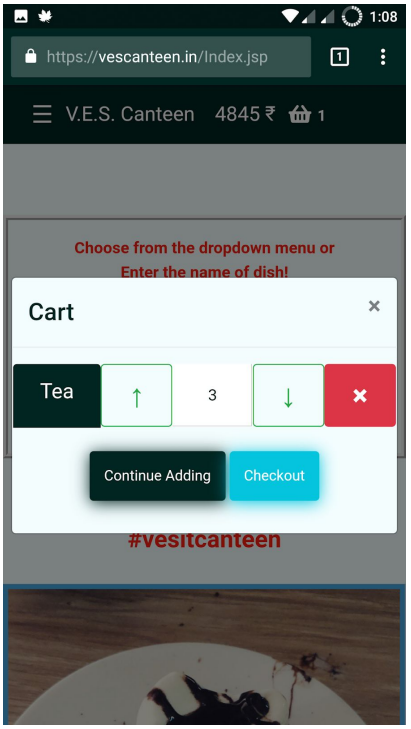


Figure 5.2.1.2 Cart

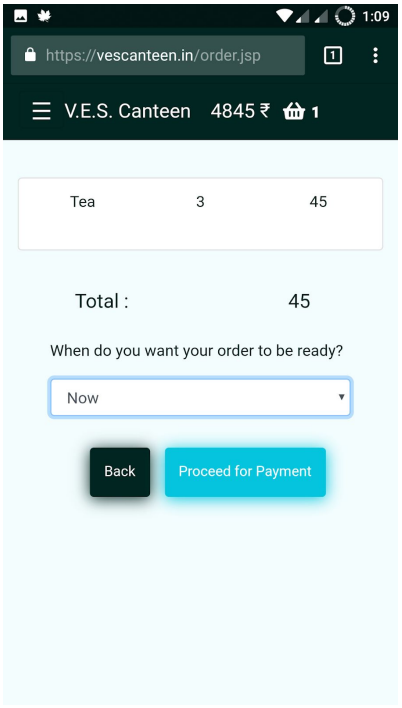


Figure 5.2.1.3 Checkout

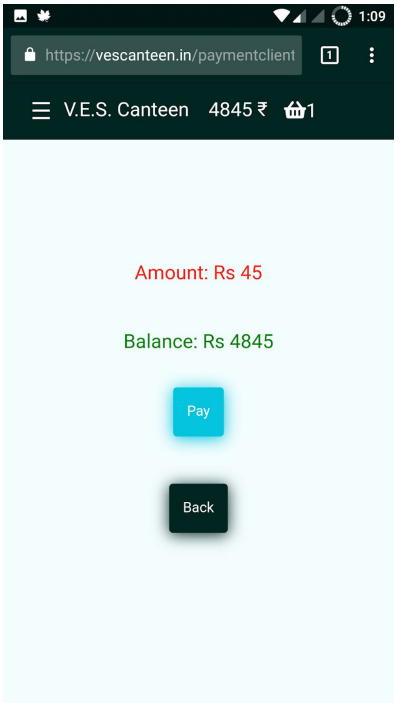


Figure 5.2.1.4 Wallet Pay

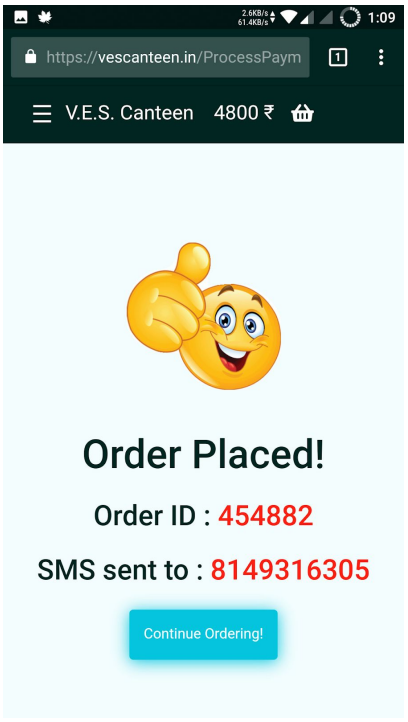


Figure 5.2.1.5 Order Placed

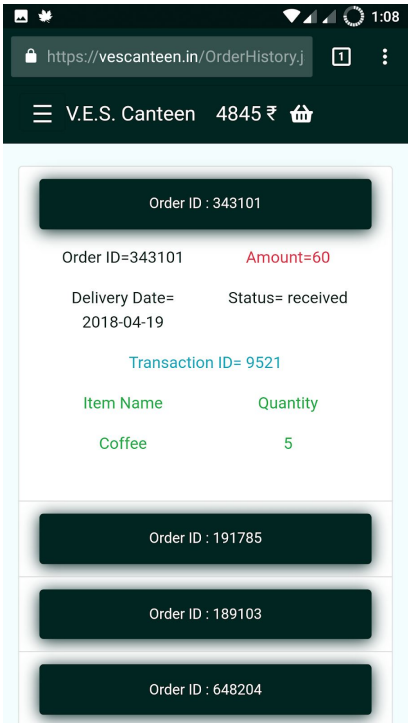


Figure 5.2.1.6 Order Details

