

VIRTUAL WALLET USING NEAR FIELD COMMUNICATION

PROJECT REPORT

Submitted by

SRIRAM KOUSHIK U (10I349)
ASHOK KUMARESHEN V (10I303)

BACHELOR OF TECHNOLOGY

BRANCH: INFORMATION TECHNOLOGY



October 2012

DEPARTMENT OF INFORMATION TECHNOLOGY

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution Affiliated to Anna University of Technology)

COIMBATORE-641 004

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr.R.Rudramoorthy**, Principal, PSG College of Technology, for giving us an opportunity to do our project with various facilities and infrastructure, without which this project would not have been possible.

We extend our heartfelt gratitude to **Dr.V.Mahesh**, Professor and Head of the Department of Information Technology, PSG College of Technology, for his unfailing support throughout this project.

We express our sincere thanks to our faculty guide **Dr.R.Vidhyapriya** and Tutor **Mr.S.Sivakumar**, Assistant Professor (Sr.Gr), Department of Information Technology, PSG College of Technology, who was always there to help us and played a major role in the completion of the project.

We would like to thank our faculty **Mr.S.Sivakumar**, Assistant Professor, Department of Information Technology, PSG College of Technology, who helped us with various information regarding to Near Field Communication techniques, which formed the base of the project.

SYNOPSIS

VIRTUAL WALLET is the realization of the concept of easy money transaction using mobile phones based on NFC (Near Field Communication) Technology. NFC enables data transmission within a range of 4cm radius. Through VIRTUAL WALLET a user is entitled to make cash transactions either by tapping the phone over a suitable tag device or bringing it to close proximity with the tag.

Virtual wallet eliminates the conventional and unsafe method of payment through credit cards. The system shows the consolidated view of his/her transactions made so far, its details, sales suggestions and much more. The Virtual wallet system either uses current credit card information or customized prepaid cards for transaction.

LIST OF FIGURES

S.No	Figure No.	Figure Title	Page No.
1.	2.1	Credit card authorization process	3
2.	2.2	Virtual wallet sequence diagram	4
3.	5.1	Flow chart diagram	9
4	5.2	Component diagram	12
5	5.3	USE CASE diagram	12
6	6.1	Home page of the app in the simulator	13
7	6.2	Login page of the app in the simulator	14
8	6.3	Registration page in simulator	15
9	6.4	Home page of the app in C7 phone	15
10	6.5	Login page of the app in C7	16
11	6.6	Database structure	18

CONTENTS

CHAPTER	Page No.
Acknowledgement	(i)
Synopsis.....	(ii)
List of figures.....	(iii)
1. INTRODUCTION.....	1
1.1 NFC Technology	1
1.2 Qt Framework	1
1.3 Virtual wallet	2
1.3.1 Mobile Application	3
1.3.2 Vendor system	3
2 LITERATURE SURVEY.....	6
2.1 System analysis	6
2.1.1 Existing system	
2.1.2 Proposed system	
2.2 Algorithms involved	7
3 SYSTEM STUDY	9
3.1 Feasibility study	9
3.1.1 Economical Feasibility	9
3.1.2 Technical Feasibility	9
3.1.3 Social Feasibility	9
3.2 Modules	9
4 SYSTEM REQUIREMENTS.....	9
4.1 Hardware requirements	9
4.2 Software requirements	9
4.3 Phone specifications	9
5 SYSTEM DESIGN.....	11
5.1 Virtual wallet application	12
5.2 Vendor side application	13
5.3 Web server architecture	14
5.4 NFC Integration module	1
5.5 Security Module	16
5.6 Flowchart	17

5.7 UML Diagrams	17
5.7.1 Class diagram	18
5.7.2 Component diagram	19
5.7.3 Use case diagram	20
6 IMPLEMENTATION AND TESTING.....	14
6.1 Implementation	21
6.2 Testing	22
7 CONCLUSION.....	14
8 BIBLIOGRAPHY.....	15
9 APPENDIX I.....	16
10 APPENDIX II.....	21

Chapter 1

INTRODUCTION

INTRODUCTION

VIRTUAL WALLET is the realization of the concept of easy money transaction using mobile phones based on NFC (Near Field Communication) Technology. NFC enables data transmission within a range of 4cm radius. Through VIRTUAL WALLET a user is entitled to make cash transactions either by tapping the phone over a suitable tag device or bringing it to close proximity with the tag. Through the use of VIRTUAL WALLET we eliminate the conventional use of credit cards.

Mobile phones especially smart phones have invaded people's lives in a way it has never been. From now on there won't be any worries in case people just lost their credit card or when there aren't any ATMs around. Virtual wallet is a complete system which offers customer easy payment through their mobile phones just by **tapping** on the vendor's customized payment device. Virtual wallet eliminates the conventional and unsafe method of payment through credit cards. The system shows the consolidated view of his/her transactions more made so far, its details, sales suggestions and much more. It uses **NFC** as its essence technology. It is as easy for the customer to just install an application, choose a method for payment and tap around to start making payments.

The report is divided mainly in 5 chapters covering literature on Virtual wallet based on NFC (chapter 2), our system study (chapter3), System requirements (chapter4) , System and Design (chapter5), Implementation and Testing (chapter6).

Appendix I contains code snippets and Appendix II contains the output screenshots

1.1 NFC TECHNOLOGY

Near field communication (NFC) is a set of standards for smart phones and similar devices to establish radio communication with each other by touching them together or bringing them into close proximity, usually no more than a few centimeters. Present and anticipated applications include contactless transactions, data exchange, and simplified setup of more complex

communications such as Wi-Fi. Communication is also possible between an NFC device and an unpowered NFC chip, called a "tag".

1.2 Qt FRAMEWORK

Qt is a full development framework with tools designed to streamline the creation of applications and user interfaces for desktop, embedded and mobile platforms.

Qt framework - intuitive APIs for C++ and CSS/JavaScript-like programming with **Qt Quick** for rapid UI creation

Qt Creator IDE - powerful cross-platform integrated development environment, including UI designer tools and on-device debugging

Tools and tool chains - All you need: simulator, local and remote compilers, internationalization support, device toolchains and more.

With Qt, you can **reuse code** efficiently to target multiple platforms with one code base. The modular C++ class library and developer tools easily enables developers to create applications for one platform and easily build and run to deploy on another platform. We use Qt framework to develop the application targeted at Symbian based Nokia phones.

1.3 VIRTUAL WALLET

1.3.1 Mobile application

A mobile application for the customer which enables him/her for making payments through his mobile is developed using Qt framework. The mobile application provides a complete package of functionalities for the user regarding the transactions that he/she has made so far, and payment details in a consolidated manner.

1.3.2 Vendor System

The Vendor system is the software module that is implemented at the shop keeper's or the vendor's site where Virtual Wallet service is provided to the customers .The system is implemented through VB which is in turn connected to the web server. The details of every shop providing the service is maintained in the web server which are used to track the transactions related to his/her shop.

