



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Tanveer Altaf

ATM AND WEB BANK

Multiplatform Support

Technology and Communication

2014

FOREWORD

First and foremost, to accomplish the task and achieve the target is not possible without effort. The project is the result of excellent hard work from the dedicated perspective. The contribution has transformed a tiny vision into practical application. I would love to dedicate the worth full work to people around and appreciate their throughout encouragement. I owe bundle of thanks to my dears, for guidance throughout the writing process.

Finally, I would like to thank to my family for their support and encouragement during my studies.

Tanveer Altaf

Vaasa, Finland

2/4/2014

1 CONTENTS

1	CONTENTS	1
1	INTRODUCTION	4
	1.1 Motivation.....	4
	1.2 Developed System	4
2	TECHNOLOGY	6
	2.1 HTTP	6
	2.2 HTML and CSS	6
	2.3 JavaScript.....	7
	2.4 JQuery	7
	2.5 PHP.....	8
	2.6 XML (Extensible Markup Language).....	8
	2.7 Arduino	9
	2.8 Arduino Software.....	9
	2.9 UART (Universal Asynchronous Receiver/Transmitter)	9
	2.10 USB (Universal Serial Bus).....	10
	2.11 Java Server Pages.....	11
	2.12 Java Servlet	11
	2.13 Java Beans.....	12
	2.14 MySQL	12
	2.15 Apache Server	12
	2.16 Tomcat Server	14
	2.17 List of External Libraries	15
	2.18 Tools and Platforms	15
3	SOFTWARE DESIGN	17
	3.1 Agile Software Development	17
	3.2 Backlog	18
	3.3 QFD.....	18
	3.4 System Architecture	21
	3.5 Use Cases	23
	3.6 ERD.....	25

4	IMPLEMENTATION	26
4.1	Login User Form and Session	26
4.2	New User Form.....	26
4.3	Search User Form	26
4.4	Upload Files to Database	27
4.5	Download Files from Database.....	27
4.6	Shopping Cart	27
4.7	Concept of Transactions and Graph.....	27
4.8	Concept of Bank Account.....	28
4.9	Loan calculator and technologies.....	28
4.10	Update the Text on the Web	28
4.11	Feedback Form and Visitor Counter.....	29
4.12	Arduino Uno Board.....	29
4.13	Arduino Communication	29
4.14	USB Overcurrent Protection	29
4.15	4 x 4 Keypads.....	29
4.16	Arduino I/O and JSP output.....	30
4.17	Facebook API.....	30
4.18	Google API	31
5	TEST CASE	32
6	CONCLUSIONS	39
	REFERENCES.....	40
	APPENDICES	42
	Appendix 1	42
	Appendix 2	52

LIST OF ABBREVIATIONS

UART Universal Asynchronous Receiver/Transmitter

TTL Transistor-Transistor Logic

RX Receiver

TX Transmitter

I/O Input and Output

LED Light Emitted Diode

CSS Cascading Style Sheets

XML Extensible Markup Language

HTML Hypertext Markup Language

PDF Portable Document Format

XSL Extensible Style sheet Language

ICSP In-Circuit Serial Programming

USB Universal Serial Bus

JSP Java Server Pages

HTTP Hypertext Transfer Protocol

PHP Hypertext Preprocessor

API Application Programming Interface

QFD Quality Function Deployment

ERD Entity Relationship Diagram

MVC Model-View-Controller

1 INTRODUCTION

1.1 Motivation

Information technology is spreading into all aspects of societies and mostly it is broadly expanding into the businesses. Logistics, navigation industry, banking sector, media and security control systems all are examples of the information technology. Precision and accuracy of mathematical algorithms and software performance is significant everywhere and it can cause dangerous outputs in case of failure. The significance of the software system is center point. Consequently, there is an increasing need for the application of strict engineering.

To err is human and erroneous results can never be avoided. Fallible traits of man can never be denied. The radical thing in the system development is to see the behavior of the system by giving the required variables and expect the system to work accordingly. Test cases are the most robust solutions of the system behavior analysis.

