PROJECT REPORT ON

MODULARISATION OF MOBILE SHOPPING ASSISTANCE SYSTEM

BY

TEJ PRATAP SINGH	**- B80384332**
SHUBHAM JAKHETIA	**- B80384319**
SOURABH SHUBHAM	**- B80384327**

UNDER THE GUIDENCE OF

PROF. PRASAD HALGAONKAR

IN FULFILLMENT OF

BACHELOR ENGINEERING (COMPUTER)
DEGREE OF SAVITABAI PHULE PUNE UNIVERSITY

MAY/JUNE 2014-15

DEPARTMENT OF COMPUTER ENGINEERING

M.I.T. College of Engineering

PUNE – 411038

CERTIFICATE

This is to certify that theproject report entitled“**Modularizationof Mobile Shopping Assistance System**”

Submitted by

SHUBHAM JAKHETIA	**- B80384319**
TEJ PRATAP SINGH	**- B80384332**
SOURABH SHUBHAM	**- B80384327**

is a record of bonafide work carried out by them, under my guidance, in partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering(Computer) at M.I.T. College of Engineering, Pune under University of Pune.

Date: Place: PUNE

Prof. Prasad Halgaonkar	**Prof. R. K. Bedi**
Guide,	**Head, Dept. of Computer Engineering,**
M.I.T. College of Engineering	M.I.T. College of Engineering
Pune – 411038	**Pune – 411038**

The Project entitled

“Modularizationof Mobile Shopping Assistance System”

By

TEJ PRATAP SINGH	**- B80384332**
SHUBHAM JAKHETIA	**- B80384319**
SOURABH SHUBHAM	**- B80384327**

Is approved of the degree of

BACHELOR OF ENGINEERING-COMPUTER

Savitribai Phule Pune University, Pune

Examiners: 1.____

2.____

Date:

Place:

ACKNOWLEDGEMENT

The success of this project depends largely on the encouragement and guidelines of many. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would also like to thank our Head of the Department, Prof.R.K.Bedi for providing us the opportunity to present this report.

We are grateful to our project guide Prof. Prasad Halgaonkar, for his guidance and whole hearted support and very valued constructive criticism that has driven to complete the project successfully. We feel motivated and encouraged every time we attend her meeting. Without her encouragement and guidance this project would not have materialized.

Last, but not the least, we would like to thank our parents for always encouraging us.
We would like to take this opportunity to thank our friends who were always a source of encouragement.

Shubham Jakhetia
Sourabh Shubham
Tej Pratap Singh

List of Tables:

List of figures:

Table of contents:

ABSTRACT

Shopping assistance systems make a great impact on shopping malls revenue, so considering that in mind, Majority of shopping malls has adopted a shopping assistance system that suites to their malls architecture. We a proposing a shopping assistance system framework that can be implemented to any architecture (weather it is based in NFC, QR Code, Barcode, Network or a combination of all). This framework is also capable of capturing the new and upcoming mobile technologies.

So, we proposed a Modularized Mobile Shopping System Assistance System which will be supported by different technologies and secured encryption – decryption algorithm for providing safe and secure transaction to ease the shopping process. We are considering possible solutions in the field of Mobile Interaction that can be applied in order to perform these tasks in future applications based on current and upcoming communication and interaction technologies. The tasks are represented by several modules, like shopping list management, shopping basket management and payment. These modules conceptually describe the available opportunities to realize specific parts of a customer shopping process based on several technologies and customer behavior. In particular, a deep insight into solutions based on Near Field Communication (NFC) is provided. It is a substantial advantage for the retailer as it enables the analysis of the benefits and drawbacks of technologies and services. Furthermore, it allows the investigation independent from basic technologies like barcodes. Regardless of the used technology, the modules can be linked with each other which allow a simple integration of several modules in a retailer's infrastructure.

1. Literature Survey

Shopping is an essential part of human life. That’s why many people tried to make the process better. Some tried to enhance the process of information retrieval like barcodes on each product, and others tried to make payment better like using credit card payment, but some people found flaws in those like thermal camera attack. So, it is not possible to stick to a system for long.

To make a system that can bear these changes, we introduce a modular system. In this system the whole shopping procedure is divided into important tasks like information retrieval and payment.

The modules are:

1. Shopping list management
2. Orientation
3. Product information
4. Shopping basket management
5. Coupon redemption
6. Customer loyalty reward
7. Payment

Each module uses a specific set of technologies currently in the market. A new technology can replace the existing one in a module irrespective of the technology being used in next module. That is achieved using a particular set of inputs in each module and there subsequent output will be used as input in next module

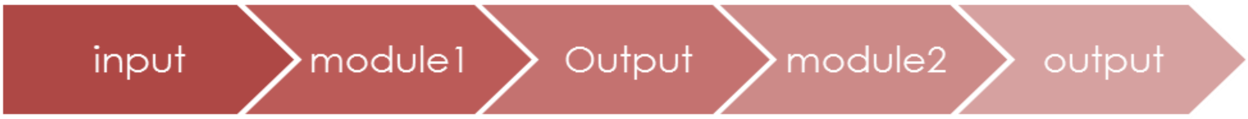


Fig. 1:Modularity

Technologies available to be used with modules are:

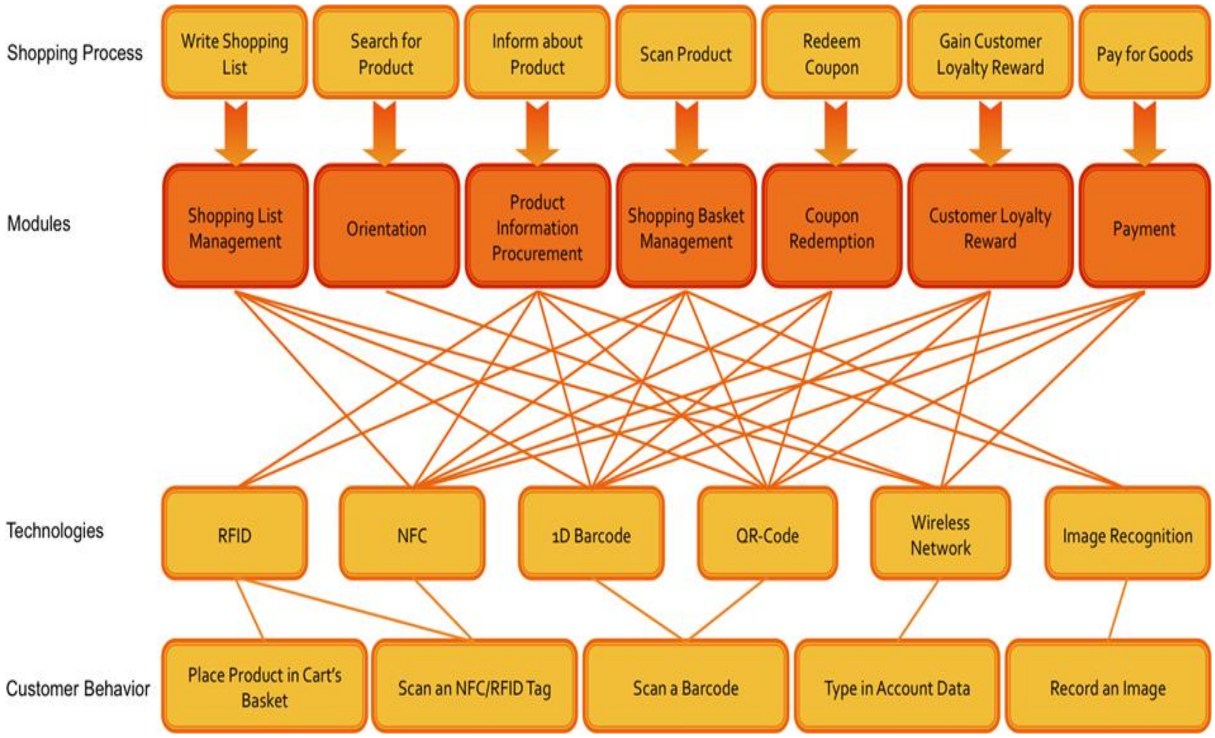


Fig 2 : Tech. available with modules

Each store uses different technology and architecture for its operation. This system can be used in any kind of retailer’s store architecture.

2.Problem Definition

2.1 Problem Statement:-

Overview of all possible mobile solutions that simplify the customer‘s shopping process based on current and upcoming technologies.

2.2 Contribution:-

- This project will help in making shopping procedure more digitalize.
- Shopping procedure will become less stressfull and speedy.
- This project will help in lessning the requirement of man power.

2.3Features of Project:-

- This project is a combination of various application present in market but using more advanced and latest technologies.
- In this project shopping procedure is divided in 7 modules:-
    - Shopping list management.
    - Orientation.
    - Product information procurement
    - Shopping basket management
    - Coupon Redemption
    - Customer loyalty reward
    - Payment

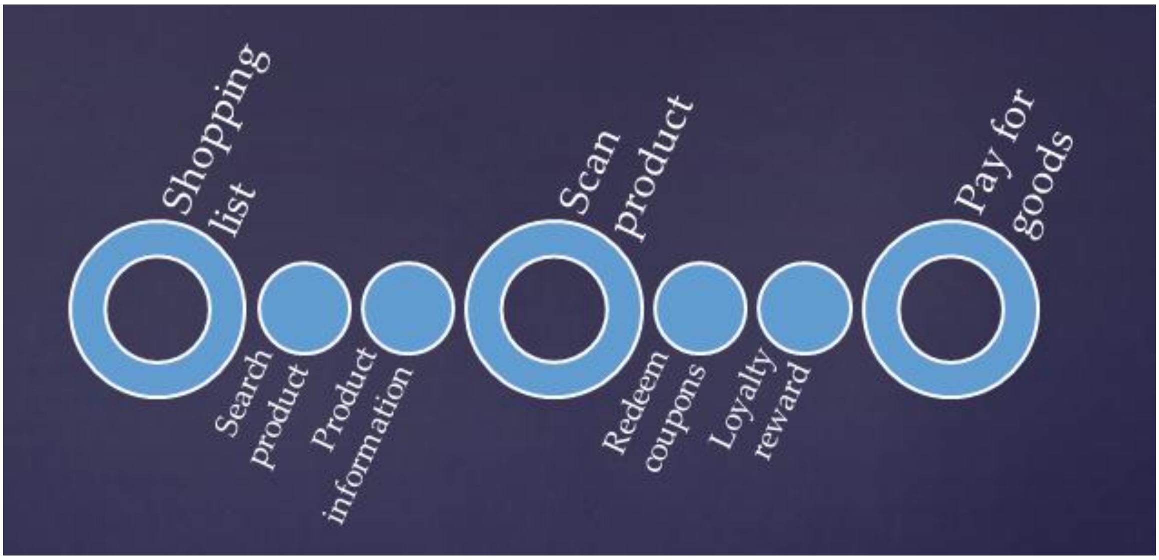


Fig. 3 : Process Workflow

- For managing this modules we are using following technologies:-
  - RFID
  - NFC
  - Barcode
  - QR Code
  - Wireless Network
- It will help in speeding up the shopping procedure because of various technologies used as mentioned above.
- This project requires NFC enabled device at both end ie at customer end as well as at the cashier end or shopkeeper end.
- Individual linkage of modules and technologies/use cases allows easy integration of several modules in a retailer’s infrastructure.

Modules	**Technology**
Shopping List Management	NFC, barcode, qr-code, Wireless network
Orientation	Wireless network
product Information	NFC, barcode, qr-code, Wireless network
Shopping basket Management	NFC, barcode, qr-code
Redeem Coupon	NFC, barcode, qr-code
Gain Customer Loyalty Reward	NFC, barcode, qr-code, Wireless network
Pay for Goods	NFC, barcode, qr-code, Wireless network

Table 1 : Technologies used

3. Platform & Technology

3.1 Platform

Platforms required by both end user/customer and shopkeeper/cashier are same but it is compulsory for end user to use mobile devices.

The application will run on any device supporting Android (version 4.0 or above) or windows (version 7 or above).

- Platform:
  - For End Users/Customer:
    - Android mobile devices.
    - Windows phone mobile devices.
  - For Shopkeeper/Cashier:
    - Android devices.
    - Windows phone devices.
    - Web Browser.

3.2 Technology

Different technologies that are being used in these applications are:-

- Technology:
  - Java: - Java is one of the most used front end programming language and fortunately it is there for world's most selling mobile operating system android. Android provide api's to develop applications which can interact with any of the available hardware such as microphone, camera, Bluetooth, wifi and NFC.
  - Php :- Php is the most used server-side programming language. It is used to create dynamic webpages with the help of mysql database.
  - 1D BARCODE :- They have already found their way into the shopping process. They are printed on nearly every product, which is a substantial advantage for mobile interaction scenarios based on 1D barcodes. Another benefit of 1D barcodes is that they

can be scanned at the most POS.Detecting barcodes with a reader requires a line of sight.

- QR CODE :- QR codes are a special form of 2D barcodes. The biggest advantage of using QR codes instead of 1D barcodes is that they have a higher data capacity. The amount of data which can be stored in the QR code depends on the size (number of rows and columns) of the code.
- RFID :- RFID is a very versatile technology. Assuming that every product will be equipped with an RFID tag in the future, which will probably be the case within the next decade, The technology can be used in nearly every shopping task. The advantage of this technology is that every single tag has a unique identifier. RFID has a range of several meters and does not require a line of sight.
- NFC :- NFC is an extension to the RFID technology which limits the reading range to a few centimeters. An NFC reader allows precise detection of tagged objects because usually only one tag is situated within the small reader range. This opens up a lot of new opportunities in shopping scenarios, especially for product identification and payment.
- Wireless Network :- Different scenarios can also make use of the wireless networks, e.g. Wireless LAN, 3G and Bluetooth. They can e.g. be used for logging in to the retailer's system or to submit bank account data in the payment process.

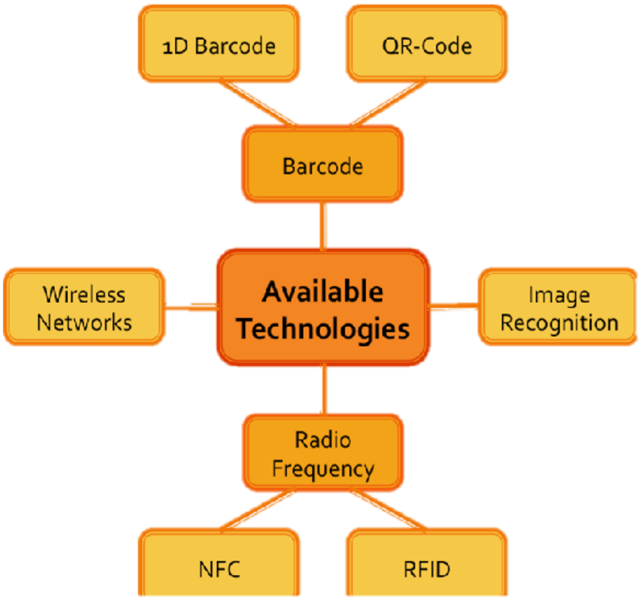


Fig. 4 : Available Technologies

4. SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Introduction

The Software will be based on the project titled *"Modularization Of Mobile Shopping Assistance System".* The software will make use of the concepts developed in the paper. It will demonstrate how mobile and technologies like RFID, NFC, BARCODE etc. can be helpful during shopping. The users of this software system include customer, cashier, helpers, shop owners, bank.

- **Product Overview**

The product is planned to be Java based, operating system independent. The system will be developed modularly, i.e. in modules, and Unified Process software methodology will be followed. In this system tasks are represented by several modules, like shopping list management, shopping basket management payment etc. These modules conceptually describe the available opportunities to realize specific parts of a customer shopping process based on several technologies and customer behavior. In particular, a deep insight into solutions based on Near Field Communication (NFC) is provided. It is a substantial advantage for the retailer as it enables the analysis of the benefits and drawbacks of technologies and services. The system will be developed modularly, i.e. in modules, and Unified Process software methodology will be followed.

4.2 SPECIFIC REQUIREMENTS

4.2.1 External User Interface Requirements

Software Product Features:-

Following features are being provided by the UI.

1. Login system for customers.
2. View the shopping list being managed by him /her.
3. Previous shopping list.
4. Previous shopping bills.
5. Customer loyalty points and coupons.
6. Current shopping bill either generating or generated.
7. His/her bank account informations.

4.2.2 Software System Attributes:-

4.2.2.1 Reliability:-

The software will, by default, be developed to be reliable. This includes development using reliable technologies and software system. We plan to use Java 5.5+, which is being used as industry standard for Java. MySql server will be used, again which is an open standard and widely used. System will be tested continuously to ensure reliability.