Chapter 2

LITERATURE SURVEY

2.1 SYSTEM ANALYSIS

2.1.1 EXISTING SYSTEM

The current cash transaction system involves a credit card, a credit card swiper which reads data from the credit card authenticates and proceeds with the cash transaction. Alternate way is to pay through cash.

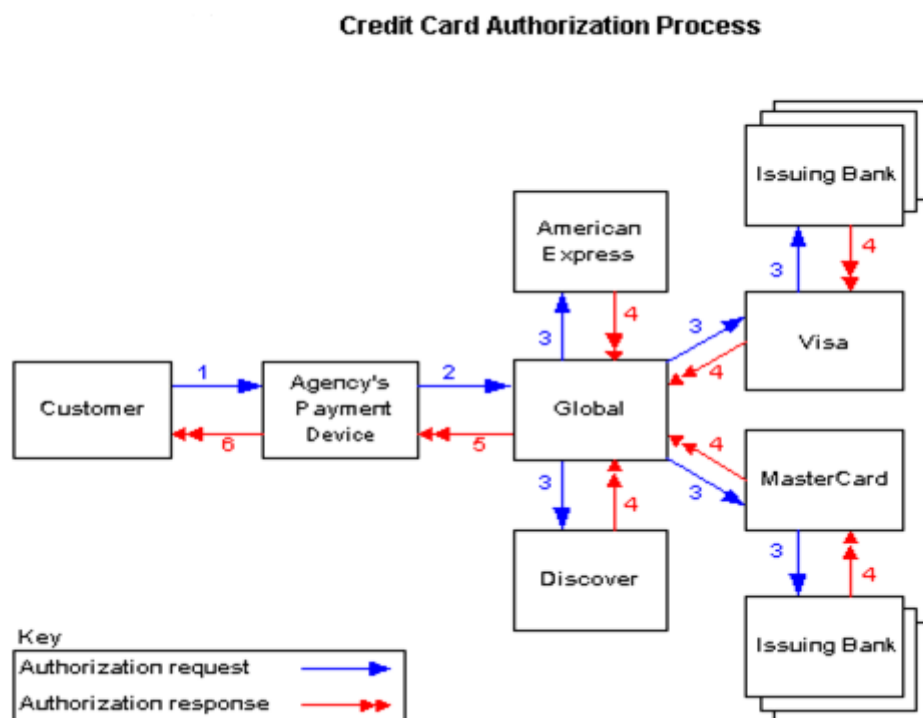


Figure 2.1 Credit card authorization process

The disadvantages of current cash systems are as follows:

1. A credit card can be misused by someone else other than the destined owner.
2. Large valued transactions cannot be carried out by mere cash.
3. Credit card transactions are more prone to hacks.

2.1.2 PROPOSED SYSTEM

VIRTUAL WALLET system enables a user to transfer money through mobile phones using NFC technology.

The basic sequence diagram of the system:

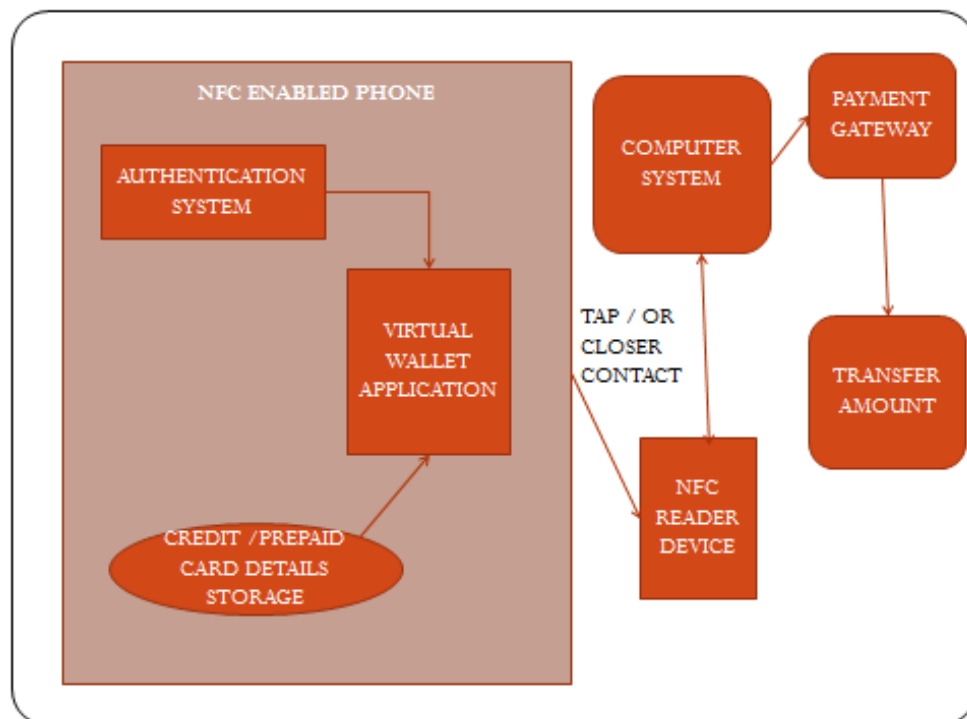


Figure 2.2 Virtual wallet sequence diagram

The advantages of proposed model are as follows:

- The system provides a higher level of security.

- The system involves NFC whose radius of data transmission is 4cm which makes the system lesser prone to data tapping.
- The system provides a consolidated view of all the transactions made so far.
- In case of theft of mobile phone, the system requires a secret PIN for transaction
- Allows transaction through multiple cash cards through one compact mobile application

2.2 ALGORITHMS INVOLVED

We use the **HMAC(Hash Based Message Authentication Code)** algorithm for generating the prepaid card number. The algorithm is used for generating 14 digit random numbers. This algorithm is implemented using

PHP. Every random number generated has a hash value which is equivalent to a fixed price starting from 1000 and continued with denominations in 1000 up to 50000. Every time a user requests a prepaid number, it inputs a price value and prints equivalent hash value number. When the number is inputted in the mobile system the equivalent amount is updated in the user database.

An insight into the HMAC algorithm:

In cryptography, a **hash-based message authentication code (HMAC)** is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the *data integrity* and the *authenticity* of a message. Any cryptographic hash function, such as MD5

or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits, and on the size and quality of the key.

An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function. For example, MD5 and SHA-1 operate on 512-bit blocks. The size of the output of HMAC is the same as that of the underlying hash function (128 or 160 bits in the case of MD5 or SHA-1, respectively), although it can be truncated if desired.

CHAPTER 3

SYSTEM STUDY

3.1 FEASIBILITY STUDY

3.1.1 ECONOMICAL FEASIBILITY

Virtual wallet is a simple system whose cost includes the building cost of the application while hosting it in the market, the NFC reader device. The above costs denote the costs beared by the developer. The customer on the other hand could download the app for free from the OVI store. The customer has to pay a certain amount of service charge for every transaction that he/she makes as denoted in the below table for the vendor the cost include the buying of NFC processing device.