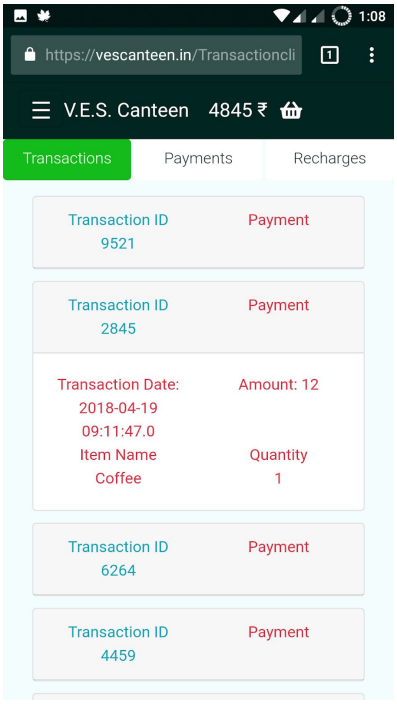


Figure 5.2.1.7 Transaction Details

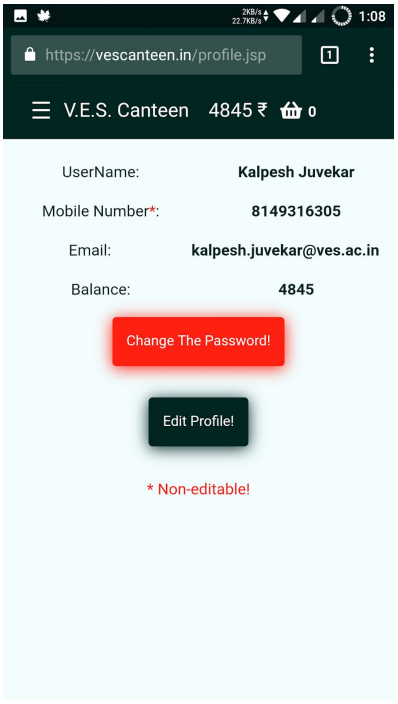


Figure 5.2.1.8 Profile

5.2.2 Client SMS Screenshots

Figure 5.2.2.1 OTP Message

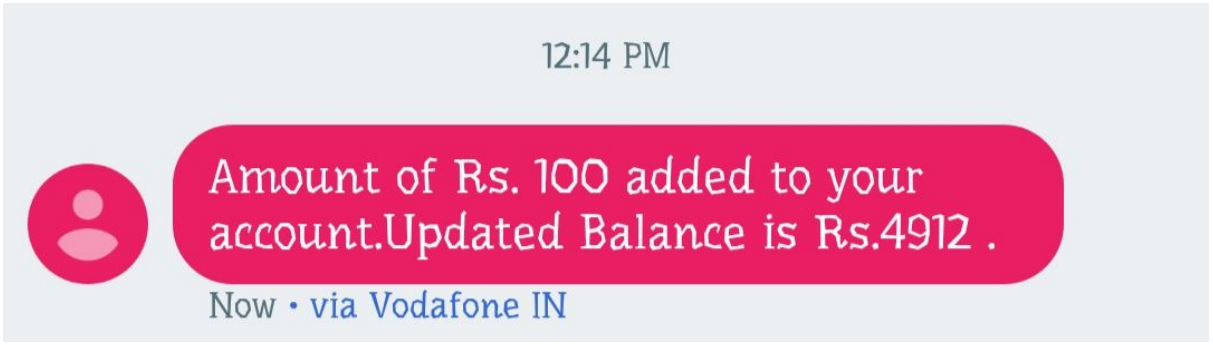


Figure 5.2.2.2 Wallet Recharge Message

Figure 5.2.2.3 OrderID Message

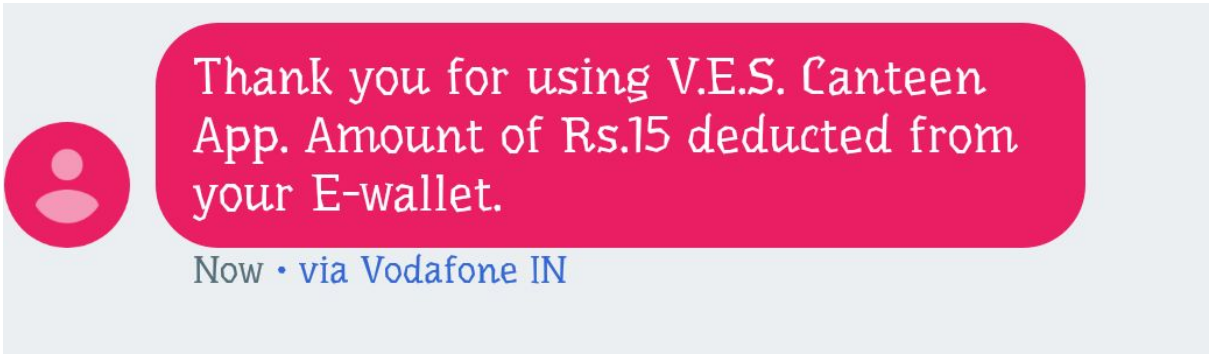


Figure 5.2.2.4 Wallet Payment Message