4.2.2.2 Availability:-

The system will be continuously available. The server system will be continuously available and waitingfor accepting commands as well as executing commands.

4.2.2.3 Security:-

The system will make use of encryption for authentication purposes like login and accessing data from database.

4.2.2.4 Maintainability:-

The system will be developed modularly, so can be easily maintained. It will be developed into components so that can be easily upgradable and maintainable.

4.2.2.5 Portability:-

System is completely portable to any platform or architecture, since it is developed in Java.

4.2.2.6 Performance:-

Performance is of high priority to us. System will perform to the best and QA control will be done to ensure its performance and efficiency standard.

4.2.3 Database Requirements:-

MySql server will be used as the back end to store tables. JDBC will be used in our application to fire queries to MySql server.

4.3 User Interface:-

The UI will primarily be provided for the client side, Development of UI will be done using Java Swing. This will include rich UI for displaying graphs, menus and flexible UI.

The following core principles will be kept in mind designing UI:

- Flexible
- Consistent
- Easy-to-learn
- Easy-to-remember
- Configurable

4.4 Hardware Interfaces:-

Since a client-server architechture will be used. A hardware interface will be required for connecting client side to server. A LAN needs to be present for the communication to take place. The software can also be run on a virtual server, since our IPv4 address is configurable.

4.5 Software Interfaces:-

Primarily, MySQL database system is required. The software can easily be configured for other database systems. Provided the interfaces are satisfied.

4.6 Communication Protocols:-

TCP/IP the network protocol planned to be used for the client-server communication.

4.7 Change Management:

The requirements may change overtime. It is inevitable. That's why an iterative and agile software development methodology will be used. Unified process is the planned SEP. Changes will beencorporated in each iteration. The details of our project plan are discussed next.

5. SOFTWARE PROJECT PLAN

5.1 Software Development Model: -

Unified Process approach will be used as the primary methodology to develop the software product. This includes four phases

1. Inception
   a. Taking project off the ground.
   b. Deciding the scope of the project
   c. Preliminary risk analysis
2. Elaboration
   a. Detailed requirement analysis
   b. Validation with stake holders
   c. Preparing UML diagrams
   d. Detailed risk assessment
   e. Preparing test cases
   f. Project plan development
   g. SRS development
3. Construction
   a. Prototype development
   b. Software development
   c. Unit testing
   d. Integration testing
   e. Verification and validation
4. Transition
   a. Deployment
   b. Alpha and beta testing
   c. Technical support

Each phase will include the 5 core workflows:

1. Requirements
2. Analysis
3. Design
4. Integration
5. Testing

5.2 Project Scheduling:-

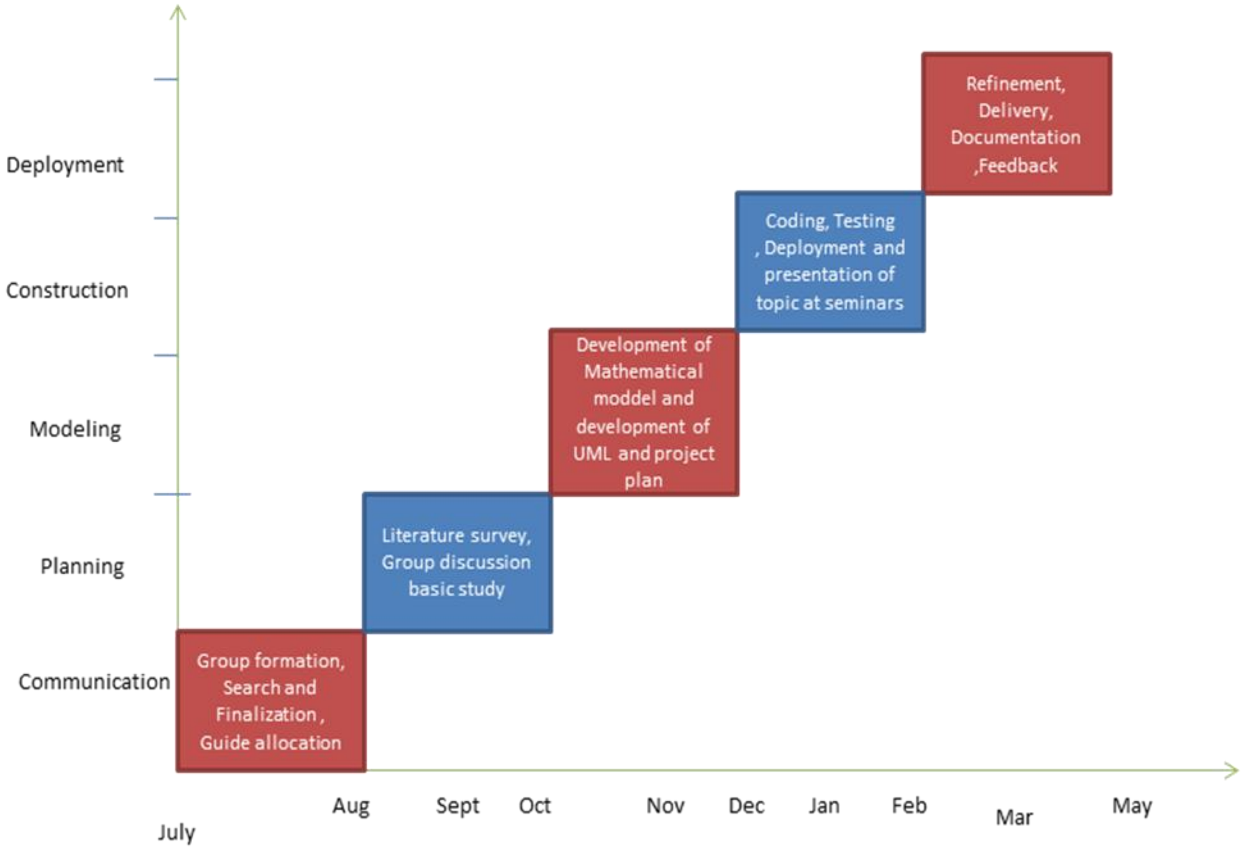

Deployment
Construction
Modeling
Planning
Communication
Group formation, Search and Finalization , Guide allocation
Literature survey, Group discussion basic study
Development of Mathematical moddel and development of UML and project plan
Coding, Testing , Deployment and presentation of topic at seminars
Refinement, Delivery, Documentation ,Feedback
July
Aug
Sept
Oct
Nov
Dec
Jan
Feb
Mar
May

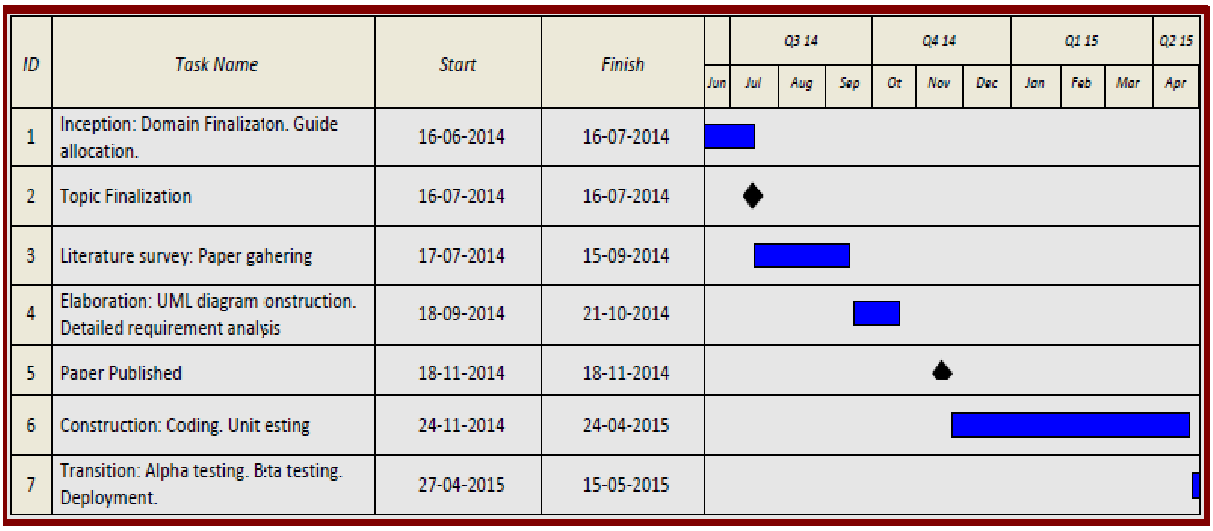
Fig. 5: Software Development Plan

Project schedule in brief is as shown in the figure.

Milestones are

1. SRS completed
2. Database schema development
3. Server developed
4. UI completed
5. Alpha testing
6. Beta testing
7. Product deployed

GANNT CHART:-



ID	Task Name	Start	Finish
1	Inception: Domain Finalizaton. Guide allocation.	16-06-2014	16-07-2014
2	Topic Finalization	16-07-2014	16-07-2014
3	Literature survey: Paper gahering	17-07-2014	15-09-2014
4	Elaboration: UML diagram onstruction. Detailed requirement analysis	18-09-2014	21-10-2014
5	Paper Published	18-11-2014	18-11-2014
6	Construction: Coding. Unit esting	24-11-2014	24-04-2015
7	Transition: Alpha testing. B:ta testing. Deployment.	27-04-2015	15-05-2015

Figure 5 :Gantt chart for project Development

5.3 Risk Management:-

Every project involves risk, however small it may b. While developing the project, we believe in the principle “if you don’t attack risks actively, they will attack you!”. That’s why, a RMMM plan has been developed.

6. High Level Design

6.1 Block Diagrams:-

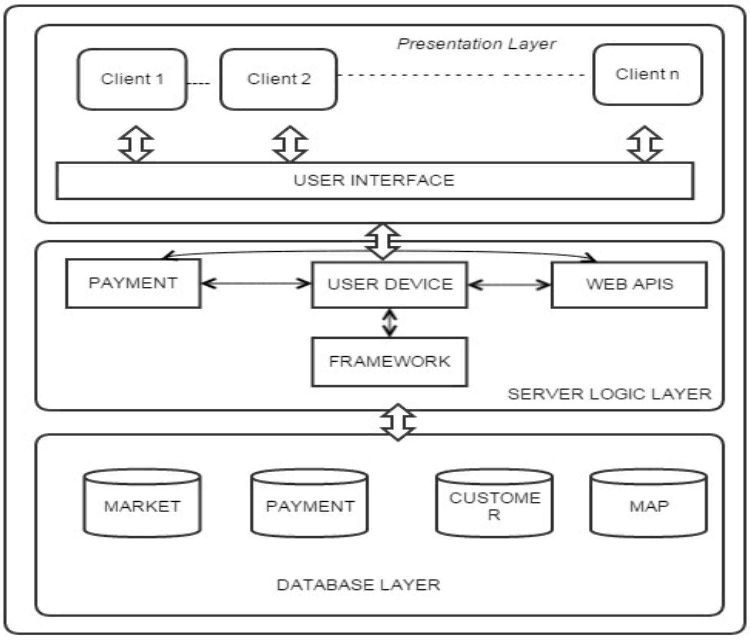


Fig. 7 : Block Diagram

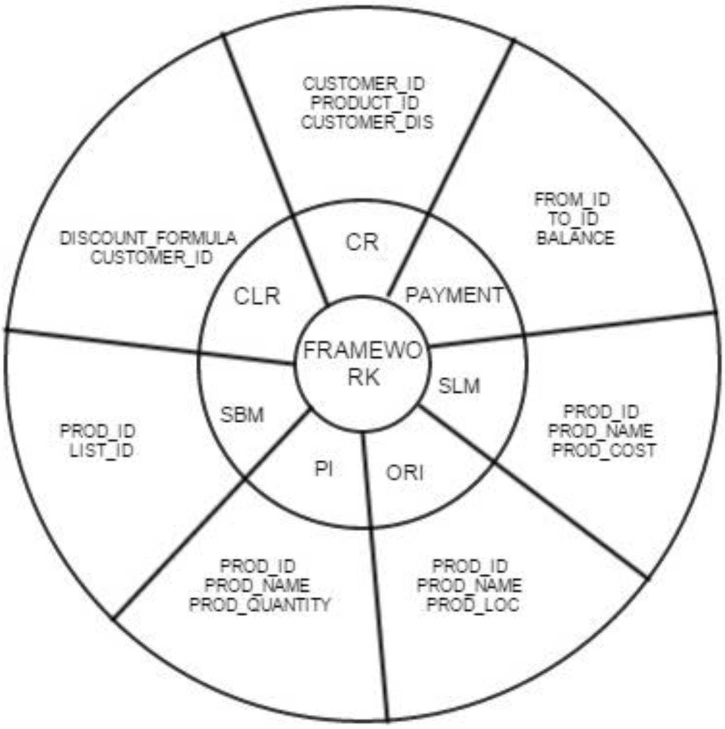


Fig 8 : Framework

6.2 Three-Tier Architechture:-

Presentation Tier:

- The top-most level of the application is the User Interface.
- The main task of this layer is to translate tasks that user can understand.

Server Tier:

- This layer co-ordinates the application, processes, commands, make legal decisions and perform calculations.

Data Tier:

- Here information is stored and retrieved from a databse or file system.

6.3 UML Diagrams:-

- Use Case Diagram

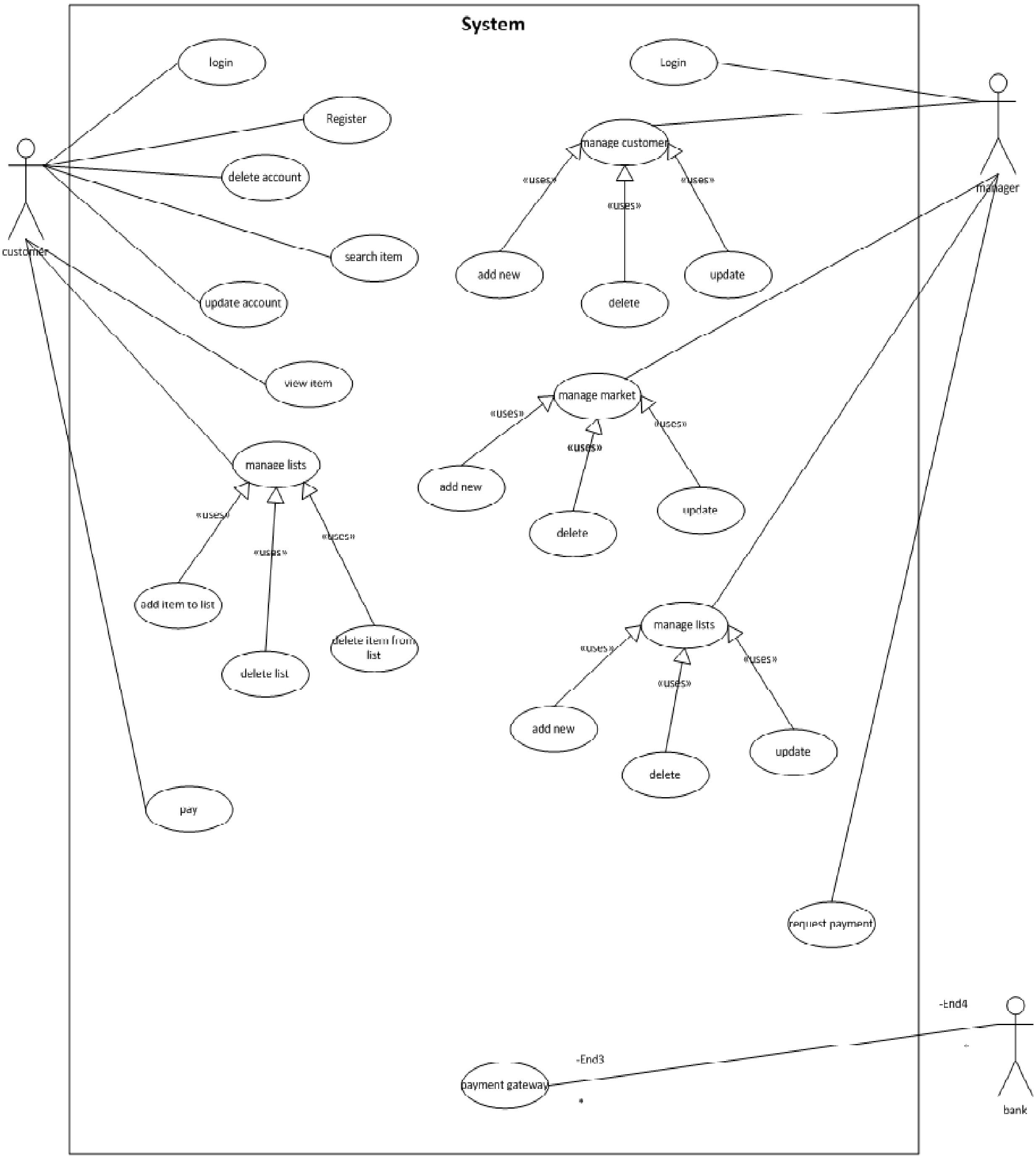


Fig 9 : Use case diagram

Activity Diagram.