TRANSACTION VALUE	% SERVICE CHARGE
<=5000	2%
<=10000	2.5%
<=50000	3.5%

3.1.1 TECHNICAL FEASIBILITY

The project can be achieved VIRTUAL WALLET is the realization of the concept of easy money transaction using mobile phones based on NFC (Near Field Communication) Technology. NFC enables data transmission within a range of 4cm radius. Through VIRTUAL WALLET an user is entitled to make cash transactions either by tapping the phone over a suitable tag device or bringing it to close proximity with the tag. VIRTUAL WALLET is an application devised for the NFC enabled Nokia Symbian Belle OS phones. The application is developed having the Nokia C7 mobile phone model as the basic framework model. The application is developed with the help of Qt framework.

3.1.3 SOCIAL FEASIBILITY

Virtual wallet mainly aims at working class people and students who own a mobile phone supporting NFC. This system mainly aims at the Indian audience whose working class is now at their peak of rush and are in need for speed. Any vendor who has a shop for himself and ready to digitize their transaction system may use the system. Roughly around 2700000 [Census India] people own a smart phone in India. Keeping aside the fact that an initial set of audience would resist change, 1000000 people would start using it in the first phase. Another major audience is the large institutions like IT

companies, colleges which see to incorporate cashless transaction. For such an audience a customized system would work the way.

3.2 MODULES

- Module 1:** Mobile application system authentication
- Module 2:** User interface for the application.
- Module 3:** Interfacing with the tag reader.
- Module 4:** A computer application for simple billing purposes.
- Module 5:** Creating a web interface for a server for storage of user details

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- A computer system for the merchant to use the billing software
- An NFC reader
 - 13.56 MHz RFID Support
 - ISO7816-3 SAM
- An NFC enabled phone(Nokia C7)(for testing purposes)

4.2 SOFTWARE REQUIREMENTS

- Qt SDK 1.2.1
- Visual Basic to create the billing software
- Qt Quick component
- Nokia phone supporting Symbian **BELLE** Operating system
- CODA – Code online Debugging agent

4.2 PHONE SPECIFICATIONS

- Nokia based phone
- Operating System - Symbian
- Os Version – Nokia Belle or Nokia Anna
- NFC Card supported
- CODA – Code online Debugging agent on phone

Chapter 5

SYSTEM DESIGN

5.1 Virtual wallet application

A fast way to pay

When you shop online, just sign into your account and use your stored payment info to make a purchase. No need to fill out long credit card forms or shipping details.

And when you pay in-store, all you have to do is quickly tap the back of your phone on the NFC terminal, and you're good to go

Safer wallet, safer shopping

Thanks to the Google Wallet PIN, encrypted storage of card numbers of the payment cards stored online, and the ability to remotely disable a lost wallet, you're safe and secure when shopping with Google Wallet.

Saving made simple

With Virtual Wallet you can redeem Offers automatically at participating stores. We Offers are deals on products and services at local and online merchants. When you buy or save an offer, it automatically syncs to your Virtual Wallet so it's always with you wherever you go.

All your shopping all in one place

Our online management center lets you view all your Virtual Wallet activity all in one place. Add or delete payment cards and shipping addresses. See both your online and in-store transaction history. It's all of your shopping, visualized.

5.2 VENDOR SIDE APPLICATION**5.3 WEB SERVER ARCHITECTURE****5.4 NFC INTEGRATION MODULE****5.5 SECURITY MODULE****A wallet you can lock**

Unlike leather wallets, Google Wallet is password protected. On your phone, a 4-digit Google Wallet PIN prevents unauthorized access. For added protection use your Android's screen lock. Online, Google has multiple layers of password protection to help you create a defense against cyber criminals.

Encryption is the key

The credit and debit cards you store in Google Wallet are safely encrypted on secure servers in a secure location. When you pay in-store, Google actually pays the merchant, and then processes the transaction with your selected credit or debit card. So neither the merchant nor the Android operating system ever gets your real payment card information.*