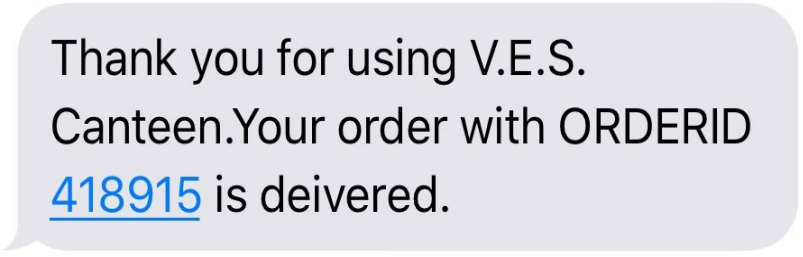


Figure 5.2.2.5 Order Delivered Message

Figure 5.2.2.6 Online Order Accepted Message

Figure 5.2.2.7 Online Order Declined Message

Figure 5.2.2.8 E-wallet Refund Message

5.2.3 Administrator Application Screenshots

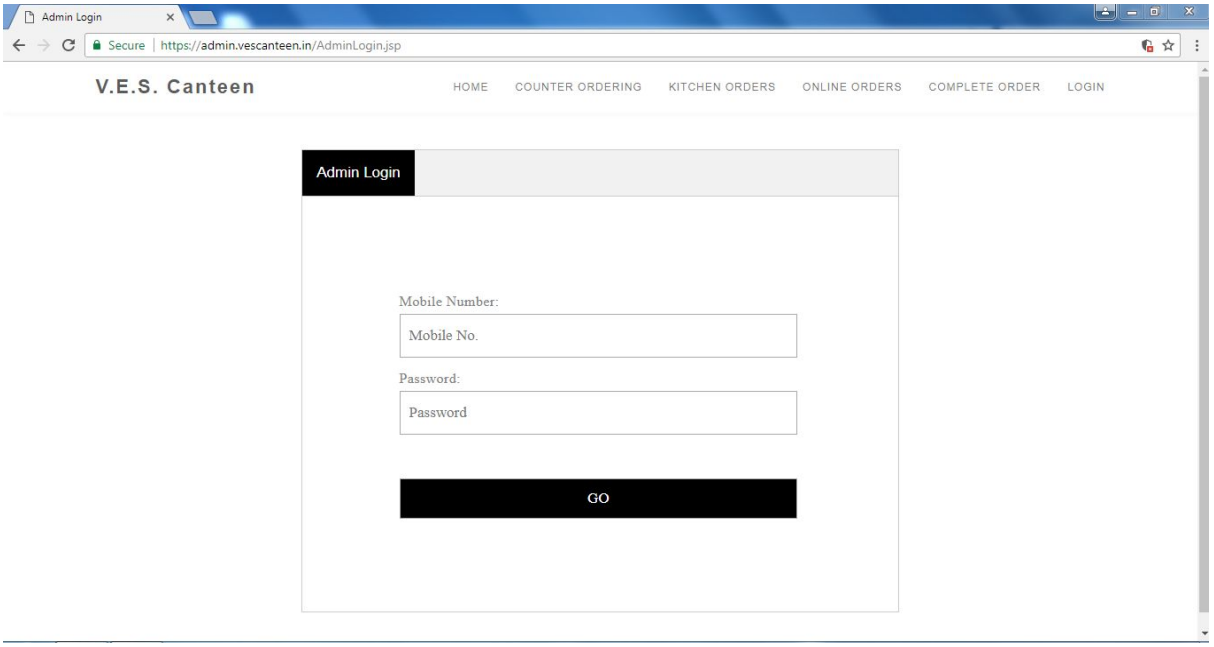


Figure 5.2.3.1 Administrator Login