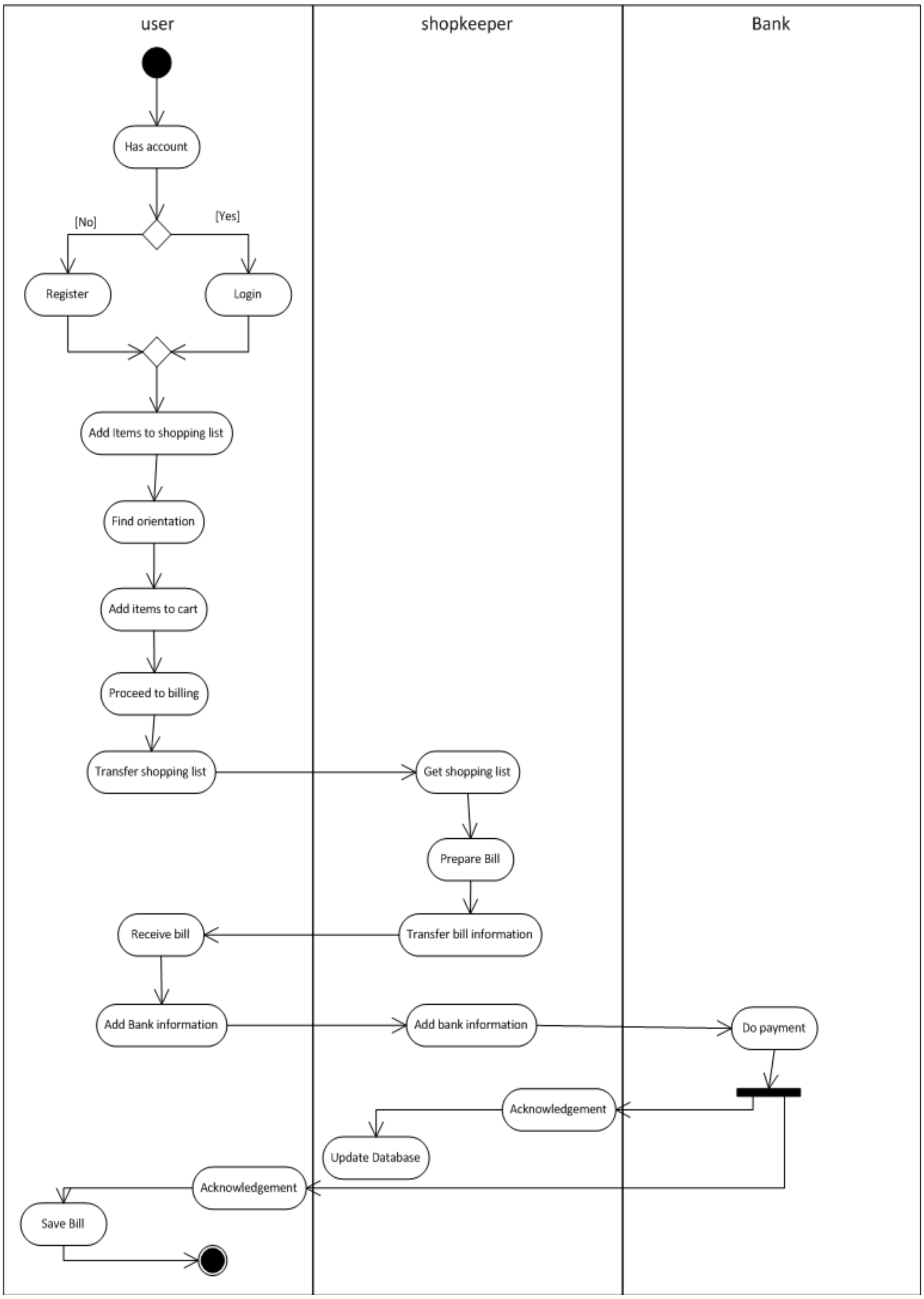


Fig. 10 : Activity Diagram

Class Diagram.

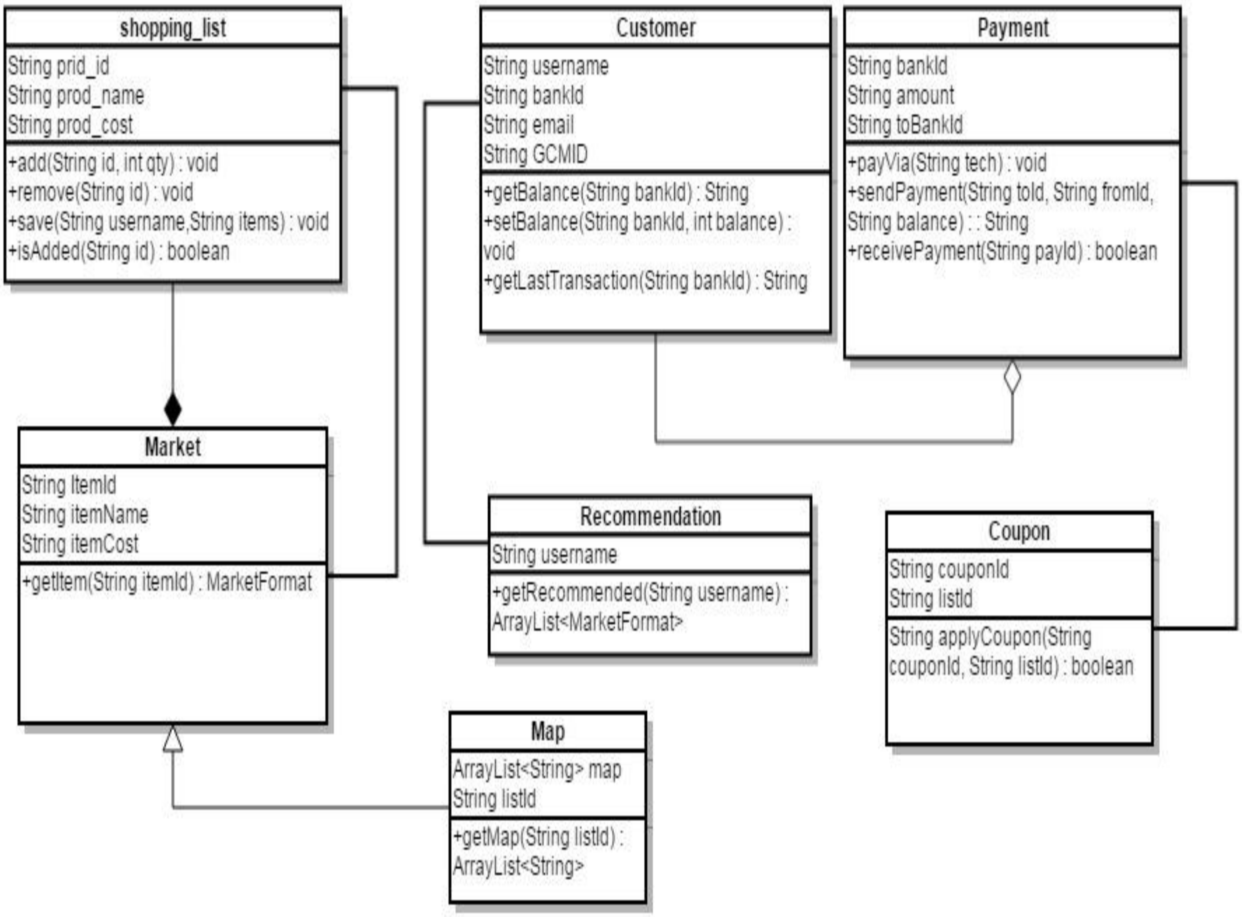

shopping_list
String prid_id
String prod_name
String prod_cost
+add(String id, int qty) : void
+remove(String id) : void
+save(String username,String items) : void
+isAdded(String id) : boolean
Customer
String username
String bankId
String email
String GCMID
+getBalance(String bankId) : String
+setBalance(String bankId, int balance) : void
+getLastTransaction(String bankId) : String
Payment
String bankId
String amount
String toBankId
+payVia(String tech) : void
+sendPayment(String toId, String fromId, String balance) : : String
+receivePayment(String payId) : boolean
Market
String ItemId
String itemName
String itemCost
+getItem(String itemId) : MarketFormat
Recommendation
String username
+getRecommended(String username) : ArrayList<MarketFormat>
Coupon
String couponId
String listId
String applyCoupon(String couponId, String listId) : boolean
Map
ArrayList<String> map
String listId
+getMap(String listId) : ArrayList<String>

Fig 11 : Class diagram

Component Diagram.

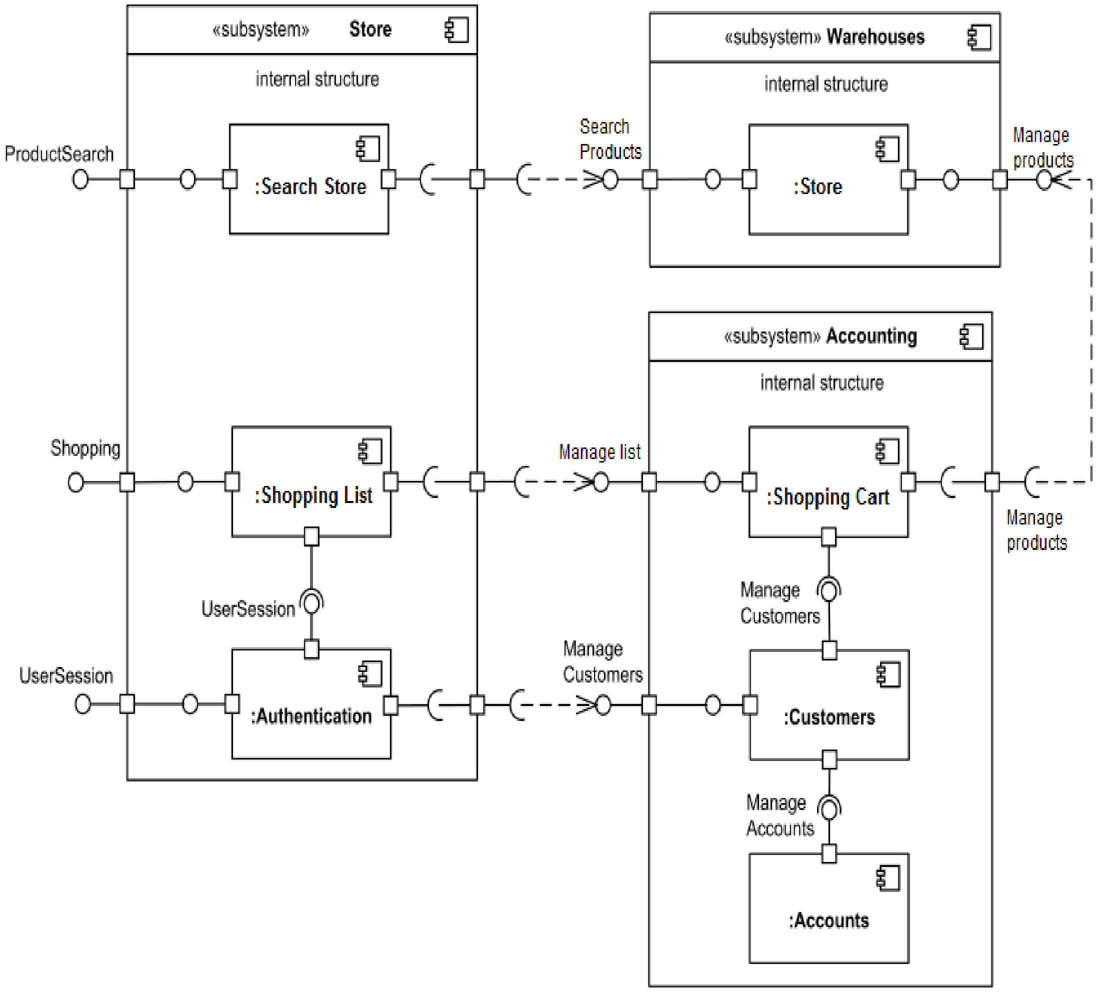
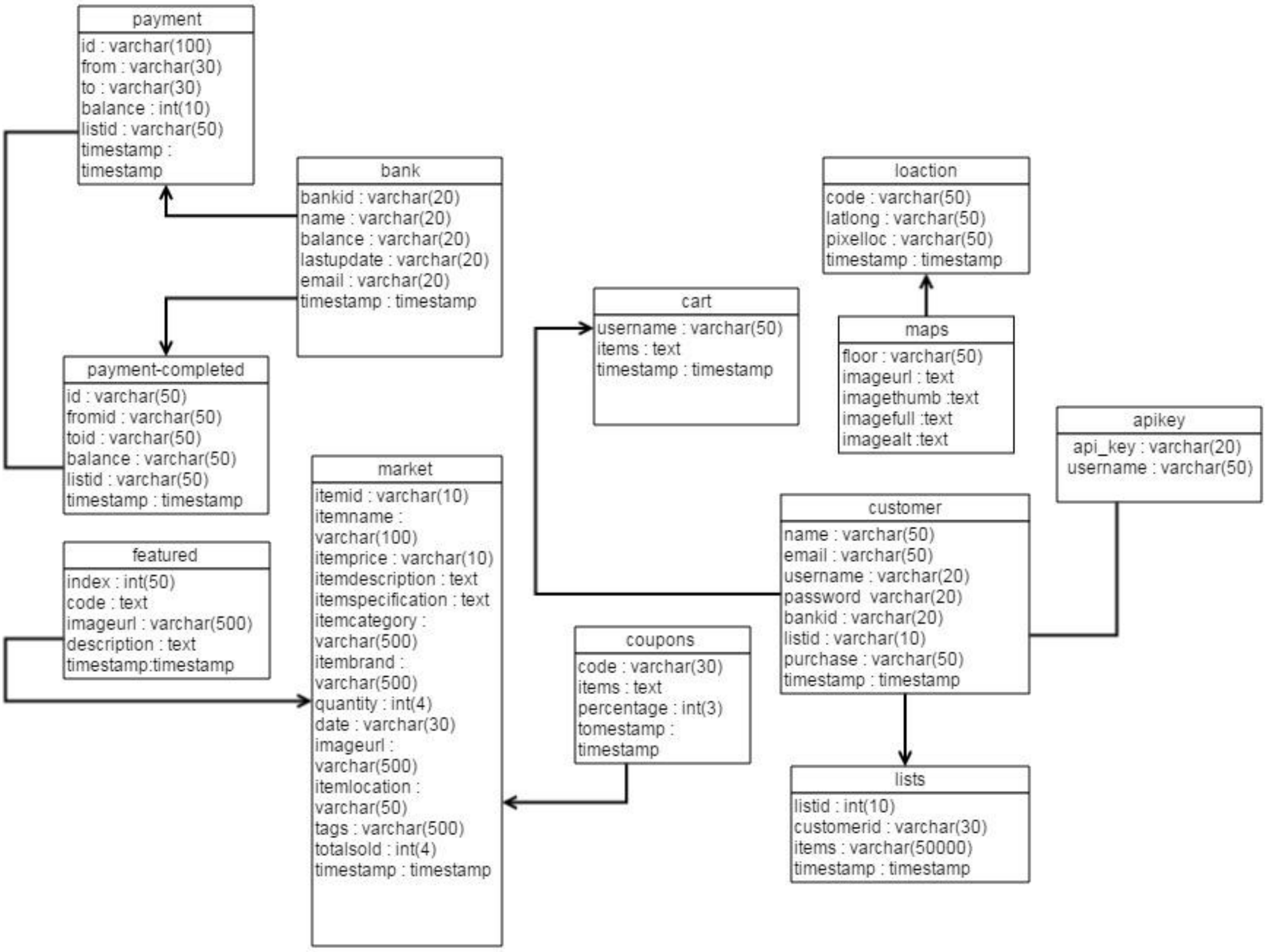


Fig 12 : Component Diagram

6.1Database Designs

- **UserDB. (Server Side)**



- **CustomerDB.** (Client Side)

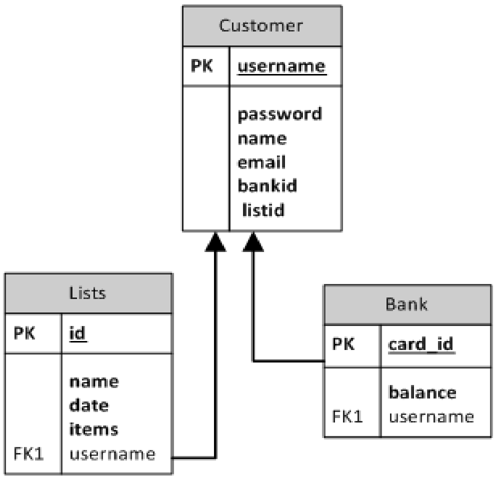


Fig 13 : Database designs

7. IMPLEMENTATION

7.1Project Workstation Selection

All modules are implemented in Java, Android and PHP and records are stored in MySQL.