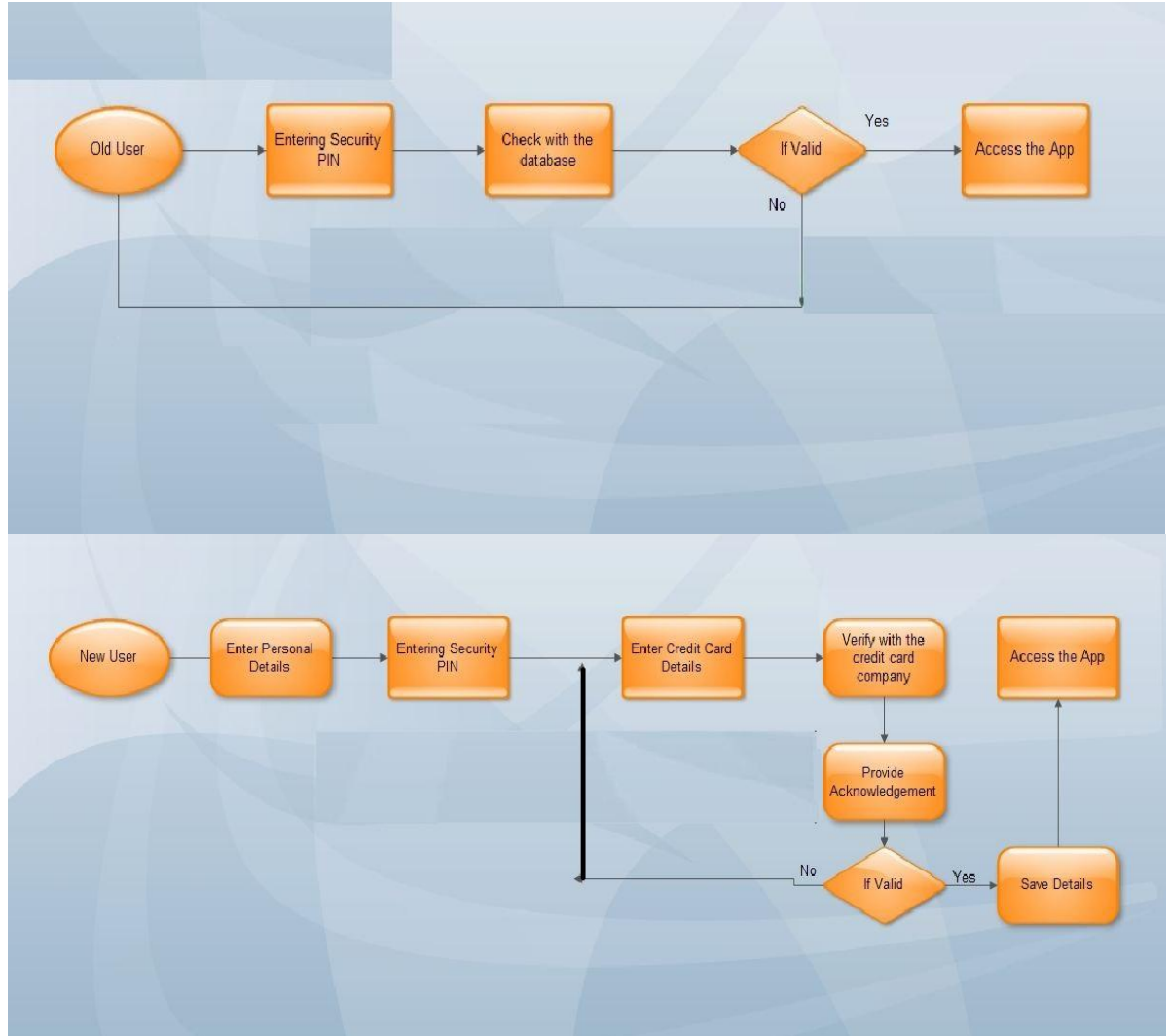
Nothing to see here

Google Wallet always keeps the account numbers on your cards hidden. When your cards appear on your phone's screen in-store, the numbers are not visible. And when you pay online, we only display the last four digits.

Not sharing is caring

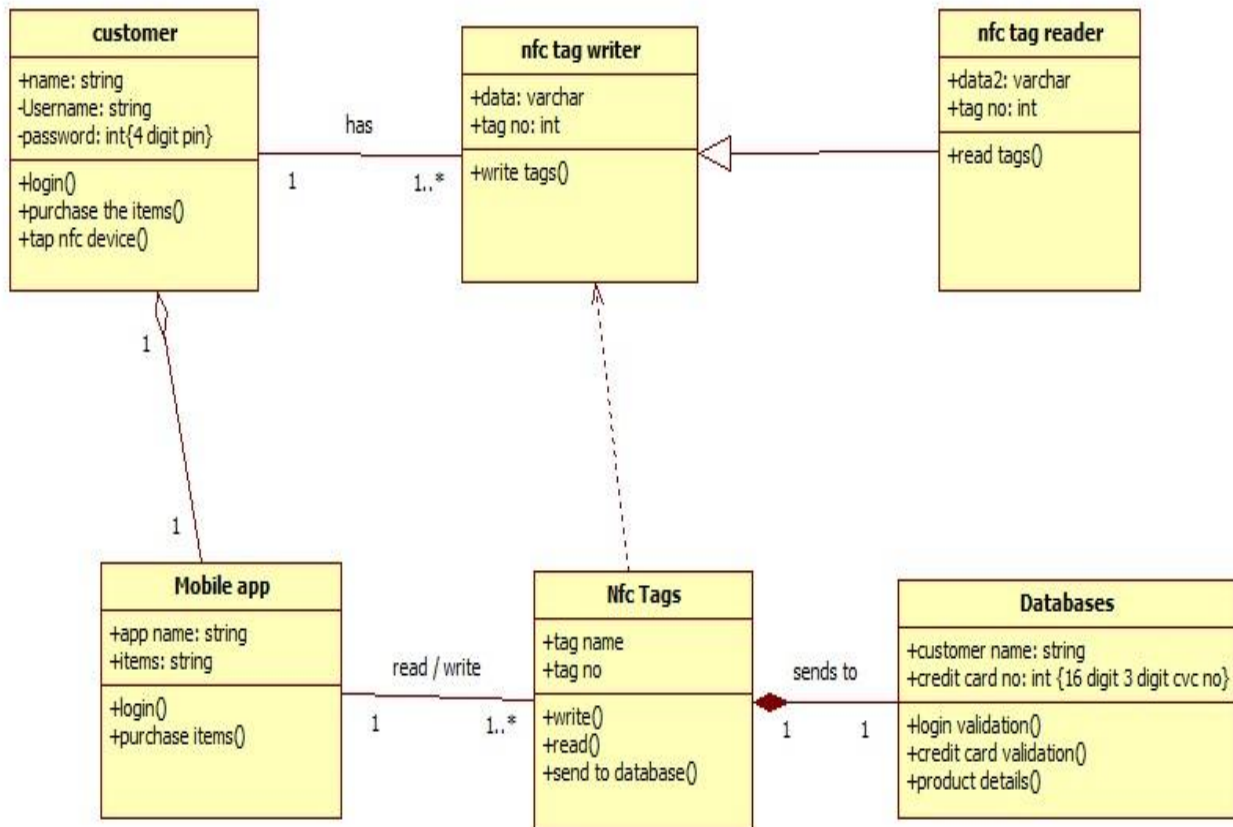
Google Wallet does not share the credit and debit card info you store in your wallet with any merchant online or offline. No one should have access to that information, except you.