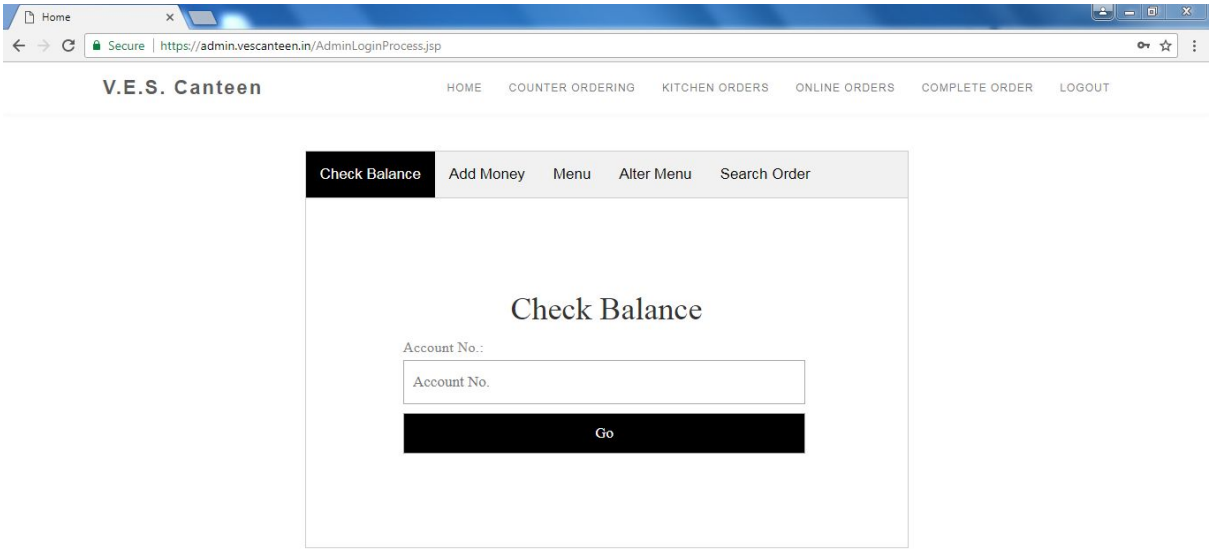


Figure 5.2.3.2 Administrator Home/Check Balance

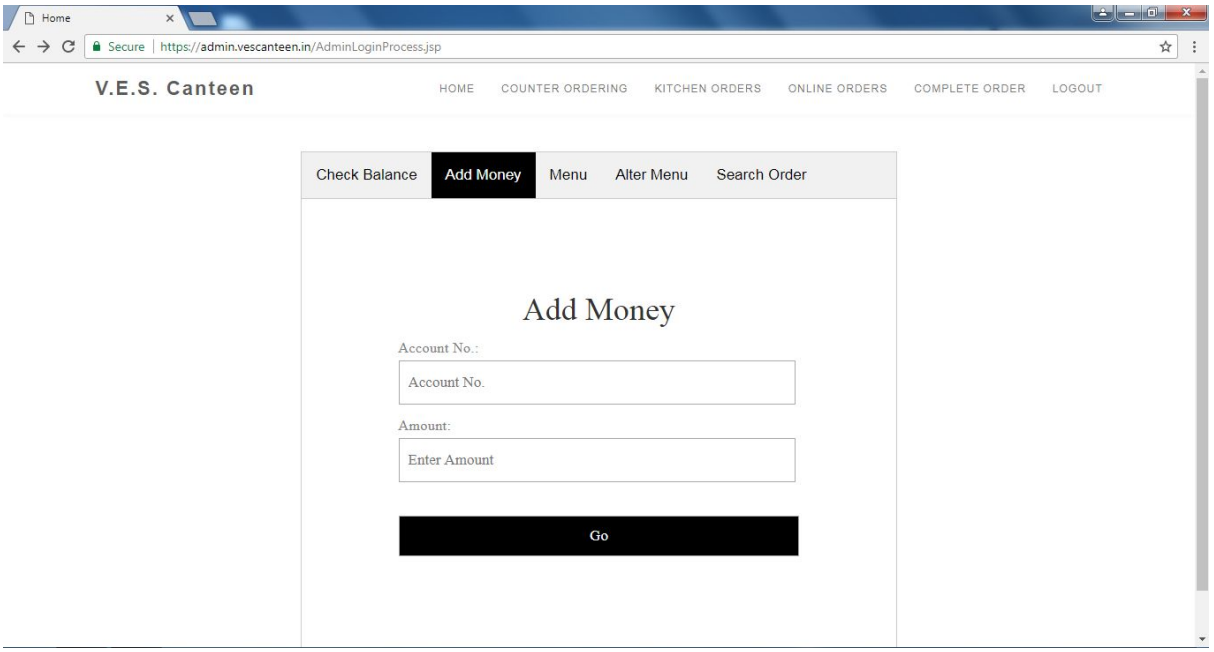


Figure 5.2.3.3 Add Money

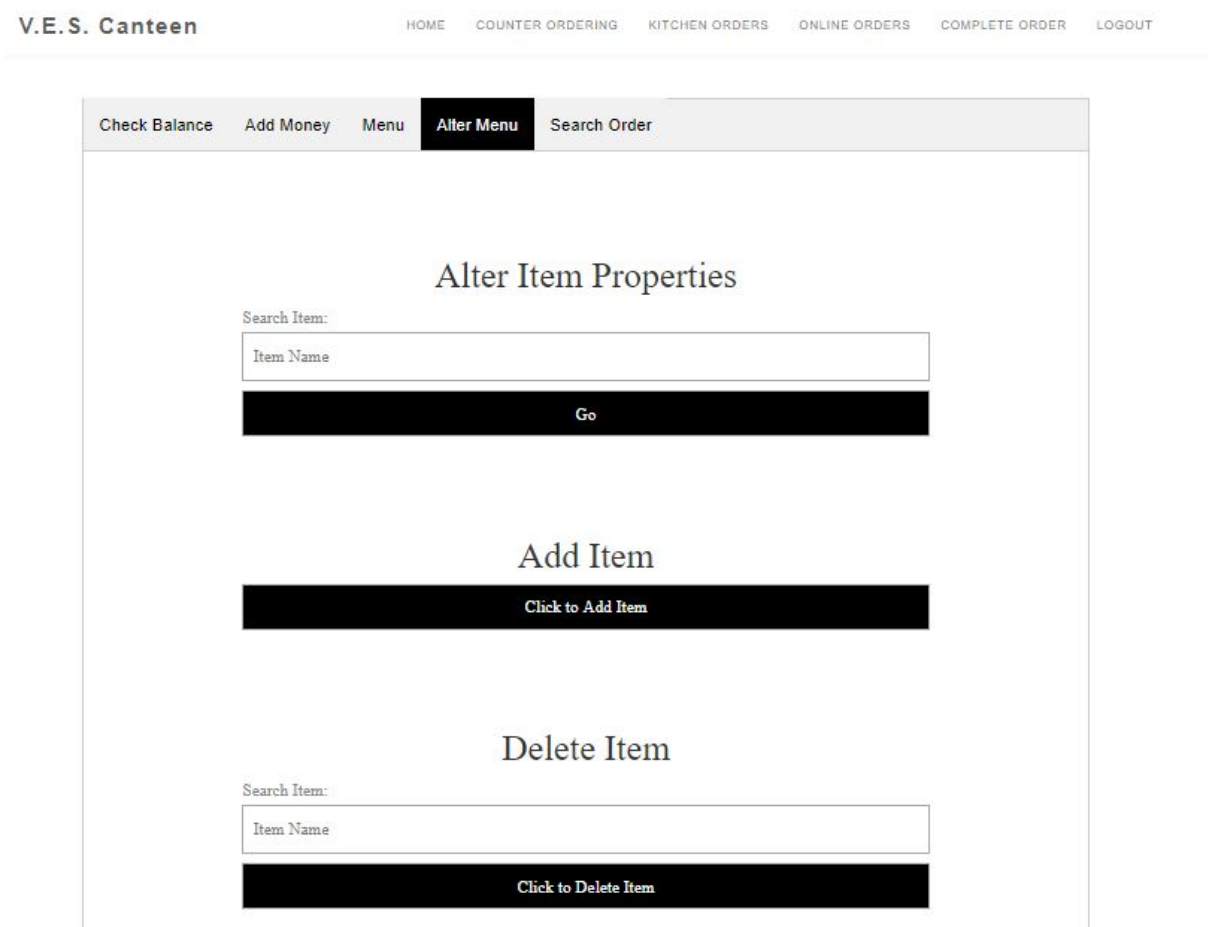


Figure 5.2.3.4 Alter Menu

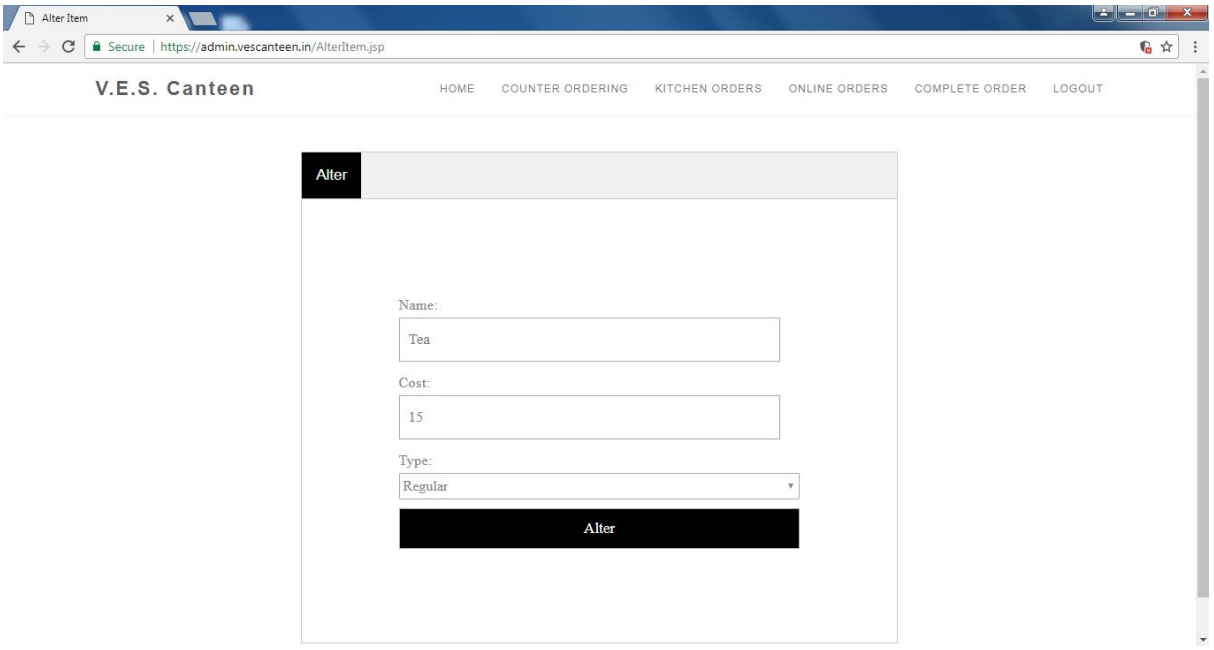


Figure 5.2.3.5 Alter Item

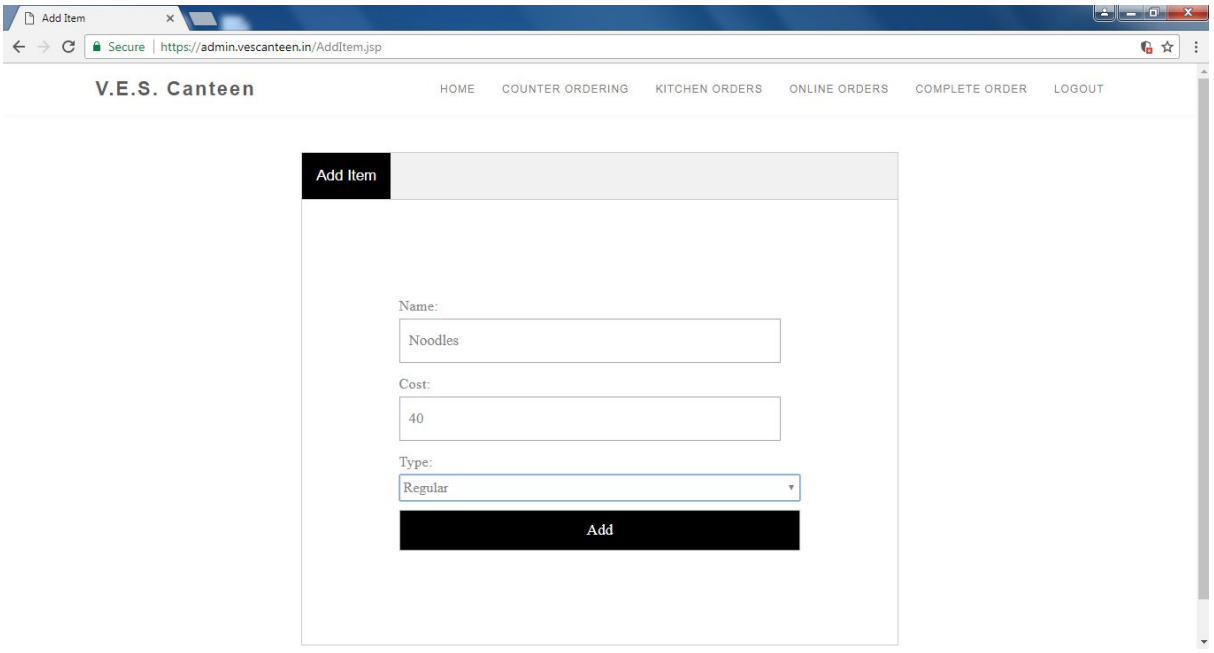


Figure 5.2.3.6 Add Item