7.2Installation setup

- **ADT installationsteps:**

StartingtheDownload

1. Goto https://developer.android.com/sdk/index.html
2. Find Download Android Studio.
3. ClicktheDownloadbuttonforthedownloadoptionthatyouwanttoinstall.
4. Savetheinstallerfiletoyoursystem.

Installingthe Software

FollowtheinstructionsinthissectiontoinstalltheIDEonyoursystem.Theseinstallation instructionsapplytoallsupportedplatforms.Forthelistofsupportedplatformsandsystem requirements,seethereleasenotes.

To set up Android Studio on Windows:

1. Launch the .exe file you just downloaded.

2. Follow the setup wizard to install Android Studio and any necessary SDK tools.

   On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

   Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable JAVA_HOME that points to your JDK folder, for example C:\Program Files\Java\jdk1.7.0_21.

The individual tools and other SDK packages are saved outside the Android Studio application directory. If you need to access the tools directly, use a terminal to navigate to the location where they are installed. For example:

\Users\<user>\sdk\

To set up Android Studio on Mac OSX:

1. Launch the .dmg file you just downloaded.

2. Drag and drop Android Studio into the Applications folder.

3. Open Android Studio and follow the setup wizard to install any necessary SDK tools.

Depending on your security settings, when you attempt to open Android Studio, you might see a warning that says the package is damaged and should be moved to the trash. If this happens, go to **System Preferences > Security & Privacy** and under **Allow applications downloaded from**, select **Anywhere**. Then open Android Studio again.

If you need use the Android SDK tools from a command line, you can access them at:

/Users/<user>/Library/Android/sdk/

To set up Android Studio on Linux:

1. Unpack the downloaded ZIP file into an appropriate location for your applications.
2. To launch Android Studio, navigate to the android-studio/bin/ directory in a terminal and execute studio.sh.

   You may want to add android-studio/bin/ to your PATH environmental variable so that you can start Android Studio from any directory.
3. If the SDK is not already installed, follow the setup wizard to install the SDK and any necessary SDK tools.

   Note: You may also need to install the ia32-libs, lib32ncurses5-dev, and lib32stdc++6 packages. These packages are required to support 32-bit apps on a 64-bit machine.

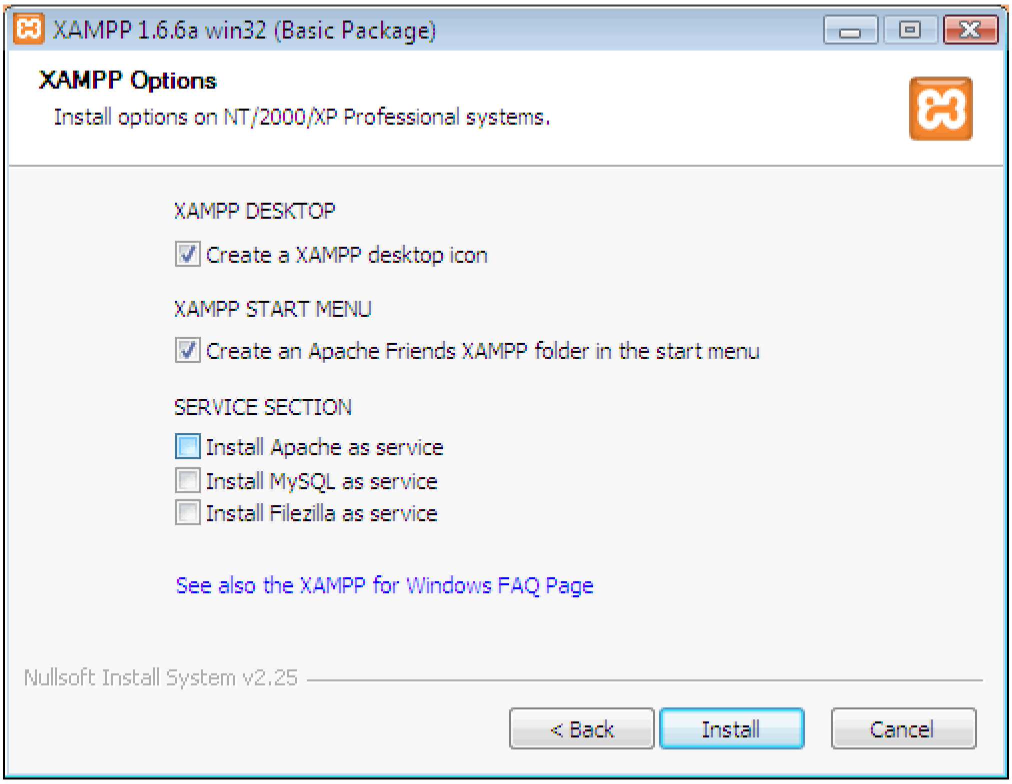
XAMPP installationsteps:

1. Download the software from: http://www.apachefriends.org/en/xampp-windows.html#641 SelecttheInstalleroptionundertheBasicPackage.Youmaybe takentoapage thatpresents youwithabunchofdifferentdownloadlocations.Justclickoneofthedownloadbuttons,and then savethefiletoyourdesktop.Oncedownloaded,theinstallerworkslikemostWindows installers.

   InInternetExplorer,youmaygetawarningaboutdownloadingthefile.Clicktheyellow informationbarthatappears abovetheWebpageinIE,andchooseDownloadFile…
2. Double-click the.exe file you downloaded. Awindowopens,askingyoutoselectthelanguage you would like to use Ifawarningdialogappears clickthe"Allow" optiontoinstallXAMPP.
3. Choosea language from the menu, and then click OK.
   ASetupWizardwindowappears, readytostepyouthroughthesetupprocess. InVistayoumayseeamessagewarningyouthatXAMPPmaynotworkwheninstalledin theC:\Program Filesdirectory.ThedefaultinstallationisinC:\XAMPPsoyoudon’thaveto worryaboutthis problem.
4. Click the Next button.

Theinstallersuggests puttingtheapplicationonyourmaindriveatC:\XAMPP.Youcan pretty muchinstallitanywhere,butwiththeVistaoperatingsystemyoumay encounter problemsifyouinstallitinC:\ProgramFile

5. Click the Next button once again.
   TheXAMPPOptionswindowappears(seebelow).Inmostcases,it'sfinetoleaveall the window's checkboxesjustas yousee;seethenotebelowfordetails.



XAMPP 1.6.6a win32 (Basic Package)

XAMPP Options

Install options on NT/2000/XP Professional systems.

XAMPP DESKTOP

☑ Create a XAMPP desktop icon

XAMPP START MENU

☑ Create an Apache Friends XAMPP folder in the start menu

SERVICE SECTION

□ Install Apache as service
□ Install MySQL as service
□ Install Filezilla as service

See also the XAMPP for Windows FAQ Page

Nullsoft Install System v2.25

Fig. 14 : XAMPP Installation

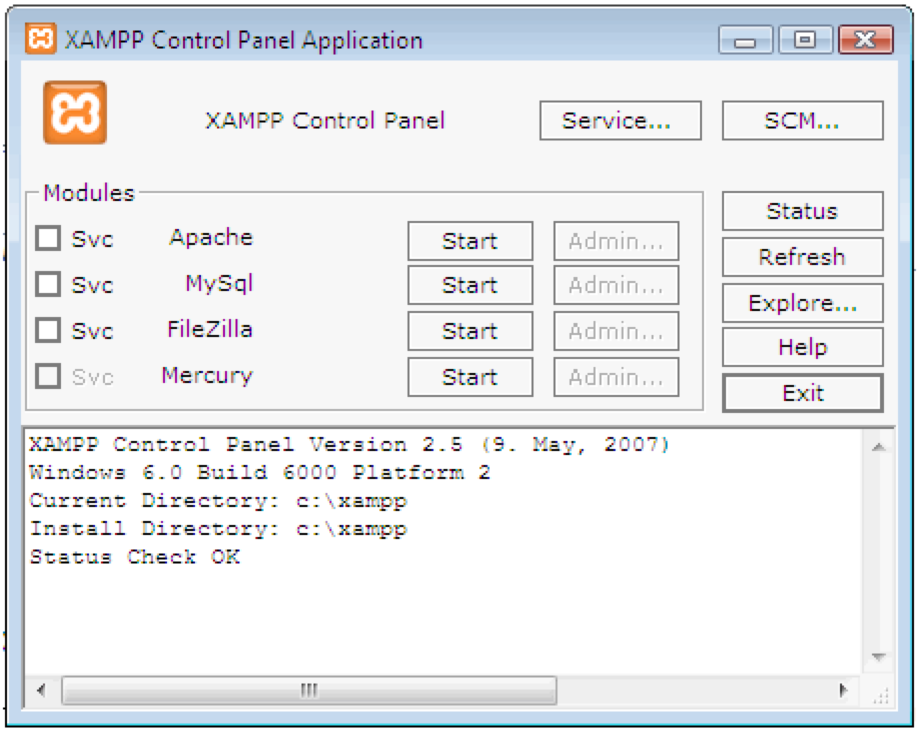
Ifyouplanondoingalotofdevelopment,dayinanddayout,youmightwanttoturnonthe –InstallApacheasservice‖and–InstallMySQLasservice‖checkboxes.A servicestartsup every timeyou turn onyourcomputer,soApache,PHP,andMySQLare alwaysrunning. However,ifyouwon'tbebuildingdatabasesitesfrequently,oryoudon'thavealotofRAM inyourcomputer,don'tturn on theseboxes(you'lljusthavetomanuallystarttheservers whenyouwishtobuilddynamicpages,usingtheXAMPPcontrolpaneldescribedonthe nextpage).

6. Click install.
   Theinstallerplacesall thefilesontoyoursystem.Thisprocesstakesawhile,sincealot of programs andfiles arebeinginstalled.

7. Finally,click the Finish button.
   A windowappears–congratulating‖you(way todouble-clicktheinstallerprogram!),and askingwhetheryouwishtostarttheXAMPPControlpanel.

8. ClickYes, to open the XAMPP Control Panel (seescreenshot below).
   TheXAMPPControlPanelletsyoustartandstoptheApacheWebserverandMySQL

databaseserver.


XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
□ Svc Apache Start Admin...
□ Svc MySql Start Admin...
□ Svc FileZilla Start Admin...
□ Svc Mercury Start Admin...
Status
Refresh
Explore...
Help
Exit

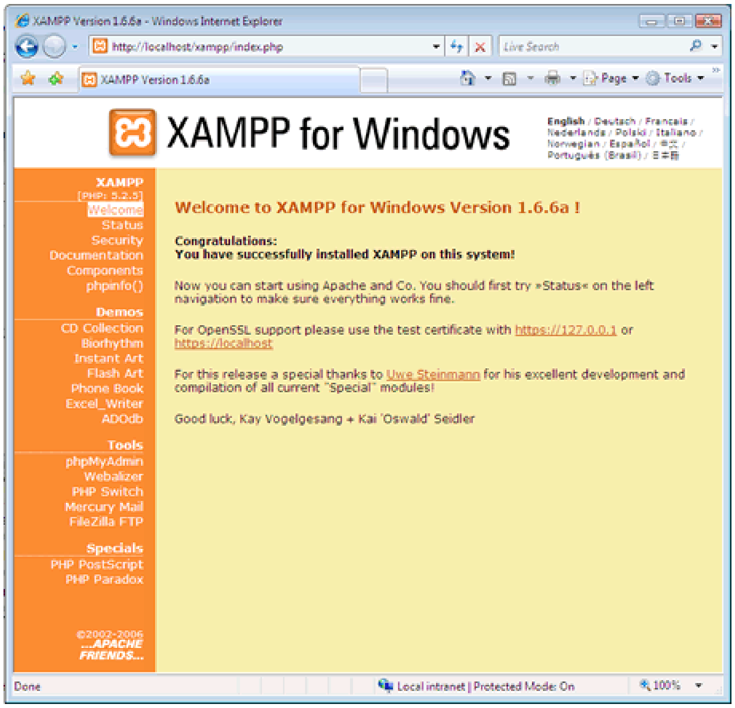
```
XAMPP Control Panel Version 2.5 (9. May, 2007)
Windows 6.0 Build 6000 Platform 2
Current Directory: c:\xampp
Install Directory: c:\xampp
Status Check OK
```

Fig 16 : XAMPP control panel application

In thisfigure,both Apache andMySQLarecurrently NOTrunning,asindicatedby theword Starttotherightoftheirnames.ClicktheStartbuttonstoturn theserverson.Youcan open theControlPanelbyclickingtheXAMPPControlPanelshortcutonyourdesktop.

9. Ifthebuttons totherightofApacheandMySQL sayStart,clickthemtostarttheWebserver and the MySQL database server. You probablyget aWindowssecurityalert about bothMySQLand Apache: Click the Unblock button in bothcases. This action allowsthetwo servers to run, and tells the Windowsfirewallprotectionservicethateverythingis OK.IfApacheand MySQL arealready running,thesebuttonssay Stop.(Clicking themturnsofftheWebserverandMySQL.) WheneveryoustartApache,PHPautomaticallystartsaswell.Atthispoint,youshouldhave acompletetestingserverrunningonyourmachine.Youjustneedtomakesureit’s working.

10. To do so, launch a Web browser, and, in the Location bar, type http://localhost/. Youencounterapagethatlistsabunchoflanguages;clickthelanguageyouprefer,and you’retakentoakindofWeb-basedcontrolpanelforXAMPP(seescreenshotbelow).



Welcome to XAMPP for Windows Version 1.6.6a !

Congratulations:
You have successfully installed XAMPP on this system!

Now you can start using Apache and Co. You should first try »Status« on the left navigation to make sure everything works fine.

For OpenSSL support please use the test certificate with https://127.0.0.1 or https://localhost

For this release a special thanks to Uwe Steinmann for his excellent development and compilation of all current "Special" modules!

Good luck, Kay Vogelgesang + Kai 'Oswald' Seidler

Fig. 16 : XAMPP Interfacae

Onceinstalled,youcanviewyourXAMPPhomepagefromhttp://localhost/xampp/.From the left-hand list of links, you can access helpful programs and information, such as phpMyAdmin(forworkingwith theMySQLdatabase)and phpinfo()forfinding outmore abouttheserversetup.

Onceyou'veinstalledXAMPP,you'llseeashortcutcalledXAMPPControl Panel onyour desktop.Double-clickthisicontocontrol theserversyou'vejustinstalled—youcanturnthe serversoffandon,aswellasturn themintoservices(whichlauncheach timeyoustartup yourcomputer).