5.6 FLOW CHART

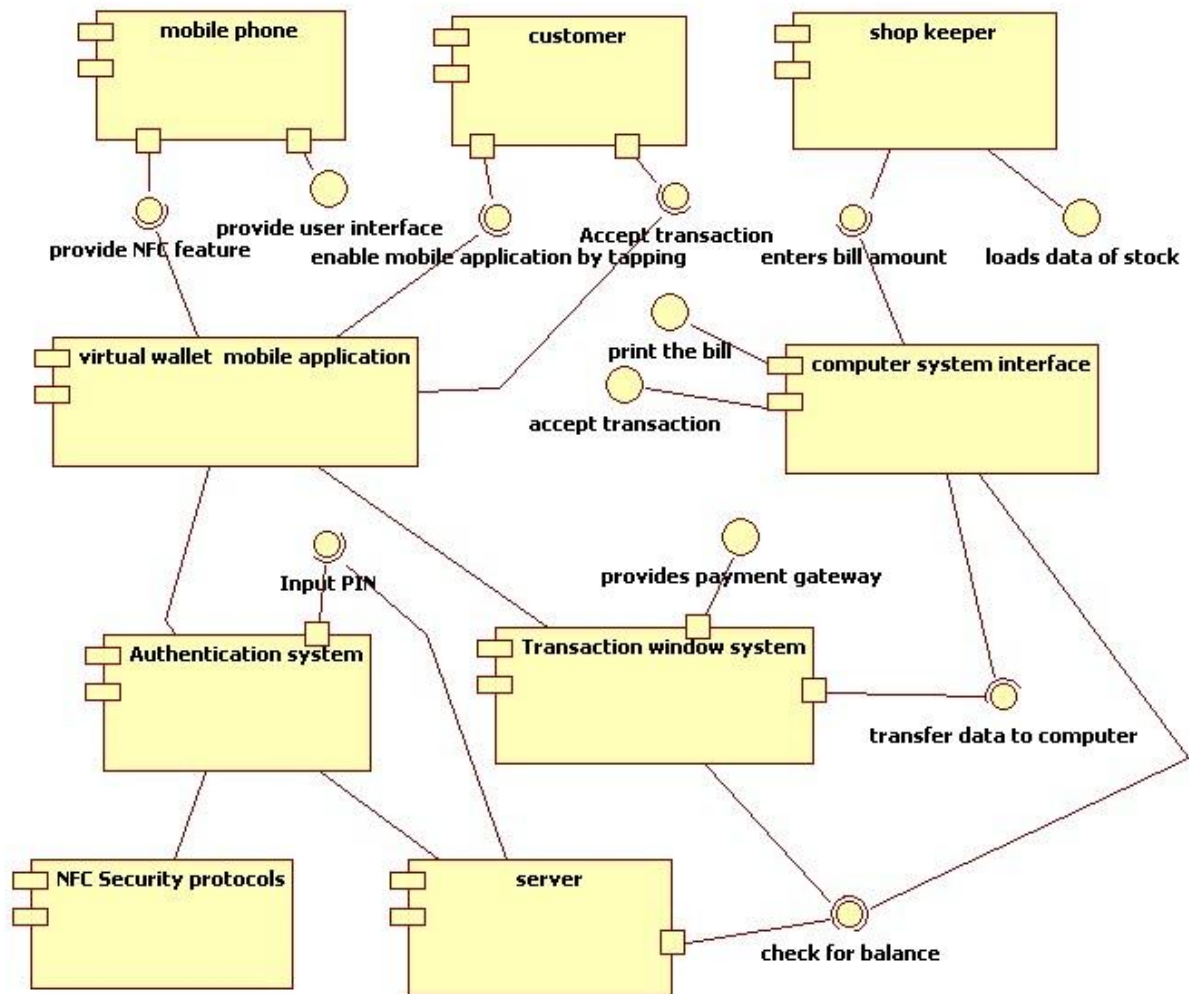


5.6 UML DIAGRAMS

5.6.1 CLASS DIAGRAM



5.6.2 COMPONENT DIAGRAM



CHAPTER 6

IMPLEMENTATION AND TESTING

6.1 IMPLEMENTATION

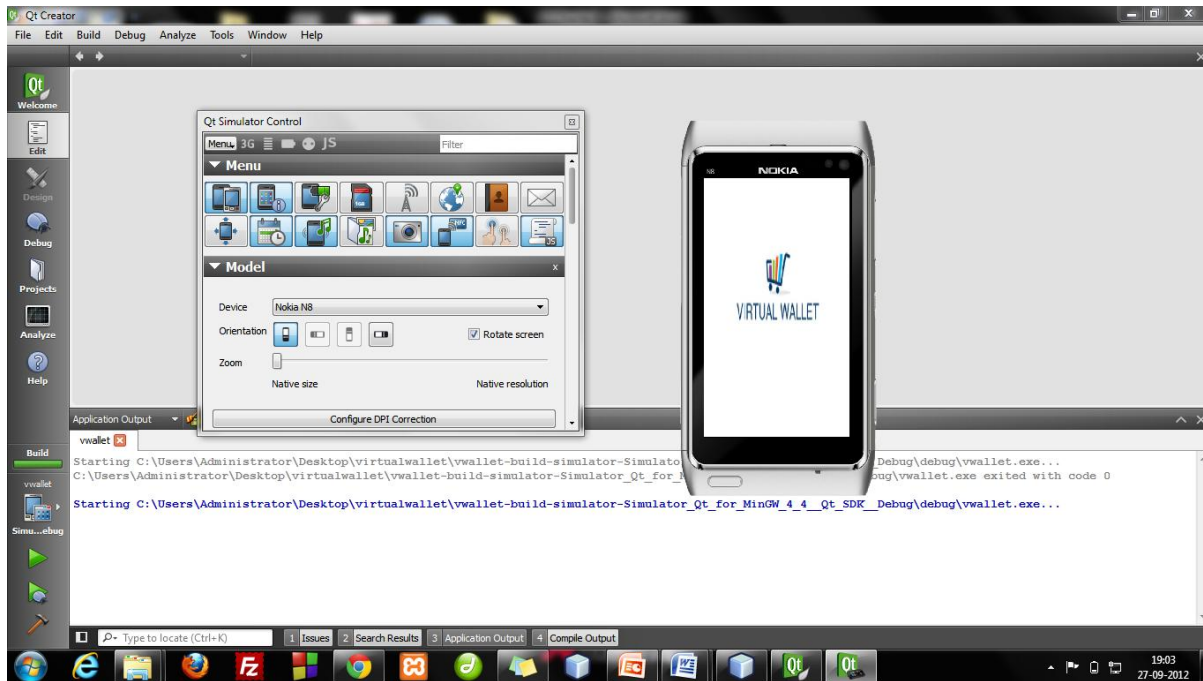


Figure 6.1 Home page of the app in the simulator

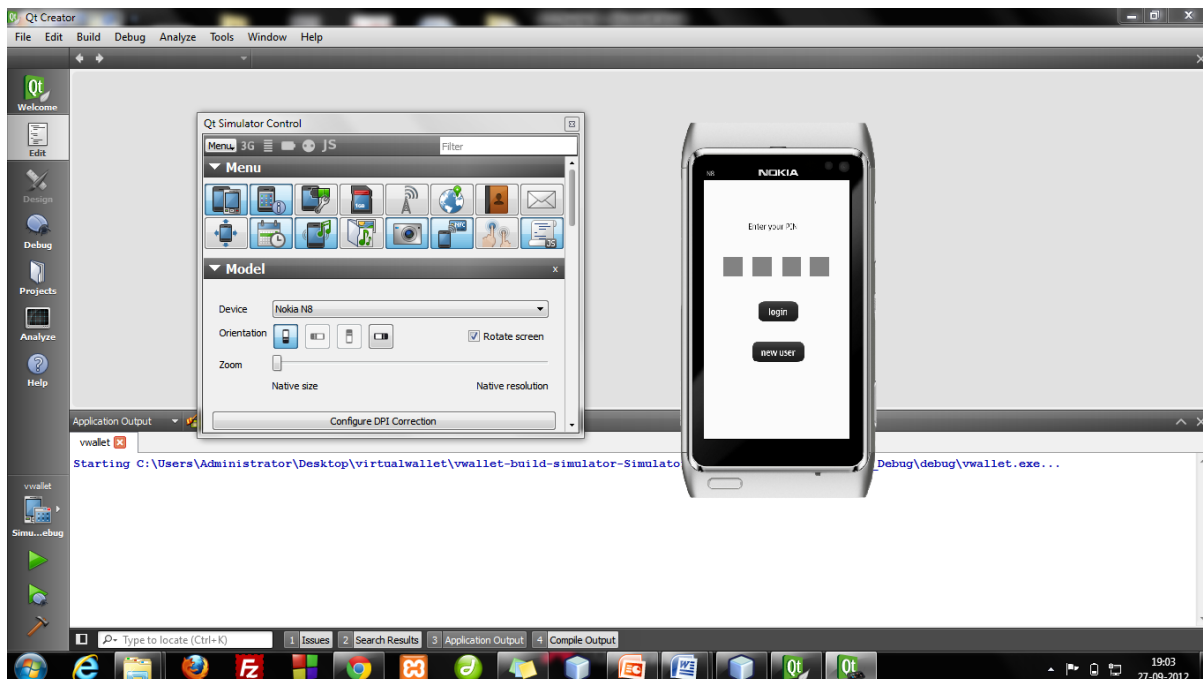


Figure 6.2 Login page of the app in the simulator

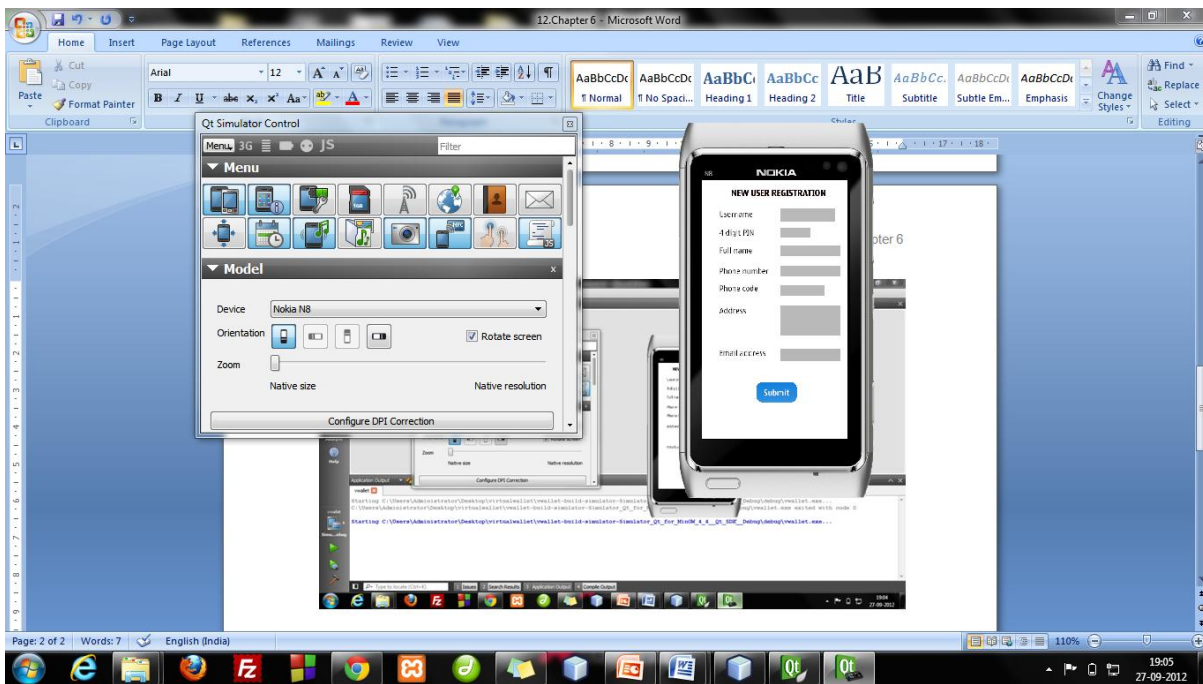


Figure 6.3 Registration page of the app in the simulator

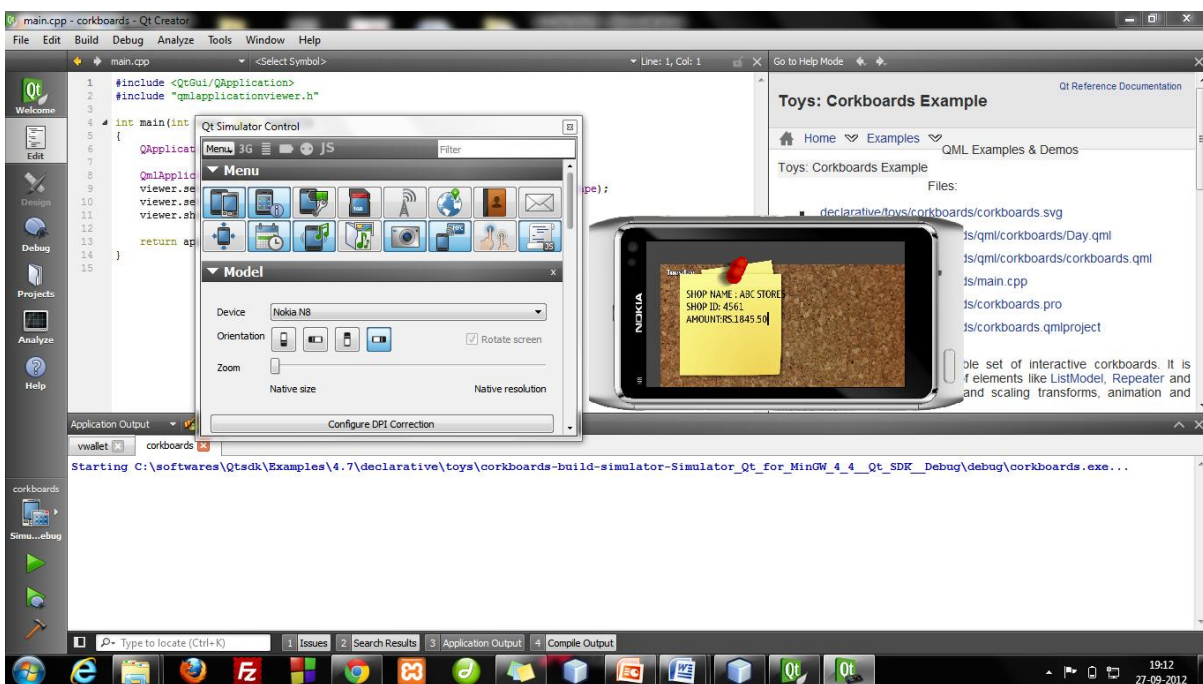


Figure 6.4 Transaction page of the app in the simulator



Figure 6.5 Transaction page of the app in a C7 mobile



Figure 6.6 Login page of the app in a C7 mobile

The screenshot shows the phpMyAdmin interface with the following table structures:

members

Column	Type	Null	Default	Comments
fname	varchar(255)	No		
lname	varchar(255)	No		
gender	varchar(1)	No		
city	varchar(30)	No		
state	varchar(30)	No		
country	varchar(20)	No		
email	varchar(255)	No		
college	varchar(255)	No		
mobile	varchar(20)	No		
acode	varchar(1)	No		

prepaid

Column	Type	Null	Default	Comments
cardno	bigint(15)	No		unique card number
cash	decimal(7,2)	No		equivalent cash value for the card
owner	varchar(30)	No		the user who owns the card after buying it...initially null
availability	varchar(4)	No		"avl" or "sold" for available or sold

transaction

Column	Type	Null	Default	Comments
transid	int(7)	No		transaction id
amount	decimal(7,2)	No		transaction value
username	varchar(30)	No		the user who made the transaction
shopid	varchar(10)	No		the shop where the transaction took place

user

Column	Type	Null	Default	Comments
username	varchar(30)	No		max of 30 since the email can be username
name	varchar(20)	No		name of customer
email	varchar(30)	No		email address of the user
address	varchar(100)	No		residential address
phno	bigint(10)	No		phone number
phcode	mediumint(8)	No		phone code

Figure 6.7 Database structure on web server

6.2 TESTING

The system was tested for various input amounts and shop codes. The testing was done entirely in the Qt simulator which makes the testing quite easier. The testing included simulating various amount values for various shop transactions. A complete exhaustive set of test cases were generated tested and the system was updated to accommodate those changes. The basic testing strategy used was by dividing the system into various **modules** and **individually testing** each of them. The Authentication module was tested for SQL injection and various other trespassing mechanisms. The User Interface module was checked for easy user interaction, unresponsive links. The NFC integration module was tested with the tag reader with various test case input values. The security module was tested for the working of the various NFC protection mechanisms in the NFC protocol used.