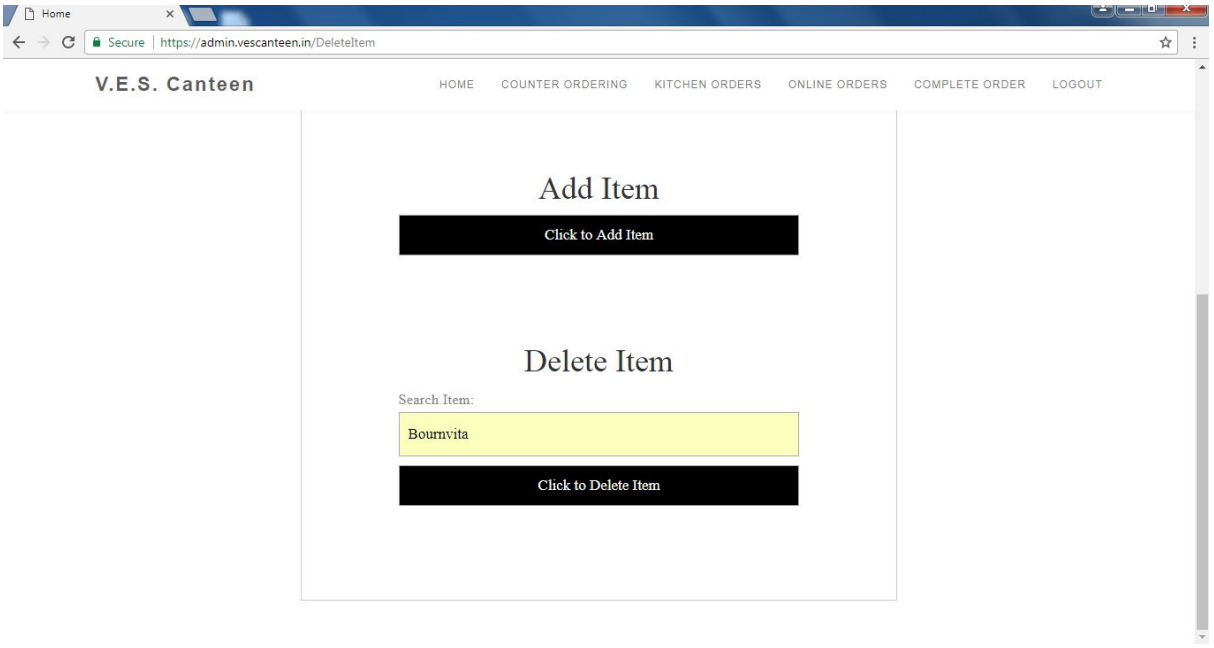


Figure 5.2.3.7 Delete Item

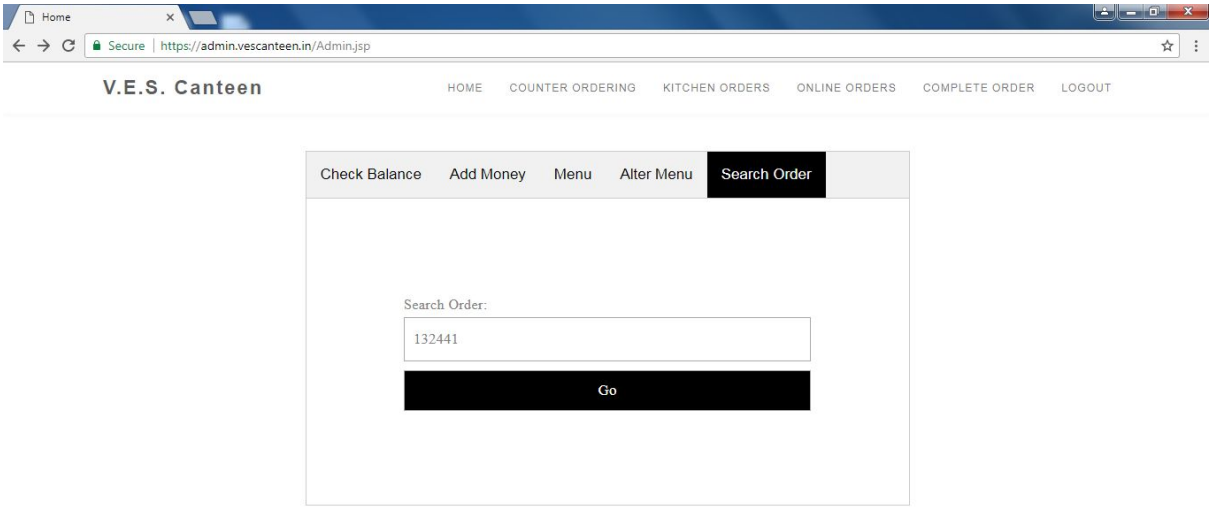


Figure 5.2.3.8 Search Order

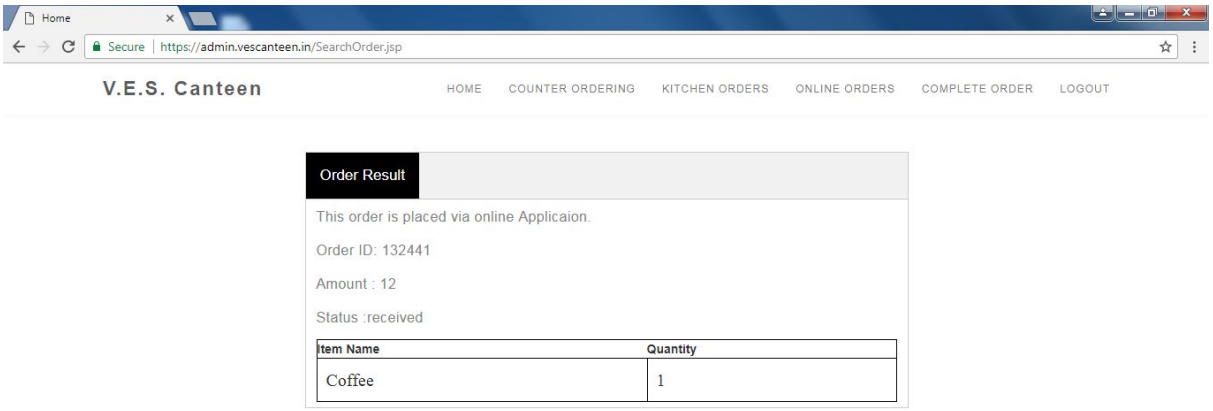


Figure 5.2.3.9 Search Order - Result

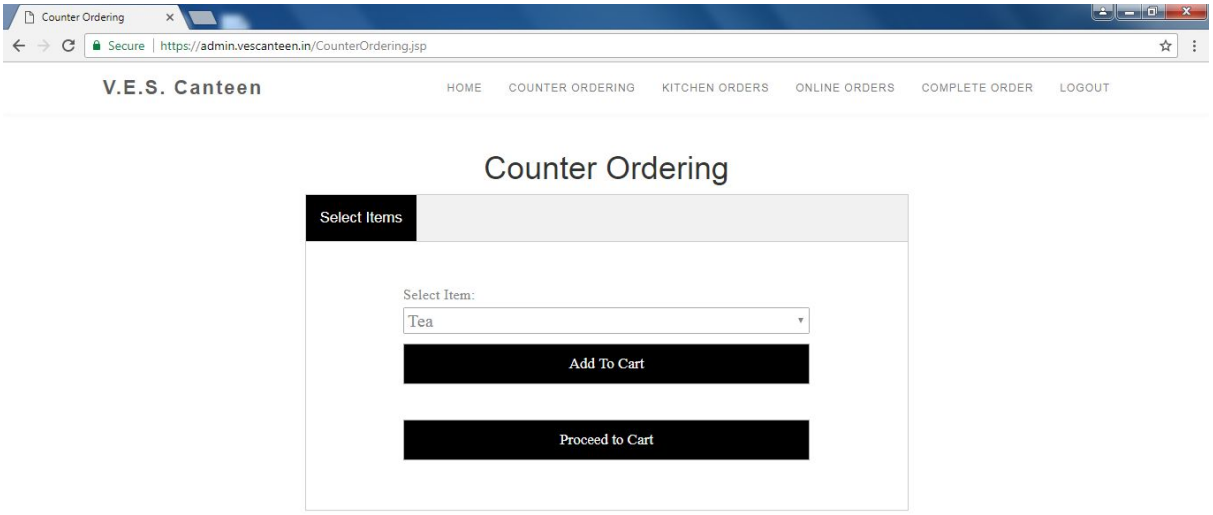


Figure 5.2.3.10 Counter Ordering Menu

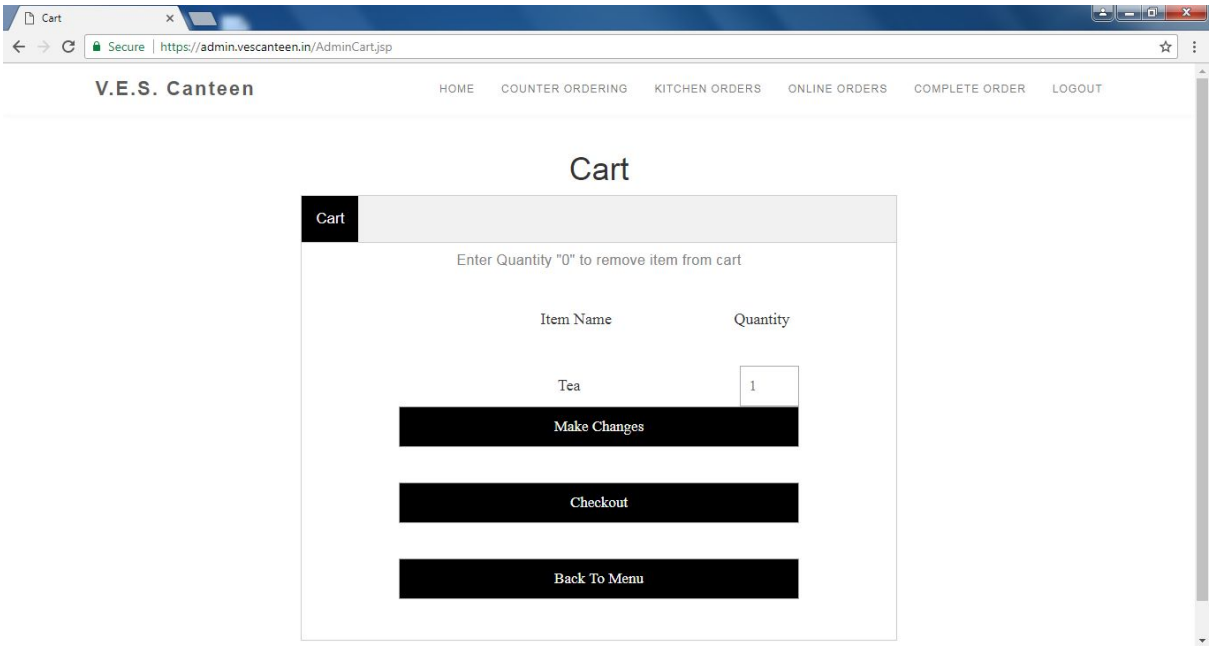