USERINTERFACE DESIGN

- Login/Register Page.
- Home Page of Offers and featured items.
- Shopping list management page
- Orientation /map page
- Information Retrieval Page
- List View Page
- Coupon Redemption Page
- Payment Pageobtained

8. Validation of Software

8.1.Introduction

Validation of software is an important part of any project. We have done rigorous validation so that the software produces accurate results as well as it meets the requirements specified. We have also done efficient testing so that the product possesses the qualities of avalibility, scalability, readability, durability. The product has gone through ISO 9001 Quality assurance model. Quality has been basis of the project

8.2 Selection of Project testing tool

As we have used Netbeans to develop entire project we have used, testing tools within Netbeans itself. These testing tools have been used for

- Integration testing
- Unit testing
- Profiling
- Efficiency testing
- JUnit test
- Stress testing
- Load testing

8.3 White Box Testing

In White box testing we have done code coverage testing which includes

- Control flow testing
- Data flow testing
- Branch testing
- Past testing
- Decision coverage

Testing:

TC ID	DESC.	STEPS	ORDER OF EXECUTION	RELATED REQ.	AUTHOR	RESULT
1	Unit Testing	1. Separate modules were checked individually 2.Check the algorithm for proper input andoutput	1.Modules, interfaces, Localdatastructures, Boundary, Conditions, Independent paths	NA	Tej Pratap Singh	Pass
2	IntegrationTesting	1. Check the integrity of the system with other components	NA	Components	Shubham Jakhetia	Pass
3	Coverage testing	1.Each required part of thecodeis reached	1.NA	1. Each part of code is utilized	Tej Pratap Singh	Pass

4	Validationtesting	1.Check whether sentiments Producing correct polarityornot 2.The results predicted is close toactual business or not	1.Datawill sent to and from Server 2.Inputsareprovidedto Framework	1. Server 2.Framework	Sourabh Shubham	Pass

5	Stress testing	1.Check the loadbearing Capacity of system	1.Largeno.ofUsersare provided tothesystem	NA	Shubham Jakhetia	Pass
6.	Load testing	1.The breakpoints are obtained where system fails	1.Processingisdoneuntil systembreaks	NA	Sourabh Shubham	Pass
7.	Usability testing	NA	1.User Friendlessness is Verified.User Caneasilygive inputall fields areclearly mentioned	NA	Sourabh Shubham	Pass

8.	**Recover testing**	**NA**	**1. Process recovery is Checkedafter point of failure.After Failureautomaticallythe streamingstarts.**	**NA**	**Shubham Jakhetia**	**Pass**

Table 2 : Testing

9. RESULT

Test	Precondition	Input	Steps Followed	Result (Pass/Fail)	Expected Output	Actual Output
1	Database Connectivity, phpinstalled	Table name= market, Statement Type=select, Attributes= itemName, itemId, itemPrice, Clauses=where	Enter itemName Clickonsearch	Pass	Queries successfully created	Queries successfully created
2	Database Connectivity, phpinstalled	Table name= coupons, market ,lists Type=select, Attributes= itemId, itemPrice,listId, Clauses=where	Enter Enter Coupon Id And Click On Apply	Pass	Queries successfully created	Queries successfully created
3	Database Connectivity, phpinstalled	Table name= bank,customer Statement Type=select, Attributes=customerId,bankId Clauses=where	Get Amount And Click On Pay	Pass	Queries successfully created	Queries successfully created
4	Database Connectivity, phpinstalled	Table name= lists Statement Type=select, Attributes=listId Clauses=where	Get ListId And Click On Show Map	Pass	Queries successfully created	Queries successfully created

Table 3 : Result

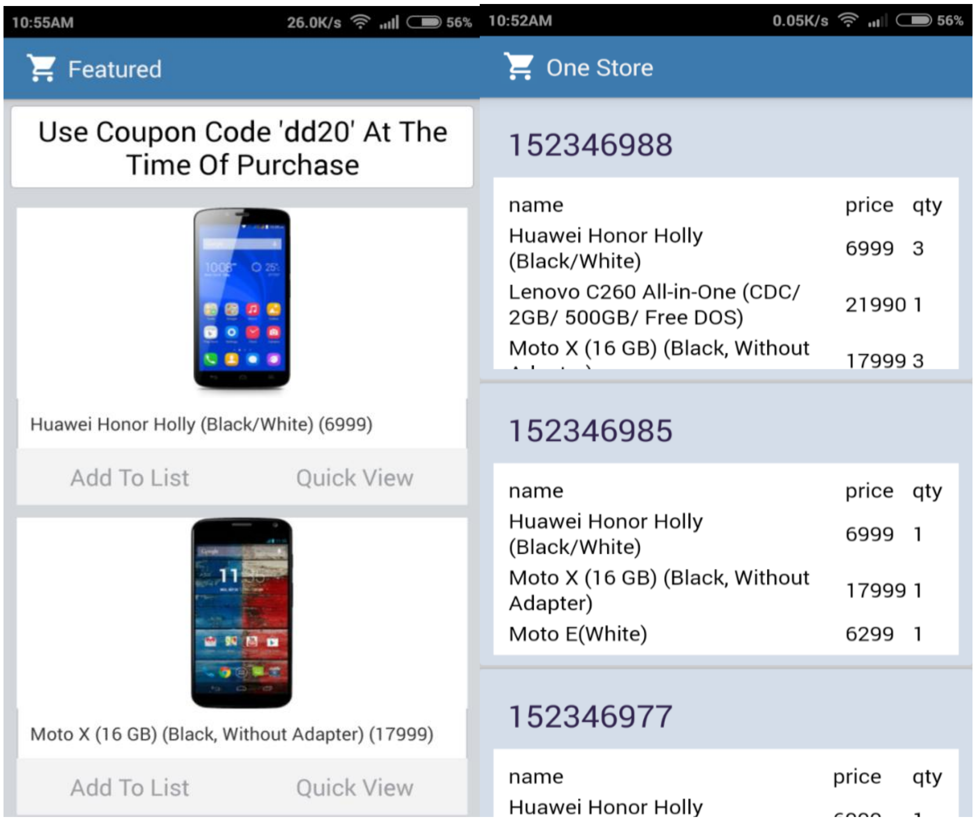
- Test Results on Redmi Note 4G in network Speed of 60 kb/s

Sr No.	Module	Time	Result (%)
1	Getting User Lists	921 ms	Success (100 %)
2	Getting Specific List	505 ms	Success (100 %)
3	Getting Map For List	1044 ms	Success (100 %)
4	Applying Coupon To List	370ms	Success (100 %)
5	Payment	425 ms	Success (100 %)

Fig 17 : Test Results

9.2Screenshots

Customer's Side:



Use Coupon Code 'dd20' At The Time Of Purchase

Huawei Honor Holly (Black/White) (6999)

Moto X (16 GB) (Black, Without Adapter) (17999)

152346988

name	price	qty
Huawei Honor Holly (Black/White)	6999	3
Lenovo C260 All-in-One (CDC/ 2GB/ 500GB/ Free DOS)	21990	1
Moto X (16 GB) (Black, Without	17999	3

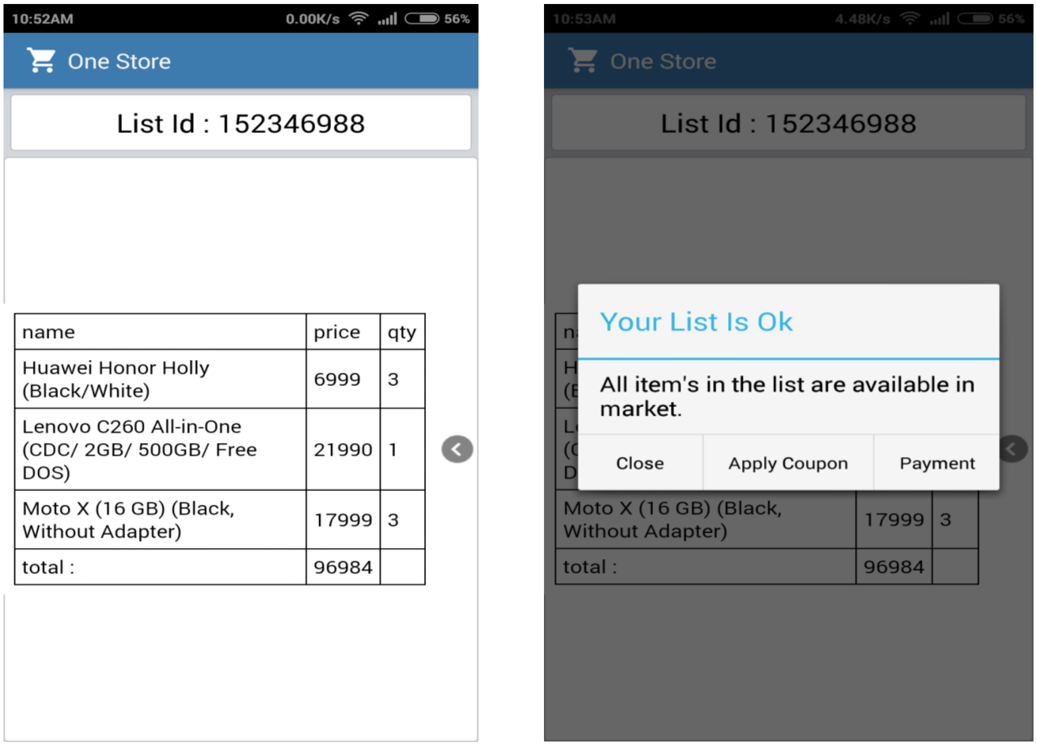
152346985

name	price	qty
Huawei Honor Holly (Black/White)	6999	1
Moto X (16 GB) (Black, Without Adapter)	17999	1
Moto E(White)	6299	1

152346977

name	price	qty
Huawei Honor Holly		

Fig 18 : Featured products and list preparation



10:52AM 0.00K/s 56%

One Store

List Id : 152346988

name	price	qty
Huawei Honor Holly (Black/White)	6999	3
Lenovo C260 All-in-One (CDC/ 2GB/ 500GB/ Free DOS)	21990	1
Moto X (16 GB) (Black, Without Adapter)	17999	3
total :	96984	

10:53AM 4.48K/s 56%

One Store

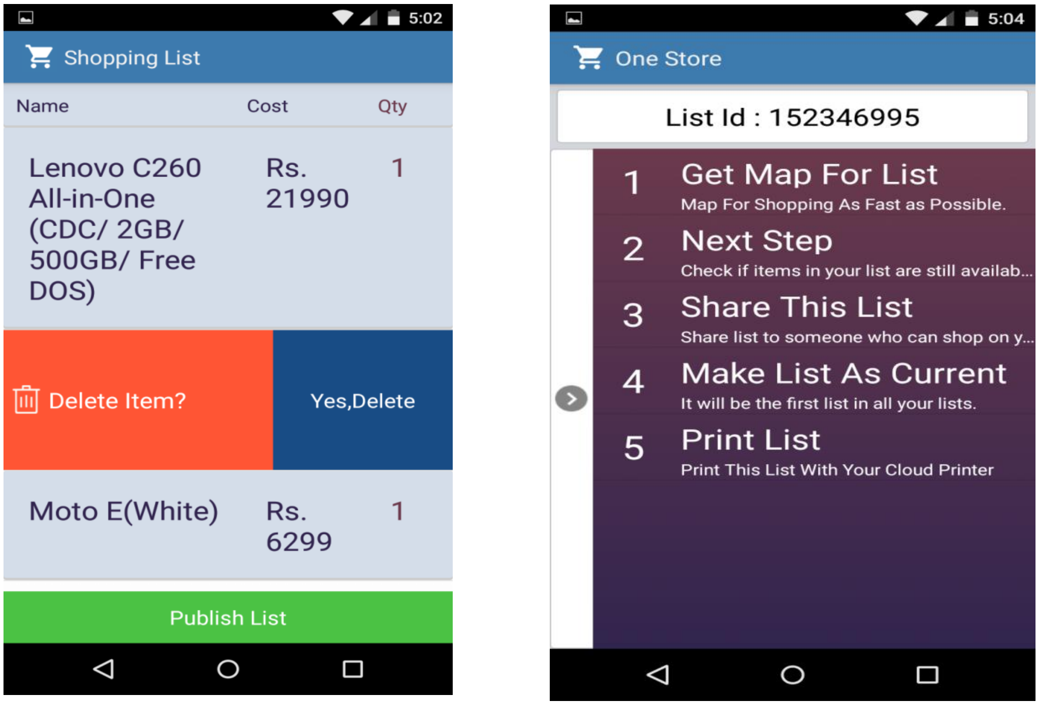
List Id : 152346988

Your List Is Ok

All item's in the list are available in market.

Close | Apply Coupon | Payment

Fig 19 : Check and publish list



5:02

Shopping List

Name	Cost	Qty
Lenovo C260 All-in-One (CDC/ 2GB/ 500GB/ Free DOS)	Rs. 21990	1
Moto E(White)	Rs. 6299	1

Delete Item? Yes,Delete

Publish List

5:04

One Store

List Id : 152346995

1 Get Map For List
Map For Shopping As Fast as Possible.

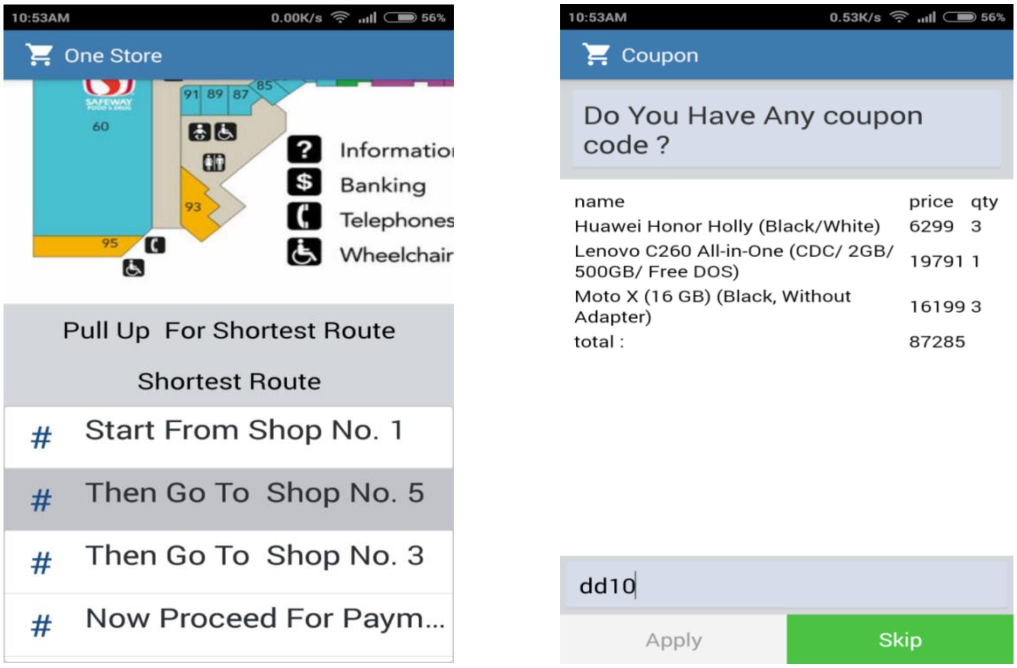
2 Next Step
Check if items in your list are still availab...