BIBLIOGRAPHY

BOOKS

1. Brian Jepson and Tom Igoe, "Getting Started with NFC", © 2012, O'Reilly Media, Inc.
2. "The Smartphone Wallet: Understanding the Disruption Ahead" by David W. Schropfer.
3. "NFC: A beginners Guide to Near Field Communication" by Chris Branden.

INTERNET

- www.google.com/wallet/
- www.ieeeexplore.com
- www.developer.nokia.com/Develop/NFC/
- www.nearfield.org/
- qt-project.org/

CONCLUSION

Thus, Virtual Wallet enhances the existing system of making transactions by making it easier and safer. The system allows the user to easily make transactions through their mobile phones over their credit cards. The system provides the user to make transactions through various options like shifting between various credit cards or prepaid cards. This transaction system is much faster than the current system and is more advantageous since it has lower chances of theft and more compact. Through Virtual wallet the customer could have the consolidated view of the transactions he/she has made so far. Future enhancements are to enable the customer to view the stores nearby that provides Virtual wallet service and the various offers provided by the stores.

APPENDIX I

IMPLEMENTATION OF VIRTUAL WALLET SYSTEM

Main.qml:

```
// import QtQuick 1.0 // to target S60 5th Edition or Maemo 5
import QtQuick 1.1
Rectangle {
    id: firstit
    width: 360
    height: 640
    Loader {
        id: mainloader
        anchors.fill: parent
    }
    MouseArea {
        id: mouseareal
        x: 0
        y: 122
        width: 360
        height: 654
        anchors.rightMargin: 0
        anchors.bottomMargin: -136
        anchors.leftMargin: 0
        anchors.topMargin: 122
        anchors.fill: parent
        /*onClicked: {

firstit.opacity=7;image1.visible=false;mouseareal.enabled=false;
        mainloader.source="login.qml";
        }

*/
        Timer {
            id: timer1;
            interval: 3000; running: true;
            onTriggered: {

firstit.opacity=7;image1.visible=false;mouseareal.enabled=false;
                mainloader.source="login.qml";
            }
        }
    }
    Image {
        id: image1
        x: 72
        y: 52
        width: 217
        height: 234
        source: "Capture.JPG"
    }
}
```



```

    }
}
// import QtQuick 1.0 // to target S60 5th Edition or Maemo 5
import QtQuick 1.1
import "auth.js" as Myfirst
import Qt.labs.components.native 1.0
//QDeclarativeEngine *engine = new QDeclarativeEngine(parent);
//QDeclarativeComponent component(engine,
//QUrl::fromLocalFile("main.qml"));
//QObject *myObject = component.create();
Rectangle {
    id: firstrect
    width: 360
    height: 640
    color: "#fbfbfb";
/*
    XmlListModel {
        id: xmlModel
        source: "http://www.psgtechclub.com/data.xml"
        query: "/rss/channel/item"
        XmlRole { name: "title"; query: "title/string()" }
        XmlRole { name: "pubDate"; query: "pubDate/string()" }
    }

    ListView {
        width: 180; height: 300
        visible: false
        model: xmlModel
        delegate: Text { text: title + ": " + pubDate }
    }
*/
    Text {
        id: text1
        x: 110
        y: 101
        width: 141
        height: 33
        text: qStr("Enter your PIN")
        font.pixelSize: 20
    }
    TextInput {
        id: text_input1
        x: 120
        y: 190
        width: 48
        height: 48
        text: qStr("")
        horizontalAlignment: TextInput.AlignHCenter
        font.bold: false
        opacity: 0.750
        font.pixelSize: 23
        maximumLength: 1
        echoMode: TextInput.Password
    }
}

```

```

    BorderImage {
        id: border_image1
        x: 61
        y: -89
        width: 99
        height: 100
        anchors.fill: parent
        source: "back.png"
        opacity: 0.650
    }
}
TextInput {
    id: text_input2
    x: 48
    y: 190
    width: 48
    height: 48
    text: qsTr("")
    horizontalAlignment: TextInput.AlignHCenter
    font.bold: false
    opacity: 0.750
    font.pixelSize: 23
    maximumLength: 1
    echoMode: TextInput.Password
    BorderImage {
        id: border_image2
        x: 61
        y: -89
        width: 99
        height: 100
        anchors.fill: parent
        source: "back.png"
        opacity: 0.650
    }
}
TextInput {
    id: text_input3
    x: 193
    y: 190
    focus: true
    width: 48
    height: 48
    text: qsTr("")
    horizontalAlignment: TextInput.AlignHCenter
    opacity: 0.750
    font.pixelSize: 23
    echoMode: TextInput.Password
    maximumLength: 1
    BorderImage {
        id: border_image3
        x: 0
        y: 0
    }
}

```



```

        width: 99
        height: 100
        anchors.rightMargin: 0
        anchors.bottomMargin: 0
        anchors.leftMargin: 0
        anchors.topMargin: 0
        anchors.fill: parent
        source: "back.png"
        opacity: 0.650
    }
}
TextInput {
    id: text_input4
    x: 262
    y: 190
    focus: true
    width: 48
    height: 48
    text: qStr("")
    horizontalAlignment: TextInput.AlignHCenter
    opacity: 0.750
    font.pixelSize: 23
    maximumLength: 1
    echoMode: TextInput.Password
    BorderImage {
        id: border_image4
        x: 61
        y: -89
        width: 99
        height: 100
        anchors.fill: parent
        source: "back.png"
        opacity: 0.650
    }
}
Image {
    id: image2
    x: 129
    y: 300
    width: 104
    height: 40
    // source: "signin.gif"
    Column {
        Button { x: 5; y: 19; width: 100; height: 50; text:
"login"; checkable: false; opacity: 1; }
    }
    Column{
        x: -10
        y: 100
        width: 130
        Button { x: 0; y: 100; width: 130; height: 50; text: "new
user"; checkable: false; opacity: 1;

```

```

        onClicked: {

firstrect.visible=false;image1.visible=false;mousearea1.enabled=false;
text1.visible=false;text_input1.visible=false;text_input2.visible=false;
text_input3.visible=false;text_input4.visible=false;
            mainloader.source="newuser.qml"
        }
    }
}