Figure 5.2.3.11 Counter Ordering Cart

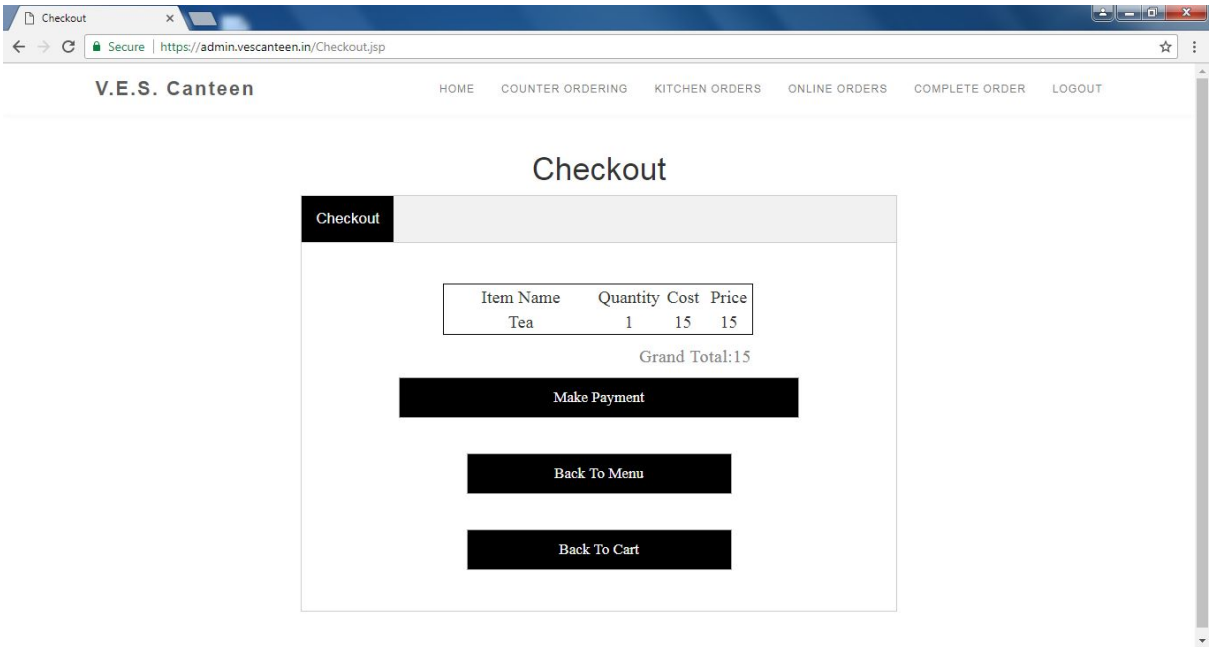


Figure 5.2.3.12 Counter Ordering Checkout

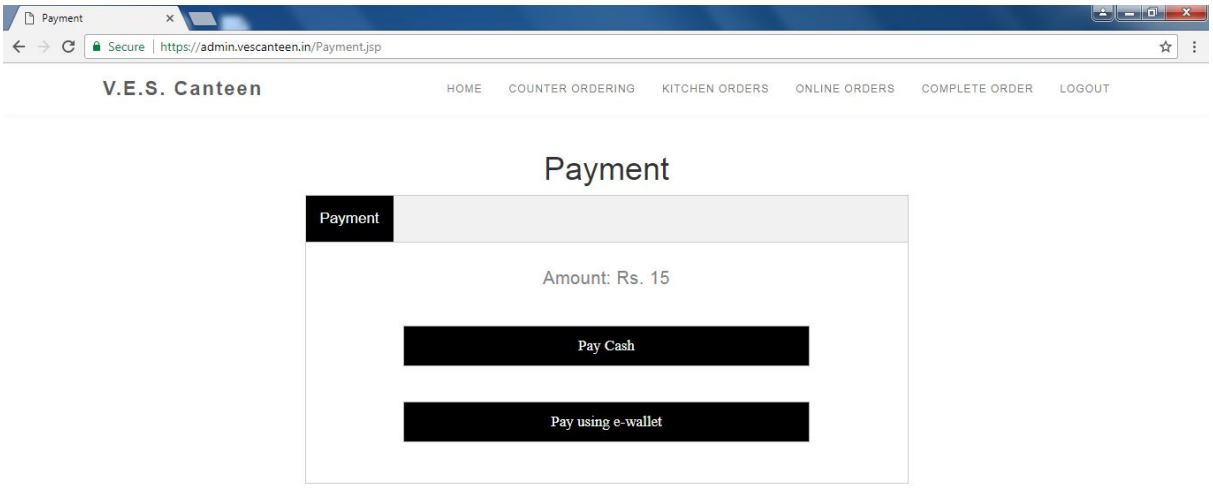


Figure 5.2.3.13 Counter Ordering Payment

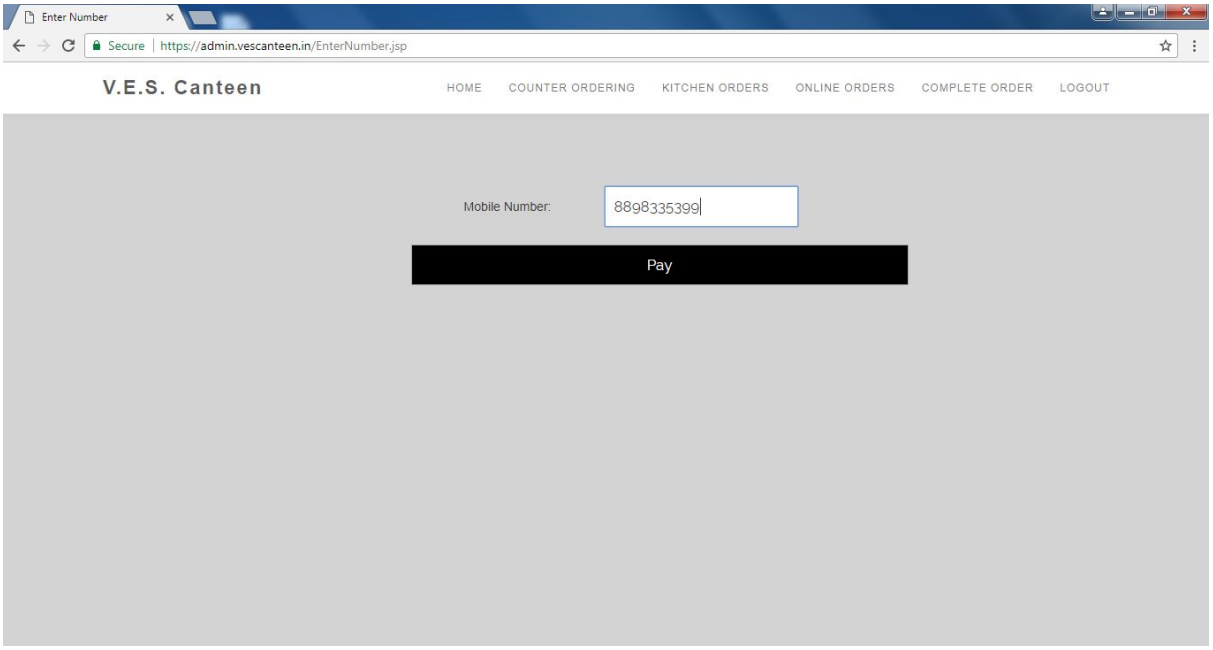


Figure 5.2.3.14 Counter Ordering Cash Payment

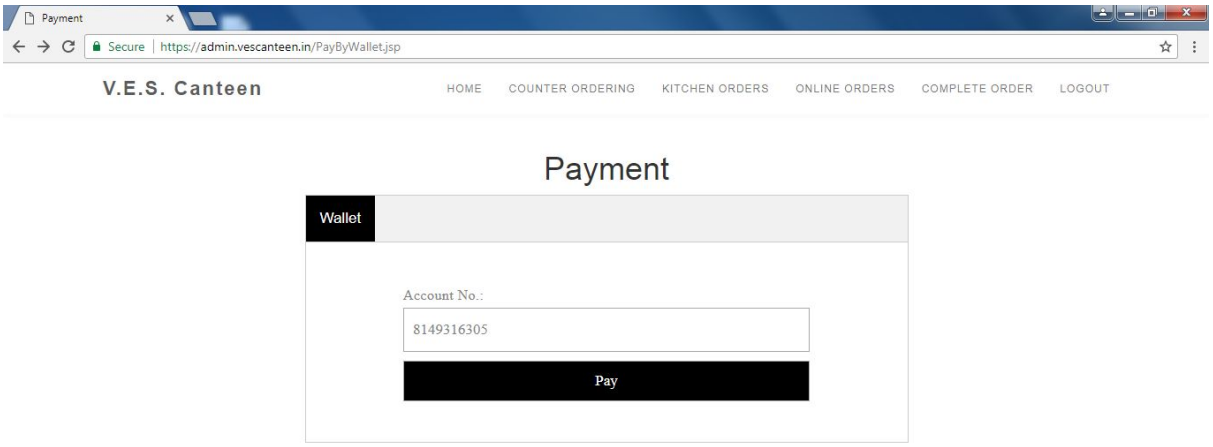


Figure 5.2.3.15 Counter Ordering E-wallet Payment