3 Share This List
Share list to someone who can shop on y...

4 Make List As Current
It will be the first list in all your lists.

5 Print List
Print This List With Your Cloud Printer

Fig 20 : List operations



Pull Up For Shortest Route

Shortest Route

# Start From Shop No. 1

# Then Go To Shop No. 5

# Then Go To Shop No. 3

# Now Proceed For Paym...

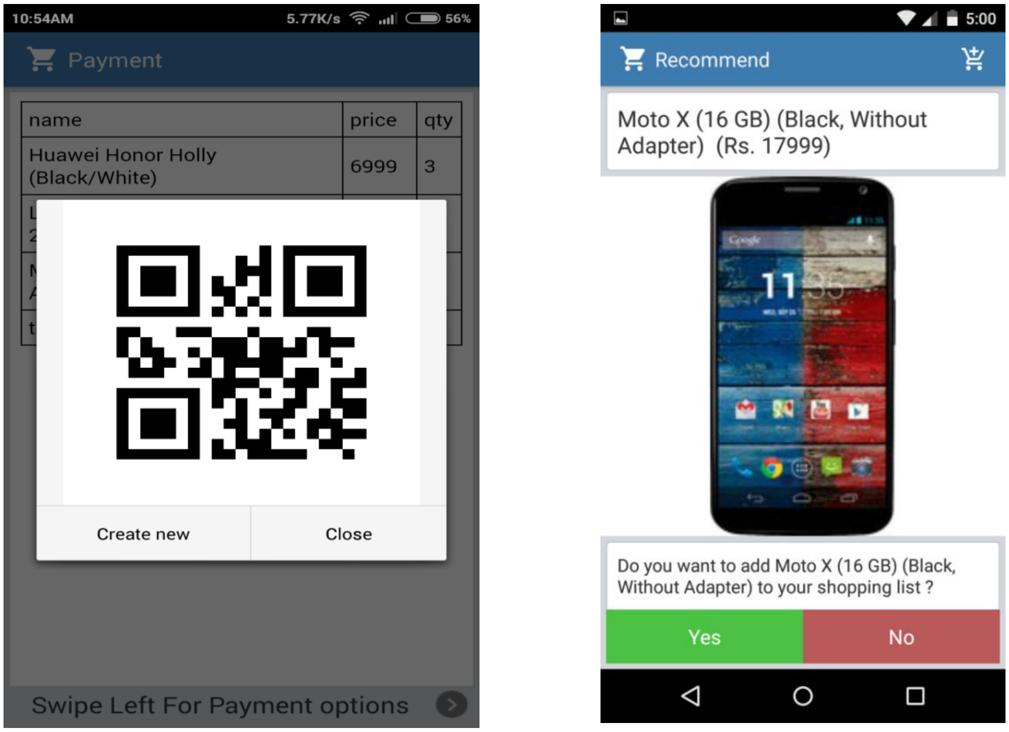
Do You Have Any coupon code ?

name	price	qty
Huawei Honor Holly (Black/White)	6299	3
Lenovo C260 All-in-One (CDC/ 2GB/ 500GB/ Free DOS)	19791	1
Moto X (16 GB) (Black, Without Adapter)	16199	3
total :	87285	

dd10

Apply Skip

Fig 21: Show map and apply coupons



name	price	qty
Huawei Honor Holly (Black/White)	6999	3

Create new Close

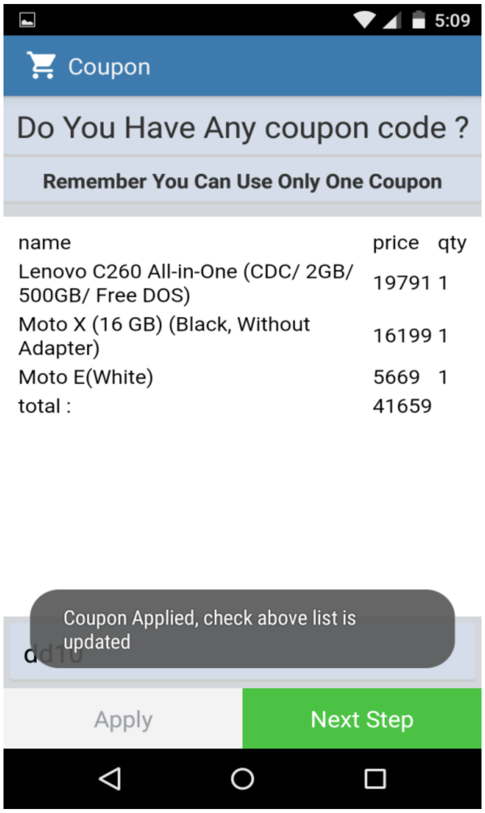
Swipe Left For Payment options

Moto X (16 GB) (Black, Without Adapter) (Rs. 17999)

Do you want to add Moto X (16 GB) (Black, Without Adapter) to your shopping list ?

Yes No

Fig 22 : Generated Barcode



Do You Have Any coupon code ?

Remember You Can Use Only One Coupon

name	price	qty
Lenovo C260 All-in-One (CDC/ 2GB/ 500GB/ Free DOS)	19791	1
Moto X (16 GB) (Black, Without Adapter)	16199	1
Moto E(White)	5669	1
total :	41659	

Coupon Applied, check above list is updated

dd10

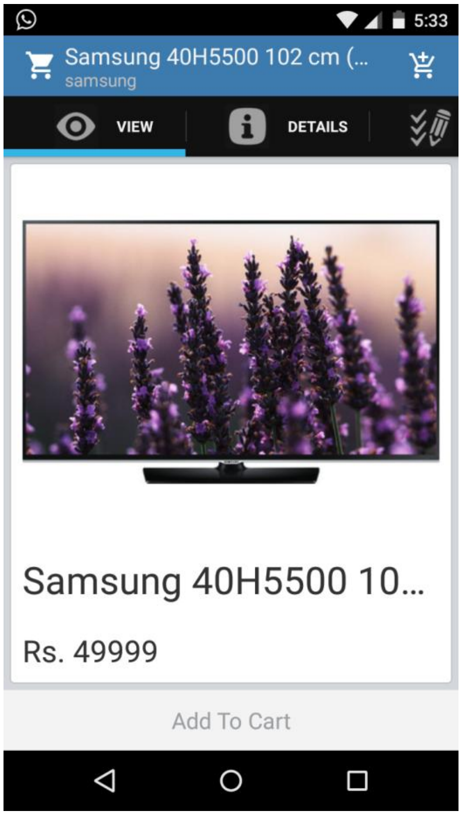
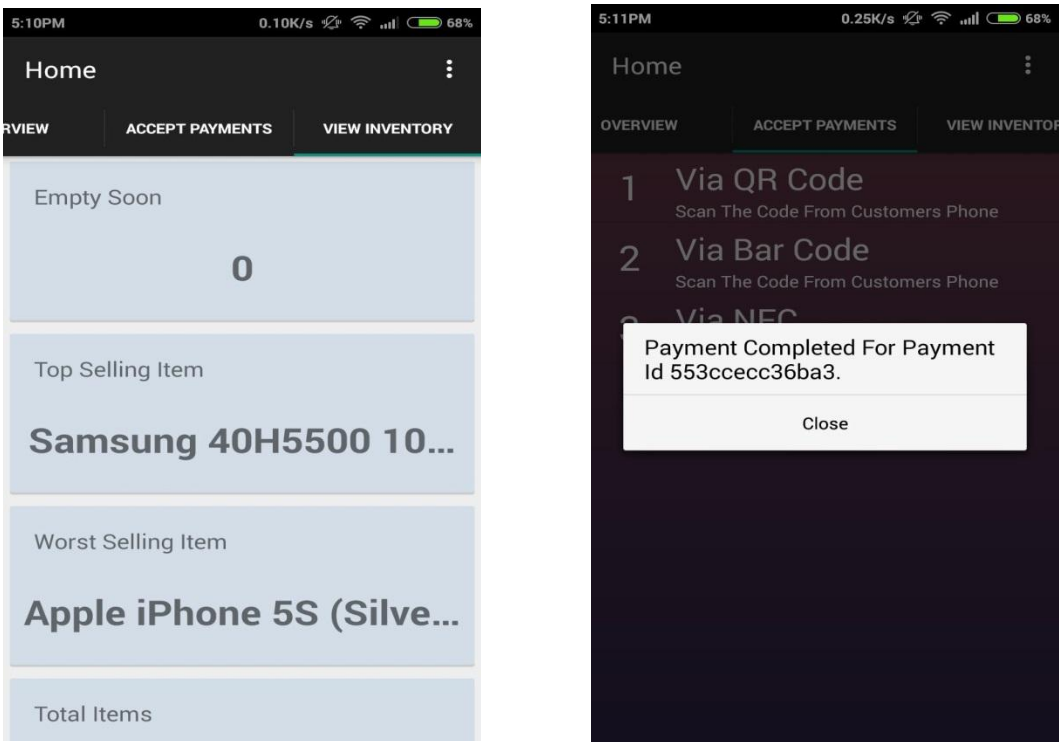


Fig 23 : Coupon Redemption

Retailer's side :



5:10PM 0.10K/s 68%

Home

RVIEW | ACCEPT PAYMENTS | VIEW INVENTORY

Empty Soon

0

Top Selling Item

Samsung 40H5500 10...

Worst Selling Item

Apple iPhone 5S (Silve...

Total Items

5:11PM 0.25K/s 68%

Home

OVERVIEW | ACCEPT PAYMENTS | VIEW INVENTOR

1 Via QR Code
Scan The Code From Customers Phone

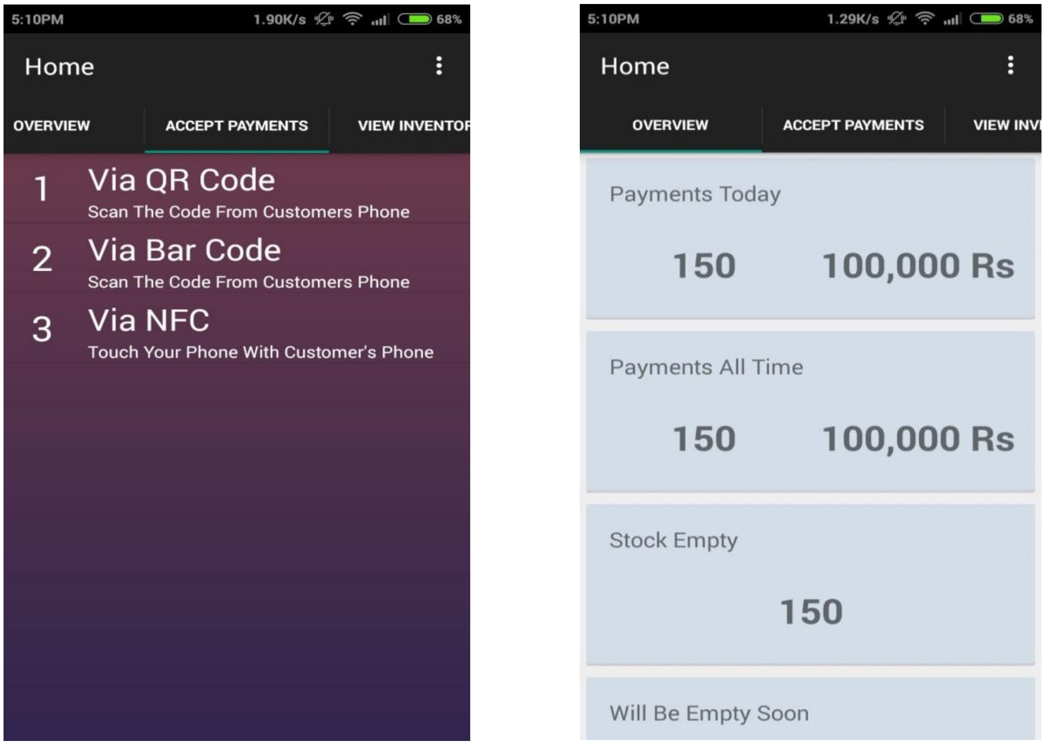
2 Via Bar Code
Scan The Code From Customers Phone

Via NFC

Payment Completed For Payment Id 553ccecc36ba3.

Close

Fig 24 : Items sold and payment



5:10PM 1.90K/s 68%

Home

OVERVIEW | ACCEPT PAYMENTS | VIEW INVENTOR

1 Via QR Code
Scan The Code From Customers Phone

2 Via Bar Code
Scan The Code From Customers Phone

3 Via NFC
Touch Your Phone With Customer's Phone

5:10PM 1.29K/s 68%

Home

OVERVIEW | ACCEPT PAYMENTS | VIEW INVI

Payments Today

150 100,000 Rs

Payments All Time

150 100,000 Rs

Stock Empty

150

Will Be Empty Soon

Fig 25 : Total sale

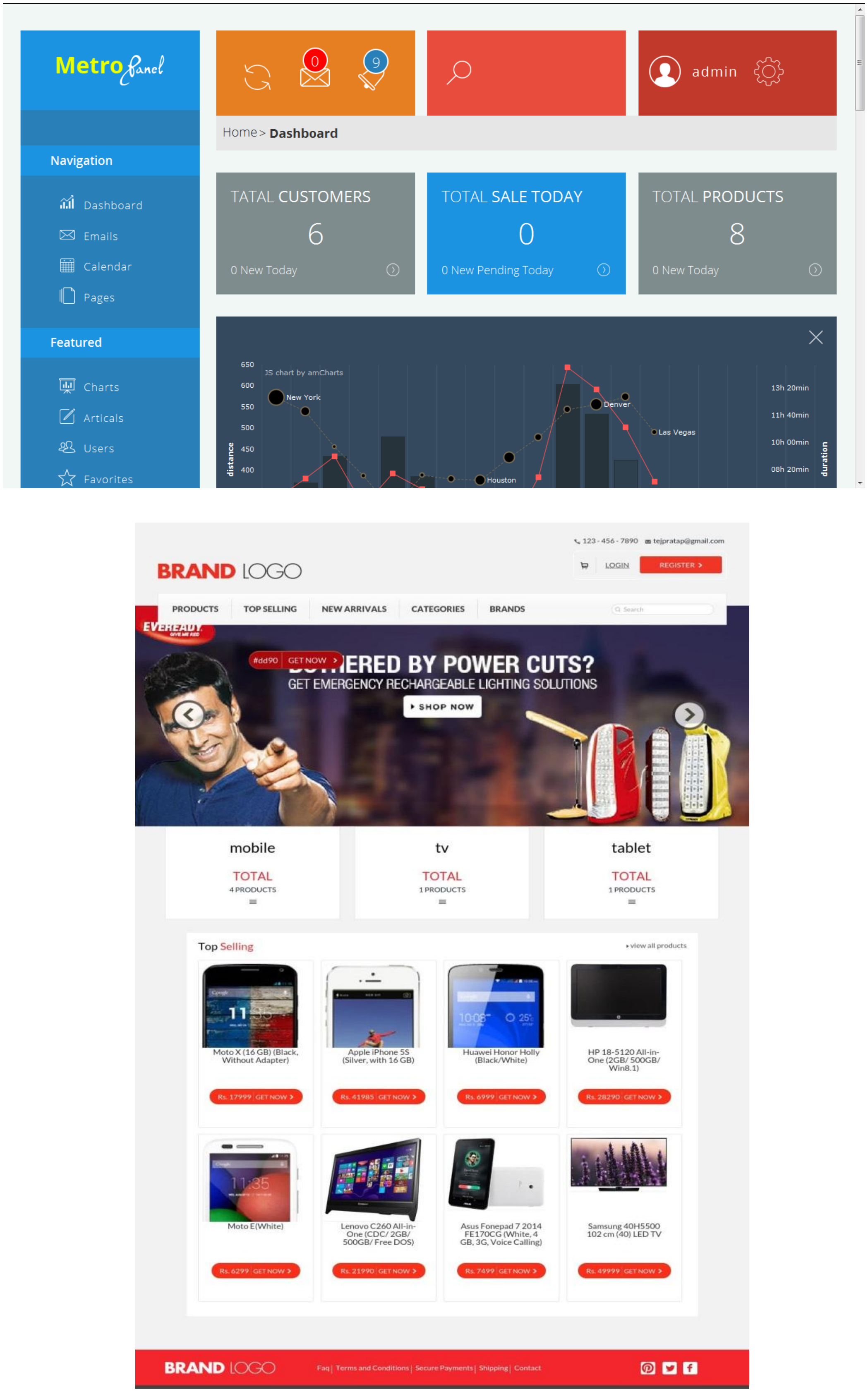


Fig 26: Website Preview

10. CONCLUSION & FUTURE ENHANCEMENT

11.1 FUTURE ENHANCEMENT:-

- This will require more secure encryption to provide more secure transaction.
- Image processing can be used for shopping basket management and bar code reading
- RFID tags can be used for shopping basket management and bill generation.

11.2 CONCLUSION:-

- Customer shopping tasks were mapped to modules.
- Each module can be implemented using a set of different technologies and requires certain customer behavior.
- Module overview helps retailers to identify possible benefits or drawbacks.
- Modules are independent from each other.
- Allows composition of modules (technologies and use cases) that fits to the market’s needs.
- Easy exchange of modules and technologies and easy integration of new upcoming technologies/ use-cases.

ANNEXURE A: Project Analysis

1. Feasibility Assessment:-

Feasibility study mentioned here involves the description of various functionalities of our project.

It includes the Operational Feasibility and Technical Feasibility. In Operational Feasibility, the operations of the project are mentioned. It describes how the project works.
In Technical Feasibility, various technologies used in the project are mentioned.

All algorithms are NP Complete. i.e. they execute in given polynomial time.

1.1 Operational Feasibility

```
 Algorithm I : Brute Force  for TSP (Finding path for user)
      Compute :
      1.Longitude[] = set of all longitudes
      2.Latitude[] = set of all Latitude
      3.Find : array of allocations
      4.Permutation(all Locations)
      5.For all permutation :
      Calculate total distance for that route
      If(current Distance < lowest distance)
      Update lowest
```

Description:-

Takes coordinates in latitude and longitude manner.
Calculate permutation (all Locations);
For all Routes calculate distance
Find The Shortest one

Algorithm 2 : XOR Encryption

Encryption Algorithm:

```
1. k = key for encryption.
2. t = text to encrypt.
3. res = {empty}
4. Loop (length of t)
        4.1 cc = character code of current char of t.
        4.2 res = res + to character from code ( k XOR cc).
5. End
```

Decryption Algorithm:

```
1. k = key for encryption.
2. t = text to decrypt.
3. res = {empty}
4. Loop (length of t)
        4.1 cc = character code of current char of t.
        4.2 res = res + to character from code ( k XOR cc).
5. End
```

Description:

Time complexity = O(n).

Where n is the length of input string.

This works on the basic mathematical principle that if

a XOR **b** equals **c**

then

c XOR **a** equals **b**.