```

Newuer.qml

```

// import QtQuick 1.0 // to target S60 5th Edition or Maemo 5
import QtQuick 1.1
import Qt.labs.components.native 1.0
import "auth.js" as Myfirst
Rectangle {
    id: newusemain
    width: 360
    height: 640
    color: "#ffffff"
    Text {
        id: text1
        x: 43
        y: 71
        width: 90
        height: 32
        text: qsTr("Username")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit1
        x: 195
        y: 71
        width: 134
        height: 32
        text: qsTr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image3
            anchors.fill: parent
            opacity: 0.280
            source: "back.png"
        }
    }
    Text {
        id: text2
        x: 43
        y: 117
    }
}

```

```

        width: 82
        height: 32
        text: qStr("4 digit PIN")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit2
        x: 195
        y: 117
        width: 74
        height: 26
        text: qStr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image1
            width: 100
            height: 100
            opacity: 0.280
            source: "back.png"
            anchors.fill: parent
        }
    }
    Text {
        id: text3
        x: 43
        y: 212
        width: 133
        height: 32
        text: qStr("Phone number")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit3
        x: 195
        y: 212
        width: 147
        height: 26
        text: qStr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image7
            x: 0
            y: 0
            width: 147
            height: 26
            anchors.topMargin: 0
            anchors.fill: parent
            opacity: 0.280
            source: "back.png"
        }
    }
    Text {

```

```
        id: text4
        x: 43
        y: 161
        width: 90
        height: 32
        text: qsTr("Full name")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit4
        x: 195
        y: 161
        width: 147
        height: 26
        text: qsTr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image5
            anchors.fill: parent
            opacity: 0.280
            source: "back.png"
        }
    }
    Text {
        id: text5
        x: 43
        y: 255
        width: 101
        height: 32
        text: qsTr("Phone code")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit5
        x: 195
        y: 261
        width: 109
        height: 26
        text: qsTr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image6
            anchors.fill: parent
            opacity: 0.280
            source: "back.png"
        }
    }
    Text {
        id: text6
        x: 43
        y: 310
        width: 74
```

```

        height: 32
        text: qsTr("Address")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit6
        x: 195
        y: 310
        width: 147
        height: 74
        text: qsTr("")
        font.pixelSize: 17
        BorderImage {
            id: border_image4
            anchors.fill: parent
            opacity: 0.280
            source: "back.png"
        }
    }
    Text {
        id: text7
        x: 43
        y: 415
        width: 125
        height: 32
        text: qsTr("Email address ")
        font.pixelSize: 19
    }
    TextInput {
        id: text_edit7
        x: 195
        y: 417
        width: 147
        height: 30
        text: qsTr("")
        font.pixelSize: 12
        BorderImage {
            id: border_image2
            anchors.fill: parent
            x: -13
            y: -350
            width: 147
            height: 30
            anchors.bottomMargin: 0
            opacity: 0.290
            source: "back.png"
        }
    }
}
Column {
    x: 135
    y: 500

```

```

        Button { width: 100; height: 50; text: "Submit";
onClicked: Myfirst.authent();    }
    }
    Text {
        id: text8
        x: 65
        y: 17
        width: 251
        height: 40
        text: qsTr("NEW USER REGISTRATION")
        styleColor: "#d96262"
        style: Text.Raised
        font.family: "Myriad Pro"
        horizontalAlignment: Text.AlignHCenter
        font.bold: true
        font.pixelSize: 21
    }
}

```

Virt.sql

```

-- phpMyAdmin SQL Dump
-- version 3.4.10.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Sep 25, 2012 at 03:44 PM
-- Server version: 5.0.91
-- PHP Version: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `psgteche_e360`
--

```

```

-- -----
--

--
-- Table structure for table `members`
--

CREATE TABLE IF NOT EXISTS `members` (
  `fname` varchar(255) NOT NULL default '',
  `lname` varchar(255) NOT NULL default '',
  `gender` varchar(1) NOT NULL default '',
  `city` varchar(30) NOT NULL,
  `state` varchar(30) NOT NULL,
  `country` varchar(20) NOT NULL,
  `email` varchar(255) NOT NULL default '',
  `college` varchar(255) NOT NULL default '',
  `mobile` varchar(20) NOT NULL default '',
  `acco` varchar(1) NOT NULL default ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `members`
--

-- -----
--

--
-- Table structure for table `prepaid`
--

CREATE TABLE IF NOT EXISTS `prepaid` (
  `cardno` bigint(15) NOT NULL COMMENT 'unique card
number',
  `cash` decimal(7,2) NOT NULL COMMENT 'equivalent cash
value for the card',
  `owner` varchar(30) NOT NULL COMMENT 'the user who owns
the card after buying it....initially null',
  `availability` varchar(4) NOT NULL COMMENT '"avl" or
"sold" for available or sold'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----
--

```

```
--
-- Table structure for table `transaction`
--

CREATE TABLE IF NOT EXISTS `transaction` (
  `transid` int(7) NOT NULL auto_increment COMMENT
'transaction id',
  `amount` decimal(7,2) NOT NULL COMMENT 'transaction
value',
  `username` varchar(30) NOT NULL COMMENT 'the user who
made the transaction',
  `shopid` varchar(10) NOT NULL COMMENT 'the shop where
the transaction took place',
  PRIMARY KEY (`transid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--

--
-- Table structure for table `user`
--

CREATE TABLE IF NOT EXISTS `user` (
  `username` varchar(30) NOT NULL COMMENT 'max of 30
since the email can be username',
  `name` varchar(20) NOT NULL COMMENT 'name of customer',
  `email` varchar(30) NOT NULL COMMENT 'email address of
the user',
  `address` varchar(100) NOT NULL COMMENT 'residential
address',
  `phno` bigint(10) NOT NULL COMMENT 'phone number',
  `phcode` mediumint(5) NOT NULL COMMENT 'phone code',
  `pin` int(4) NOT NULL COMMENT '4 digit pin code',
  `balance` decimal(10,2) NOT NULL COMMENT 'cash
balance',
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-----

--

--
-- Table structure for table `vendor`
```


--

```
CREATE TABLE IF NOT EXISTS `vendor` (  
  `shopid` varchar(10) NOT NULL COMMENT '10 letter id for  
the shop',  
  `shopname` int(30) NOT NULL COMMENT 'shop name',  
  `address` varchar(100) NOT NULL COMMENT 'shop address',  
  `phno` bigint(10) NOT NULL COMMENT 'phone number',  
  `cashsofar` decimal(10,2) NOT NULL COMMENT 'cash gained  
so far..will be reset if money is transferred to  
account',  
  `bankaccno` varchar(30) NOT NULL COMMENT 'bank account  
number',  
  `phcode` mediumint(5) NOT NULL COMMENT 'phone code'  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
  
/*!40101 SET  
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET  
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET  
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```