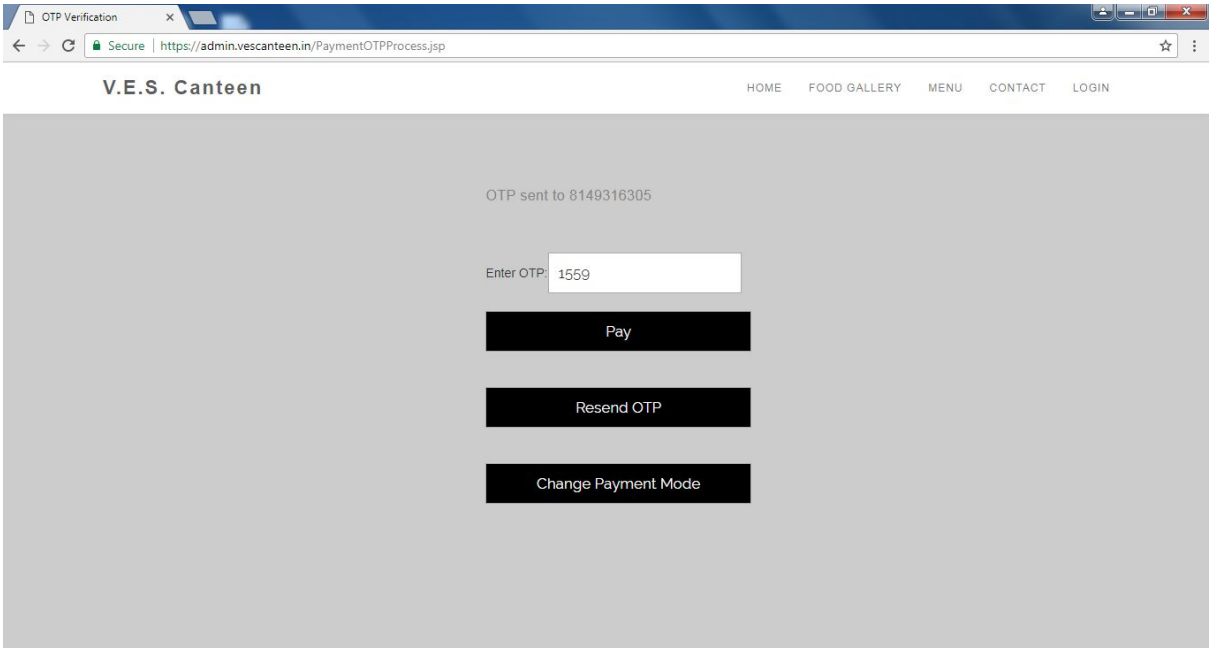


Figure 5.2.3.16 Counter Ordering E-wallet Payment OTP

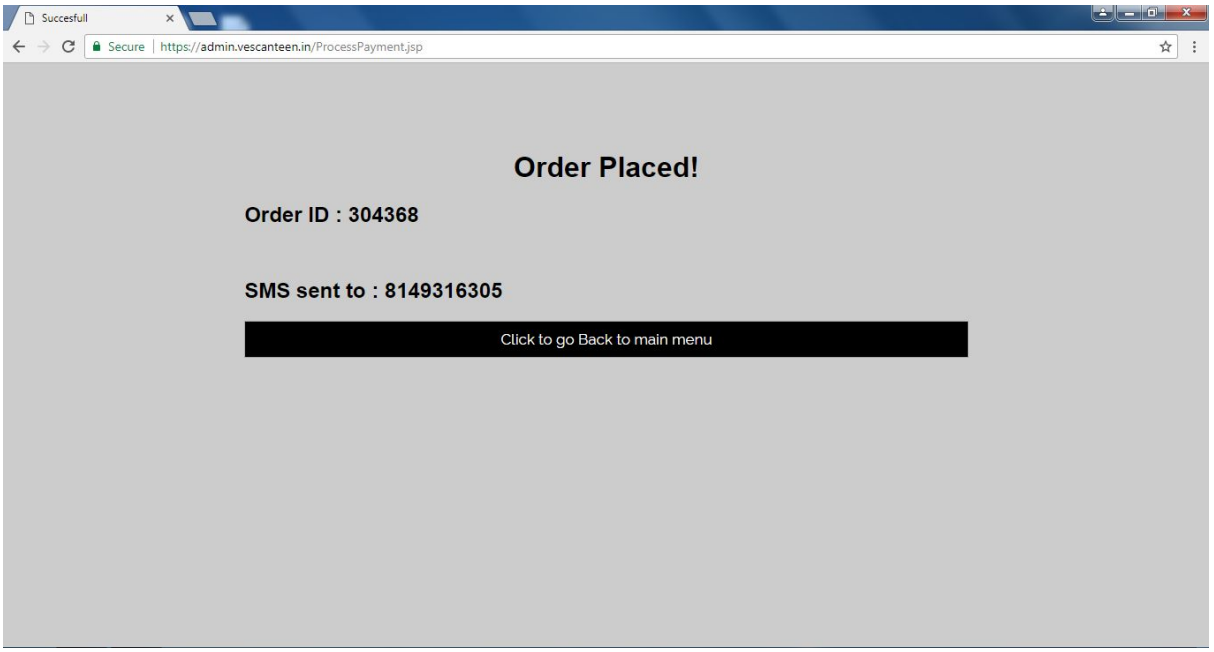


Figure 5.2.3.17 Counter Ordering Order Placed

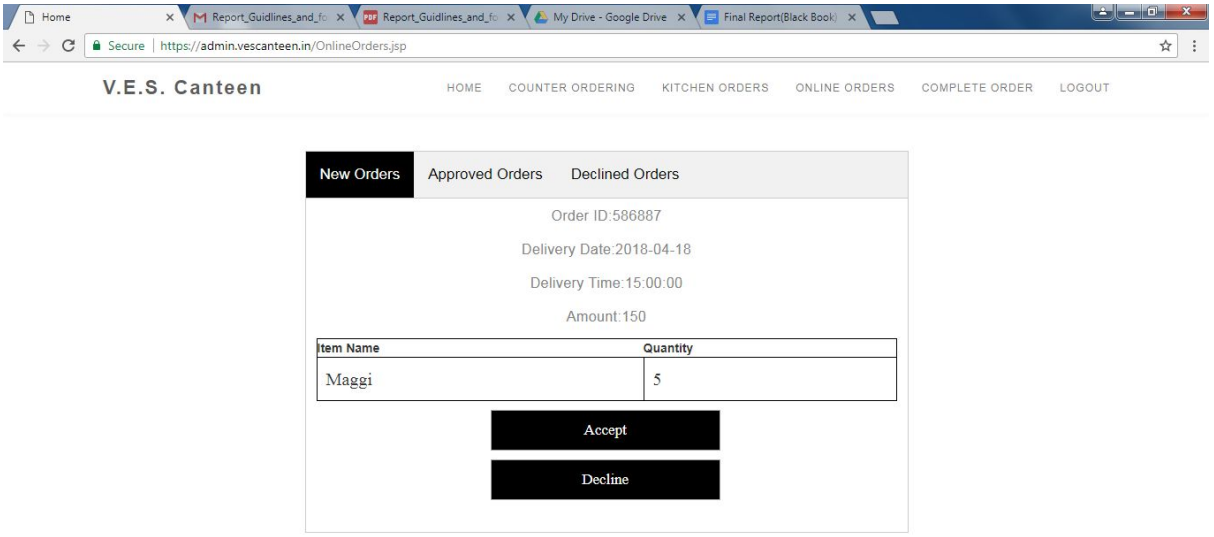


Figure 5.2.3.18 Online Orders(New)

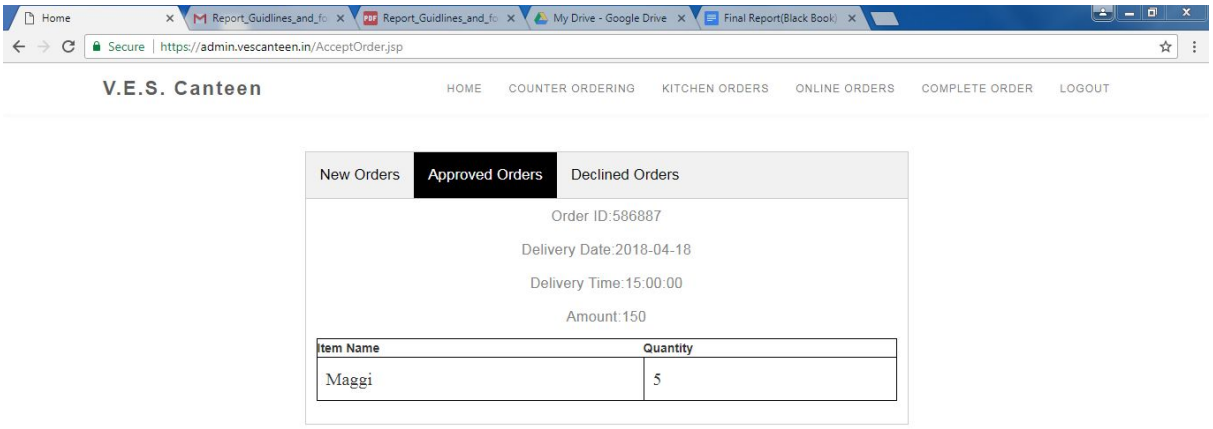


Figure 5.2.3.19 Online Orders(Approved)