So what happens here is that **a** is the XOR key (or the secret password) and **b** is the string that we want to encrypt.

In order to encrypt our string (**b**) then we convert all the characters in **b** to character codes and XOR them with the XOR key (**a**).

This gives us a number that we convert back to characters.

Now we have a string of characters that is the scrambled result of our string with the XOR key.

In order to decrypt our scrambled string all we need is the XOR key (our secret password). First we convert each character in the scrambled string into character codes and we XOR the result with our XOR key.
Again this gives us a number that we convert back into the original character.

That's it, we now have our original string back.

1.2 Technical Feasibility:-

The Project is technically Feasible using the mentioned technologies:-

1	Programming Language used	JavaScript, java, PHP
2	Hardware	Smartphone with NFC
3	Front end programming	java
4	Technology	NFC, Wireless Network

Table 4. : Technical Feasibility

2. Mathematical Model:-

2.1 For Orientation of product:

We have a situation where user has to find location of a product in the shopping mall, to give direction we are using Travelling Salesmen Problem (TSP).

$D = \min(\text{Array}[\text{all possible paths}])$
$P = O(n)$
i.e. $D = O(P)$

2.2 For sending encrypted data via NFC:

In the shopping process, user has to pay for goods at final stage. To transfer the transfer data securely, we use XOR encryption method.

The following variables are used in XOR algorithm do describe mathematical model.

For Encryption:

t := Plane text to Encrypt.
i := current character of ‘t’ while encryption.
k := Key for Encryption.
res := Cipher text (Encrypted data).

res = res + k XOR t(i)
This equation is iterated for all characters in ‘t’. (1)

Now ‘res’ is in the form of bits, we have to convert it to simple text by converting it to text from ASCII values for each stage in encryption.

For Decryption:

t := Cipher text to Decrypt (res from above equation).
i := current character of ‘t’ while decryption.
k := Key for Decryption (same as used for encryption).
res :=Plain text (Decrypted data).

res = res + k XOR t(i) (2)
This equation is iterated for all characters in ‘t’.

Now ‘res’ is in the form of bits, we have to convert it to simple text by converting it to text from ASCII values for each stage in encryption.

2.3 For Whole Project

Let F be the framework:

Then,

F = {L, O, I, Bm, Cr, Cl, P}

Where, L ,O ,I ,Bm ,Cr ,Cl ,P are the modules of framework

L = Shopping List Management

O = Orientation

Bm = Shopping Basket Management

Cr = Coupon Redemption

Cl = Customer Loyalty Reward

P = Payment

And Let T be the technologies used

Then,

T = {N, B, Qr, Rf, Ir, Nt}

Where,

N = NFC

B = Barcode

Qr = QR Code

Rf = RF Id

Ir = Image Recognition

Nt = Network

Set Theory:-

Each module consist of certain set of tasks, and each task uses a specific set of technologies to operate.

For different modules, set of technologies will be,

L = {N, B, Qr, Nt}

O = {Nt}

I = {N, B, Qr, Nt, Ir}

Bm = {N, B, Qr, Rf}

Cr = {N, B, Qr, Nt}

Cl = {N, B, Qr, Nt}

P = {N, B, Qr, Nt}

Data set used in each module is,

L = {Iid, Pr, Nm}

O = {Lid}

I = {Iid|Nm}

Bm = {Iid}

Cr = {Cid,Lid}

Cl = {CuId}

P = {Lid}

Where,

Iid = Item Id

Nm = Item Name

Pr = Item Price

Lid = List Id

Cid = Coupon Id

CuId = Customer Id

Functions,

Function	Description
getInfo(Iid) => {Iid, Nm, P}	Fetch Info of a Product
getCoupon(Lid, Cid) => {P}	Apply Coupon To List
getPath(Lid) => {Loc1, Loc2 … LocN}	Get Map For A List
getLoyalty(CuId) => {Rank}	Get Loyalty Level Of Customer
Payment([CuId, amm] | payId) => {ack}	Send Payment

Table5 : Functions

Mapping:

Function	Mapping
getInfo(Iid) => {Iid, Nm, P}	One to one
getCoupon(Lid, Cid) => {P}	Many to one
getPath(Lid) => {Loc1, Loc2 … LocN}	One to many
getLoyalty(CuId) => {Rank}	One to one
Payment([CuId, amm] | payId) => {ack}	One to one

Table 6 : Mapping

Venn diagram:

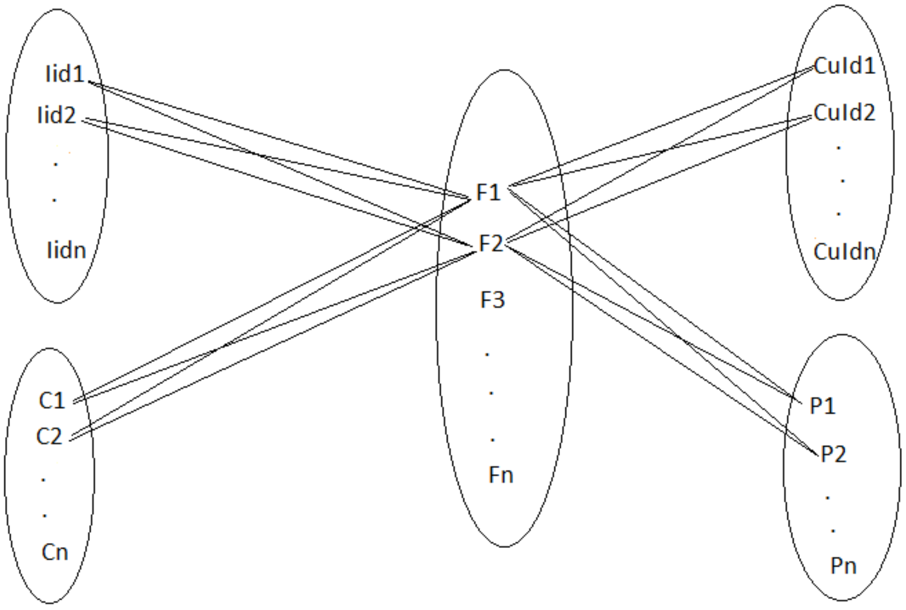


Fig 27 : Venn Diagram

STATE DIAGRAM

Diagram 1:

Consider ‘a’ as starting condition, b is state of selection a valid item, c if item is already added to shopping list, d if item is new to the shopping list, e is state of completing adding process, f is state of getting out of shopping list management process.

Success Condition: if user selected a valid item and the item is either new to his shopping list or already added to his list.

Failure condition: If user selected an invalid item.

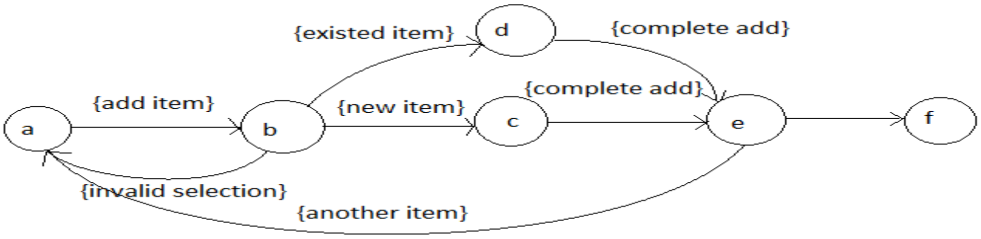


Fig 28 : State Diagram 1

Diagram 2:

Consider ‘a’ is the starting condition, ‘b’ is initialization state where user passes his id and amount to system, c is where the random generated string is encrypted, d is state to send encrypted key to shop, in ‘e’ state shop will decrypt the string, at state f, the decrypted string is matched with the users original string to complete payment.

Success condition:

If original string is matched with shops decrypted key.

Failure Condition:

Either users id is invalid, amount is invalid (i.e. not available in users account) or original string is not matched with shops decrypted key.

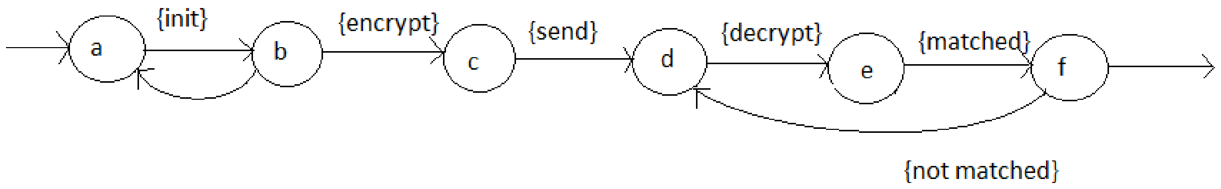


Fig 29 : State Diagram 2

3. Assessment of Requirements of Data:-

OBJECTS

1. User Interface – The user Interface object is used to display the GUI on the screen for the client/customer side of the application. Its functions include: displaying the menu, taking command from the user and passing it to the server side of the application.

2. Login – The login object is used to login to a user account.

3. Register - The register object is used to register or create a new account into the system.

4. Menu – Displays the menu, takes command, maintains the shopping lists, shopping bills and account informations and other functions.

5. CustomerDB – Stores the data of the customer and provides the information on previous and current shopping lists, bills and also provides his\her account information.

6. UserDB – Stores data related to the products available and information related to them also provides customer information and login details

7. LoyaltyPoints – Stores the coupons and points awarded to the customer after every transaction.

4. Functional Relations:

1. The menu object is associates with the CustomerDB database object to receive the data requires for the entire shopping process like list of products to buy, coupons available, amount available, amount payble, bill generation etc.

2. The CustomerDB object depends on customer like new entries to be made or delete old entries etc.

3. Objects login, register and menu depend on the GUI to get respective inputs from the Customer.

4. Login and register also depend on UserDB to store customer related information.

ANNEXURE B

1. Requirement Testing:-

	UC1	UC2	UC3	UC4
REQ 1.(Make List and Store bank account informations (optional).)	X			
REQ 2.(Add reminder for scheduled day and time of shopping.)		X		
REQ 3.(Edit list, refer a list, archieve list.)			X	
REQ 4.(Get list, display coupons, display account information.)				X

Table 7 : Requirement Testing

2. Requirement Traceability Matrix:-

TC ID	DESC.	STEPS	ORDER OF EXECUTION	RELATED REQ.	AUTHOR	AUTOMATED	RESULT
1	Requirement Testing	1. Validate use case against requirement 2. Create a requirement traceability matrix	1.identify use cases 2.check Requirements 3. Fill in the Matrix	NA	Shubham	Partial	Pass
2	Analysis model Testing	Identify all classes are covered or not	1.check all classes attributes and functions	Srs, stakeholder Requirements	Shubham	NA	Pass
3	Unit testing	1. Check the algorithm for proper input and	1. Regression algorithm input and output testing	1. Range of input and output for date	Sourabh	NA	Pass

		output		&account details.			
4	Integration Testing	1. Check the integrity of the system with other components	NA	Components	Sourabh	NA	Pass
5	Validation testing	1.Check whether scheduled dates are correct or not.	1.List will be made and scheduled.	1.List generator.	Sourabh	NA	Pass
6	Stress testing	1.Check the load bearing Capacity of system	1.Inputs will be sent to the system	NA	Shubham	NA	Pass

Table 8 : Requirement Traceability Matrix

3. Reliability Testing:-

Te st id	Precondition	Input	Steps Followed	Result (Pass/Fail)	Expected Output
1	Database Connectivity , phpinstalled	Table name= market, Statement Type=select, Attributes= itemName, itemId, itemPrice, Clauses=where	Enter itemName Clickonsearch	Pass	Queries successfully created
2	Database Connectivity , phpinstalled	Table name= coupons, market ,lists Type=select, Attributes= itemId, itemPrice,listId , Clauses=where	Enter Enter Coupon Id And Click On Apply	Pass	Queries successfully created
3	Database Connectivity , phpinstalled	Table name= bank,customer Statement Type=select, Attributes=customerId,bankId Clauses=where	Get Amount And Click On Pay	Pass	Queries successfully created
4	Database Connectivity , phpinstalled	Table name= lists Statement Type=select, Attributes=listId Clauses=where	Get ListId And Click On Show Map	Pass	Queries successfully created

Table 9: Reliability Testing

Annexure C.

1 Individual Contributions

INDIVIDUAL	CONTRIBUTION
Shubham Jakhetia	Literature Survey, UML Diagrams, Architecture, Platformsurvey, Testing, Problem Statement, Report.
Sourabh Shubham	Literature Survey, Platform Survey/Technology, Test Cases, Risk Management /RMMM plan, Problem Statement, Report.
Tej Pratap Singh	Literature Survey, SRS, Feasibility Analysis, Requirement Gathering, Problem Statement, SWOT Analysis, Report.

Table 10 : Individual Contributions

2**. Progress Report:-**

No	NAMEOF MODULE	MODULEDescriptionin brief	Responsible StudentName formodule	Duration of completion
1	**Shopping List Management**	Design of workflow and behavior of framework In shopping list amangement.	Shubham Jakhetia	**16-Dec-2014 to15-Feb-2015**
2	**Orientation**	Find shortest Path in mall	Sourabh Shubham	**15-Jan-2015 to31-Jan-2015**
3	**Server Flow Design**	Contains backend for whole shopping assiatance	Shubham Jakhetia	**01-Jan-2015 to15-Jan-2015**
4	**Server Implementation**	Handlesserver task	Tej Pratap Singh	**15-Feb-2015 to31-Feb-**

5	**Main GIU**	Designing All Activities And Fragments of Shopping Assistance System	Tej Pratap Singh	**16-Dec-2014 to01-Mar-2015**

6	**Web App**	A Web Based Interface To Handle Common operation on serverside	Tej Pratap Singh	**01-Dec-2014 to29-Dec-2014**
7	**Information Feeding**	Feed the server with information in the server	Sourabh Shubham	**01-Mar-2015 to15-Mar-2015**
8	**Information Retrieval Module**	Fetch And Show Information About product using various mobile technologies	Sourabh Shubham	16-Dec-2014 to15-Feb-2015
9	**Coupon Redemption**	Handles task of applying coupon on a list	Tej Pratap Singh	15-Feb-2015 to31-Feb-2015

10	**Customer Loyalty reward**	Providing user with a discount to returning customers, the discount is calculated with a formula	Shubham Jakhetia	15-Jan-2015 to31-Jan-2015
11	**Payment**	Handles Encryption - decryption and methods to pay via different technologies available	Tej Pratap Singh	**15-Feb-2015 to31-Feb-**

Table 11 : Progress Report

3. Participation Details:

3.1 Paper submissions:

1. Tej Pratap Singh, Sourabh Shubham, Shubham Jakhetia, “Modularity Of Mobile Shopping Assistance System”, ‘*International Journal of Computing and Technology (IJCAT)*’, Volume-1 Issue-10, Page No.: 1-5, November 2014.

3.2 Review of submitted paper in last semester

Reviewers:
Work is appreciable.

IMPLEMENTATION

Call API’s to use web framework
Class Name: URL.java

```
package com.brainstrom.onestoreframework.cloud;

public class URLs {
    public static final String HOST_NAME = "ONE STORE";
    public static final String API_ENDPOINT = 
"http://www.nfcstore.vv.si/";
    public static final String API_KEY = "tejpratap";

    public static final int SearchProductById = 0;
    public static final int SearchProductByName = 1;
    public static final int SearchProductByPrice = 2;
    public static final int SearchProductByDescription = 3;
    public static final int SearchProductByCategory = 4;
    public static final int SearchProductByBrand = 5;
    public static final int SearchProductByQuantity = 6;
    public static final int SearchProductByYear = 7;
    public static final int SearchProductByMonth = 8;

    public static final int SearchCustomerByName = 50;
    public static final int SearchCustomerByUsername = 51;
    public static final int SearchCustomerByemail = 52;
    public static final int SearchCustomerBybankid = 53;

    /*
    * Methods To Generate API Call URL with Parameters.
    */

    /*
    * Query The Database.
    */

    public String queryURL(String SQLQuery) {
        return API_ENDPOINT + "query.php?apikey=" + API_KEY + 
"&query=" + SQLQuery;
    }

    /*
    * Market APIs
    */

    /*
    * Get All Categories
    */

    public String marketCategoriesURL() {
        return API_ENDPOINT + "market/market.categories.php?apikey=" +
API_KEY;
    }

    /*
    * Get All Brands In A Specific Categories
    */

    public String marketCategoriesBrandsURL(String category) {
        return API_ENDPOINT +
"market/market.categorybrands.categories.php?apikey=" + API_KEY +
"&category=" + category;
    }

    /*
    * Delete A Item From Market
    */

    public String marketDeleteItemURL(String ItemId) {
        return API_ENDPOINT +
"market/market.categorybrands.php?apikey=" + API_KEY + "&id=" +
ItemId;
    }

    /*
    * Delete All Items From Market
    */

    public String marketDeleteAllItemsURL() {
        return API_ENDPOINT + "market/market.deleteall.php?apikey=" +
API_KEY;
    }

    /*
    * Delete Items From Market In Descending Order Of Their Arrival
    */

    public String marketFreshItemsURL(int page) {
        return API_ENDPOINT + "market/market.fresh.php?apikey=" +
API_KEY + "&page=" + page;
    }

    /*
    * Delete Items From Market In Descending Order Of Their Total
Sales
```

```
    */

    public String marketTopItemsURL(int page) {
        return API_ENDPOINT + "market/market.top.php?apikey=" +
API_KEY + "&page=" + page;
    }

    /*
    * Get Products Without Any Order By
    */

    public String marketGetItemsURL(int page) {
        return API_ENDPOINT + "market/market.showall.php?apikey=" +
API_KEY + "&page=" + page;
    }

    /*
    * Insert Item Into Market
    */

    public String marketInsertItemURL(String itemId, String itemName,
String itemPrice, String itemDiscreption, String itemSpecification,
String itemCategory, String itemBrand, String quantity, String
imageUrl, String itemLocation, String tags) {
        return API_ENDPOINT + "market/market.insert.php?apikey=" +
API_KEY + "&itemid=" + itemId + "&itemname=" + itemName +
"&itemprice=" + itemPrice + "&itemdiscreption=" + itemDiscreption +
"&itemspecification=" + itemSpecification + "&itemcategory=" +
itemCategory + "&itembrand=" + itemBrand + "&quantity=" + quantity +
"&imageurl=" + imageUrl + "&itemlocation=" + itemLocation + "&tags=" +
tags;
    }

    /*
    * Quick Glance Info Of A Product
    */

    public String marketQuickShowItemURL(String itemId) {
        return API_ENDPOINT + "market/market.quickshow.php?apikey=" +
API_KEY + "&itemid=" + itemId;
    }

    /*
    * Search A Product
    */

    public String marketSearchItemByURL(int Type, String q) {
        String ret = "";
        switch (Type) {
            case SearchProductById:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemid=" + q;
                break;
            case SearchProductByName:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemname=" + q;
                break;
            case SearchProductByPrice:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemprice=" + q;
                break;
            case SearchProductByDescription:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemdiscreption=" +
q;
                break;
            case SearchProductByCategory:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemcategory=" + q;
                break;
            case SearchProductByBrand:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itembrand=" + q;
                break;
            case SearchProductByQuantity:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&quantity=" + q;
                break;
            case SearchProductByYear:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&year=" + q;
                break;
            case SearchProductByMonth:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&month=" + q;
                break;
            default:
                ret = API_ENDPOINT +
"market/market.search.php?apikey=" + API_KEY + "&itemdiscreption=" +
q;
        }
        return ret;
    }

    /*
    * Search A Product
    */

    public String marketSearchItemURL(String q) {
        return API_ENDPOINT + "market/market.search1.php?apikey=" +
API_KEY + "&q=" + q;
    }

    /*
    * Get Product Info
    */

    public String marketShowItemURL(String itemId) {
        return API_ENDPOINT + "market/market.search1.php?apikey=" +
API_KEY + "&itemid=" + itemId;
    }

    /*
    * Update Product Info
    * Leave The Parameter as a blank string
    * You Cannot Update ItemId And It Is Used As A Index For item to
update
    */

    public String marketUpdateItemURL(String itemId, String itemName,
String itemPrice, String itemDiscreption, String itemSpecification,
String itemCategory, String itemBrand, String quantity, String
imageUrl, String itemLocation, String tags) {
        return API_ENDPOINT + "market/market.update.php?apikey=" +
API_KEY + "&itemid=" + itemId + "&itemname=" + itemName +
"&itemprice=" + itemPrice + "&itemdiscreption=" + itemDiscreption +
"&itemspecification=" + itemSpecification + "&itemcategory=" +
itemCategory + "&itembrand=" + itemBrand + "&quantity=" + quantity +
"&imageurl=" + imageUrl + "&itemlocation=" + itemLocation + "&tags=" +
tags;
    }

     /*
     * Bank Operations
     */

    /*
    * Delete All Accounts From Table
    */

    public String bankDeleteAllURL() {
        return API_ENDPOINT + "bank/bank.deleteall.php?apikey=" +
API_KEY;
    }
    /*
    * get Info Of Single Account
    */

    public String bankGetInfoURL(String bankId) {
        return API_ENDPOINT + "bank/bank.show.php?apikey=" + API_KEY + 
"&bankid=" + bankId;
    }

    /*
    * get Info Of All Accounts
    */

    public String bankGetAllInfoURL() {
        return API_ENDPOINT + "bank/bank.showall.php?apikey=" + 
API_KEY;
    }

    /*
    * Transfer Amount Between Accounts
    */

    public String bankTransferURL(String fromBankId, String toBankid, 
String balance, String listId) {
        return API_ENDPOINT + "bank/bank.transfer.php?apikey=" + 
API_KEY + "&frombankid=" + fromBankId + "&tobankid=" + toBankid + 
"&balance=" + balance + "&listid=" + listId;
    }

    /*
    * Update Amount Of Account
    */

    public String bankUpdateAmountURL(String bankId, String balance) {
        return API_ENDPOINT + "bank/bank.update.php?apikey=" + API_KEY 
+ "&bankid=" + bankId + "&balance=" + balance;
    }

     /*
     * Cart Operations
     */

    /*
    * Add To Cart
    */

    public String cartAddURL(String username, String items) {
        return API_ENDPOINT + "cart/cart.add.php?apikey=" + API_KEY + 
"&username=" + username + "&items=" + items;
    }

    /*
    * Remove From Cart
    */

    public String cartRemoveURL(String username, int pos) {
        return API_ENDPOINT + "cart/cart.remove.php?apikey=" + API_KEY 
+ "&username=" + username + "&pos=" + pos;
    }

    /*
    * Save Cart As A List
    */

    public String cartSaveURL(String username, int pos) {
        return API_ENDPOINT + "cart/cart.save.php?apikey=" + API_KEY + 
"&username=" + username;
    }

    /*
     * Coupon Operations
     */

    /*
    * Save Cart As A List
    */

    public String couponCheckURL(String coupon, String listId) {
        return API_ENDPOINT + "cart/cart.save.php?apikey=" + API_KEY + 
"&coupon=" + coupon + "&listid=" + listId;
    }

    /*
     * Customer Operations
     */

    /*
    * Get Current List Of A Customer
    */

    public String customerCurrentListURL(String username) {
        return API_ENDPOINT + 
"customer/customer.currentlist.php?apikey=" + API_KEY + "&username=" + 
username;
    }
    /*
    * Delete A Customer
    */

    public String customerDeleteURL(String username) {
        return API_ENDPOINT + "customer/customer.delete.php?apikey=" +
API_KEY + "&id=" + username;
    }

    /*
    * Delete All Customer
    */

    public String customerDeleteAllURL() {
        return API_ENDPOINT +
"customer/customer.deleteall.php?apikey=" + API_KEY;
    }

    /*
    * Login A Customer
    */

    public String customerLoginURL(String username, String password) {
        return API_ENDPOINT + "customer/customer.login.php?apikey=" +
API_KEY + "&username=" + username + "&password=" + password;
    }

    /*
    * Recommended Products For A Customer
    */

    public String customerRecommendedURL(String username) {
        return API_ENDPOINT +
"customer/customer.recommend.php?apikey=" + API_KEY + "&username=" +
username;
    }

    /*
    * Register A New Customer
    */

    public String customerRegisterURL(String name, String username,
String email, String password, String bankid, String gcmid) {
        return API_ENDPOINT + "customer/customer.register.php?apikey="
+ API_KEY + "&name=" + name + "&username=" + username + "&email=" +
email + "&password=" + password + "&bankid=" + bankid + "&gcmid=" +
gcmid;
```

```
    }

    /*
    * Search A Customer By name, username, email etc
    */

    public String customerSearchByURL(int type, String q) {
        String ret = "";
        switch (type) {
            case SearchCustomerByName:
                ret = API_ENDPOINT +
"customer/customer.search.php?apikey=" + API_KEY + "&name=" + q;
                break;
            case SearchCustomerByUsername:
                ret = API_ENDPOINT +
"customer/customer.search.php?apikey=" + API_KEY + "&username=" + q;
                break;
            case SearchCustomerByemail:
                ret = API_ENDPOINT +
"customer/customer.search.php?apikey=" + API_KEY + "&email=" + q;
                break;
            case SearchCustomerBybankid:
                ret = API_ENDPOINT +
"customer/customer.search.php?apikey=" + API_KEY + "&bankid=" + q;
                break;
            default:
                ret = API_ENDPOINT +
"customer/customer.search.php?apikey=" + API_KEY + "&name=" + q;
        }
        return ret;
    }

    /*
    * Search A Customer
    */

    public String customerSearchURL(String q) {
        return API_ENDPOINT +
"customer/customer.recommend.php?apikey=" + API_KEY + "&q=" + q;
    }

    /*
    * Search A Customer
    */

    public String customerGetCustomersURL() {
        return API_ENDPOINT + "customer/customer.showall.php?apikey="
+ API_KEY;
```

```
    }

    /*
    * Update Customer Info
    * You cannot Update Username Or BankId
    * Leave The Parameter Blank If You Don't Want To Update It
    */

    public String customerUpdateURL(String name, String username,
String email, String password, String bankid, String gcmid) {
        return API_ENDPOINT + "customer/customer.update.php?apikey=" +
API_KEY + "&name=" + name + "&username=" + username + "&email=" +
email + "&password=" + password + "&bankid=" + bankid + "&gcmid=" +
gcmid;
    }


    /*
    * Featured/Coupon Products Customer
    */

    /*
    * Get Items Under A Coupon
    */

    public String couponGetCouponURL(String code) {
        return API_ENDPOINT + "featured/featured.show.php?apikey=" +
API_KEY + "&code=" + code;
    }

    /*
    * Delete A Coupon
    */

    public String couponDeleteURL(String code) {
        return API_ENDPOINT + "featured/featured.delete.php?apikey=" +
API_KEY + "&code=" + code;
    }

    /*
    * Delete A Coupon
    */

    public String couponDeleteAllURL() {
        return API_ENDPOINT +
"featured/featured.deleteall.php?apikey=" + API_KEY;
    }
    /*
    * Get All Coupons
    */

    public String couponGetAllURL() {
        return API_ENDPOINT + "featured/featured.showall.php?apikey="
+ API_KEY;
    }

    /*
    * Lists Of Customer
    */

    /*
    * Check If All Items Are Present In Market
    */

    public String listCheckURL(String listid) {
        return API_ENDPOINT + "lists/lists.check.php?apikey=" +
API_KEY + "&listid=" + listid;
    }

    /*
    * Delete A List
    */

    public String listDeleteURL(String listid) {
        return API_ENDPOINT + "lists/lists.delete.php?apikey=" +
API_KEY + "&listid=" + listid;
    }

    /*
    * Delete All Lists
    */

    public String listDeleteURL() {
        return API_ENDPOINT + "lists/lists.deleteall.php?apikey=" +
API_KEY;
    }

    /*
    * Add A New List
    * items should be in format of :
    * <id>1</id><name>Huawei Honor Holly
(Black/White)</name><quantity>1</quantity><cost>6999</cost>
    */

    public String listInsertURL(String username, String items) {
        return API_ENDPOINT + "lists/lists.insert.php?apikey=" +
API_KEY + "&username=" + username + "&items=" + items;
    }

    /*
    * Get All Lists By A User
    */

    public String listByUserURL(String username) {
        return API_ENDPOINT + "lists/lists.listsby.php?apikey=" +
API_KEY + "&username=" + username;
    }

    /*
    * Set List To Newest Of User
    */

    public String listSetCurrentURL(String username, String listid) {
        return API_ENDPOINT + "lists/lists.setcurrent.php?apikey=" +
API_KEY + "&username=" + username + "&listid=" + listid;
    }

    /*
    * Share A List To Another User
    */

    public String listShareURL(String toUsername, String listid) {
        return API_ENDPOINT + "lists/lists.share.php?apikey=" +
API_KEY + "&customerid=" + toUsername + "&listid=" + listid;
    }

    /*
    * Get A Single List By A User
    */

    public String listShowURL(String listId) {
        return API_ENDPOINT + "lists/lists.show.php?apikey=" + API_KEY
+ "&listid=" + listId;
    }

    /*
    * Get All Lists
    */

    public String listShowAllURL() {
        return API_ENDPOINT + "lists/lists.showall.php?apikey=" +
API_KEY;
    }
    /*
    * Update Market When A List Is Sold
    */

    public String listSoldURL(String listId) {
        return API_ENDPOINT + "lists/lists.sold.php?apikey=" + API_KEY
+ "&listid=" + listId;
    }

    /*
    * Control Push Notifications To User
    */

    /*
    * Update Market When A List Is Sold
    */

    public String notificationSendURL(String listId) {
        return API_ENDPOINT + "lists/lists.sold.php?apikey=" + API_KEY
+ "&listid=" + listId;
    }

    /*
    * Orientation Stuff
    */

    /*
    * get Map For A Floor Or Segment
    */

    public String orientationShowMapURL(String floor) {
        return API_ENDPOINT +
"orientation/orientation.showmap.php?apikey=" + API_KEY + "&floor=" +
floor;
    }

    /*
    * get Path For A List
    */

    public String orientationGetPathURL(String listId) {
        return API_ENDPOINT +
"orientation/orientation.tsp.php?apikey=" + API_KEY + "&listid=" +
listId;
    }

    /*
    * Payment Methods And Related Stuff
    */

    /*
    * Send A Payment
    * @param : fromId : Bank Id Of Customer Paying
    * toId : Bank Id Of Receiver
    * listId : List For Which Payment Is Being Processed
    */

    public String paymentSendURL(String fromid, String toid, String
balance, String listid) {
        return API_ENDPOINT + "payment/payment.send.php?apikey=" +
API_KEY + "&fromid=" + fromid + "&toid=" + toid + "&balance=" +
balance + "&listid=" + listid;
    }

    /*
    * Receive A Payment
    * @param : Id : Payment Id
    * Bank Id : Bank Id Of Receiver
    */

    public String paymentReceiveURL(String id, String bankId) {
        return API_ENDPOINT + "payment/payment.send.php?apikey=" +
API_KEY + "&id=" + id + "&bankid=" + bankId;
    }
}
```

11. REFERENCE & BIBLIOGRAPHY

[1] Modularization of mobile shopping assistance systems by Paradowski, Denise; German Research Center for Artificial Intelligence Campus D3_2, 66123 Saarbrücken, Germany, Kruger, Antonio.

[2] An NFC-Based Solution for Discount and Loyalty Mobile Coupons by Sanchez-Silos, J.J.; Dept. of Computer. & Numerical Anal., Univ. of Cordoba, Cordoba, Spain.

[3] Mobile Near Field Communications (NFC) "Tap 'n Go" Keep it Secure & Private By Ann Cavoukian, Ph.D. Information and Privacy Commissioner, Ontario, Canada.

[4] Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks by Keaton Mowery, Sarah Meiklejohn, Stefan Savage.

[5] Mobile Sales Assistant: An NFCbased product information system for retailers by Stephan Karpischek, Florian Michahelles.

[6] Based on RFID and NFC Technology Retail Chain Supermarket Mobile Checkout by Fugui Ruan, Daijiang Chen.

[7]Importance Measures for a Modular Software System by Fiondella, L. ; Dept. of Comput. Sci. & Eng., Connecticut Univ., Storrs, CT ; Gokhale ,S.S.

12. GLOSSARY

1. NFC : Near Field Communication
2. RFID : Radio Frequency Identification
3. TCP/IP: Network layer protocol suite
4. UDP: User Datagram protocol
5. JDBC: Open source database connectivity tool for Java applications
6. UP: Unified Process (RUP: Rational Unified Process)
7. SEP: Software Engineering Process
8. RMMM: Risk Management and Mitigation
9. JDK: Java Development Kit
10. SQL: Structured Query Language
11. CMP: Change Management Process
12. UML: Unified Modelling Language
13. SWOT: Strengths Weakness Opportunities Threats
14. JRE: Java Runtime Environment
15. LTS: Long term support
16. SRS: Software Requirements Specification
17. UI: User Interface