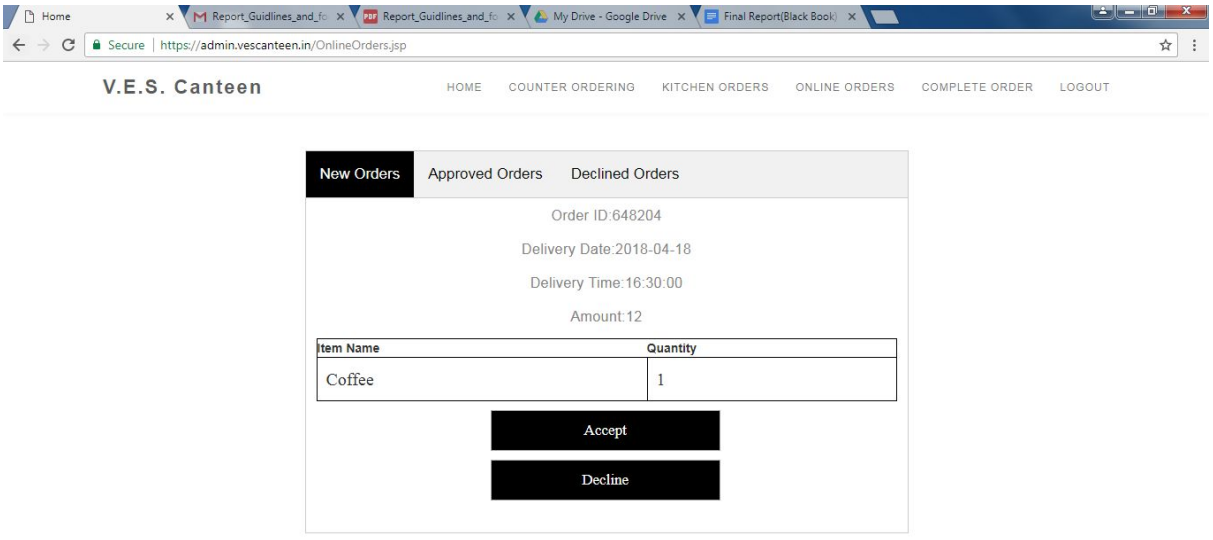
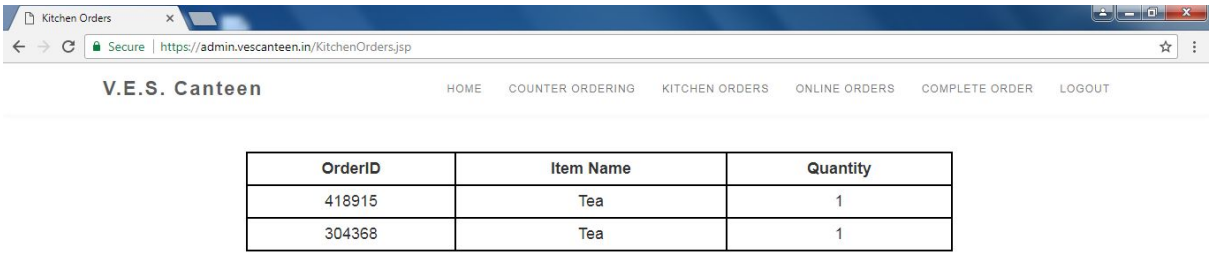


Figure 5.2.3.19 Online Orders(Declined)



OrderID	Item Name	Quantity
418915	Tea	1
304368	Tea	1

Figure 5.2.3.18 Kitchen Orders

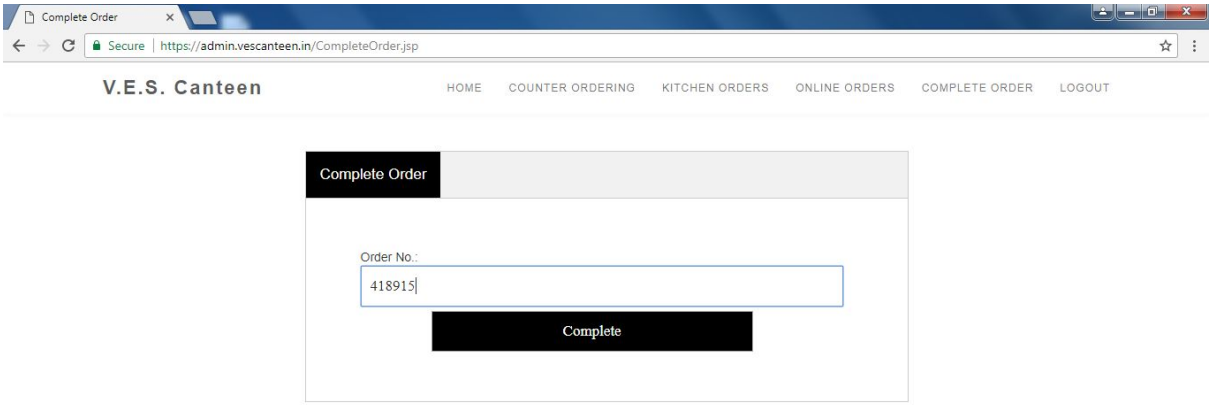


Figure 5.2.3.19 Complete Order

Chapter 6

Results

6.1 Results

- A Canteen management system which can be deployed in our own college.
- An online ordering web application to place orders in advance.
- A prepaid e-wallet which can be used by everyone to conduct quick payments while placing orders.

Chapter 7

Conclusion & Future Scope

7.1 Conclusion

ElGamal Public key Encryption scheme is a good choice for providing security to an e-wallet.

The application of primitive cryptographic functions for the proposed realization of e-wallet satisfying requirements of a secure E-wallet. Also the installation of SSL certificate to the web application adds an extra layer of security.

The web application created as a result of this project is an easy to use software which would reduce the amount of time wasted in the canteen considerably due to the features like online ordering, E-wallet payment and Displaying Orders directly in the kitchen.

7.2 Future Scope

- Improvements to the UI of the application to be made according to feedback of the customers and administrators of the canteen.
- An Android application and an IOS application to be made for a better UI.
- Integrate the E-wallet created with bank APIs in order to provide not just recharge with cash but also with bank accounts which would enhance the user experience and more users would be attracted to using the application.

Chapter 8

References

[1]Key generation algorithm design combination of RSA and ElGamal algorithm,

Ni Made Satvika Iswari Faculty of Engineering and Informatics Universitas Multimedia Nusantara Tangerang, Indonesia satvika@umn.ac.id 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia

978-1-5090-4139-8/16/$31.00 ©2016 IEEE

[2]Computational Resources for mobile E-wallet System with observers

Eligijus Sa;alausas', Jonas Muleravicius', Inga Timofejeva Kaunas University of Technology, Department of Applied Mathematics, Studentu St. 50, LT-51368, Kaunas

978-1-5386-0394-9/17/$31.00 ©2017 IEEE

[3]Canteen Food Ordering Android System

Abhishek Singh, Amit Tanwar, Aditya Sawant, Chaitanya Parulekar, Kunal Yadav, IT Department, MUMBAI University, International Journal on Recent and Innovation Trends in Computing and Communication ,

ISSN: 2321-8169

[4]Cloud Based Canteen Management System

Tazeen Khan, Daniel Yunus, Final year B.E (Computer), Rizvi College of Engineering Mumbai, Maharashtra, India. International Journal for Research in Engineering Application & Management (IJREAM)

ISSN : 2494-9150 Vol-02, Issue 08, Nov 2016

[5]**The Concept of Secure Mobile Wallet**

Hao Zhao, Sead Muftic, School of Information and Communication Technologies (ICT) Royal Institute of Technology (KTH), Stockholm, Sweden 978-0-9564263-7/6/$25.00©2011 IEEE

Acknowledgement

We would like to express our thanks of gratitude to our project guide **Mrs. Smita Jangale**(Vivekanand Education Society's Institute of Technology) who gave us the opportunity to do this project Canteen management system using e-wallet. This would not have been possible without her expertise and deep knowledge of the subject.

We would also like to thank our professor **Mrs. Pooja Shetty**(Vivekanand Education Society's Institute of Technology) for suggesting this topic to us in our third year of engineering.