The motivated approach towards the ATM and Bank Thesis is the consequence of modern era development in information technology industry.

1.2 Developed System

With the gigantic advancement in the technology and the development in various industries with enhancement of embedded implementations, the technology in a small system named as ATM and Banking for intuitive purposes was implemented in the thesis. The system was developed for learning purposes and it can be enhanced and improved further. The project work has no physical existence, it is developed as a virtual Bank which has to nothing do with the real banking system.

The developed system is about the office work, three kinds of users and administration of users. The system has a touch of shopping cart as well. Moreover the

hardware was added as a small integration of C embedded in the system. The link of hardware to the system is confined.

The project is web based and with client-server approach. The most significant part is the integration of PHP and Java together. The product is more about how to create sessions for particular users and connect the desired data to the database, further to retrieve specific data. The system is used to execute the queries and sub-queries of MySQL.

To make the system more dynamic, text files were used to edit the web site specific pages. On the run files which are editable can be used and saved quickly. Database connectivity is also part of the dynamic system. The administrator can manage the system from anywhere on the web, which makes the system more user-friendly and sustainable.

The system was developed to implement the dynamic use of JavaScript, JQuery, cascading style sheets and HTML on the user end. The combination of these technologies made the system more dynamic and multi oriented.

In specific the project was developed to be more student oriented, it has a capability to distinguish three kinds of users and create sessions for particular pages. Furthermore, the shell script was used to make the file on the run and read data from the serial port.

The hardware part of the system has a small implementation of C embedded technology which has a capability to write the data to the serial port and read from the serial port. The idea is to input data from the keypad and retrieve data from the database. On the software side Java was used to read the data and on the hardware side C embedded was used.

The web based system consists of introduction pages about the particular contents, the ATM connectivity, user login, user management, minor part of graphs, upload to share folder, download from share folder and shopping cart.

2 TECHNOLOGY

2.1 HTTP

Hypertext Transfer Protocol is an application protocol for distributed information system. HTTP is the foundation of data communication for World Wide Web. The hypertext is the structured text that uses logical links between nodes containing text. It is a protocol to exchange or transfer hypertext. The standards of HTTP were coordinated by Internet Engineering Task Force and the World Wide Web consortium. The function of HTTP is to request response in the client server computing model. In this model client sends the HTTP request to the server. The server which provides resources, such as HTML files and other contents returns the response message to client. Reference to /6/.

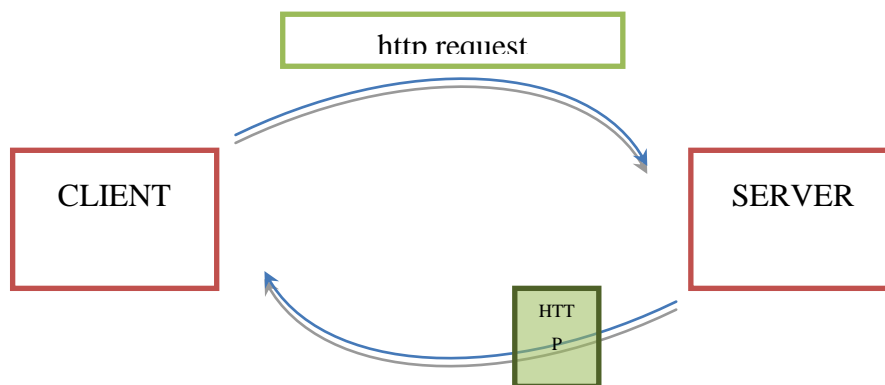


Figure 1. Client and Server HTTP request response

2.2 HTML and CSS

CSS are used to format the web pages so that pages can look sensible for the users. Cascading style sheets are used to make the changes in the appearance of HTML contents. CSS styles can be written in the HTML files or the external CSS files can be linked to the HTML files.

The syntax of CSS styles are embedded inside the `<Style></Style>` tags. The tags which can be accessed through HTML pages can be defined with id or class at-

tributes. In our case CSS was used to hide and seek HTML elements via JavaScript and JQuery. It makes web pages more dynamic when CSS is used in combination of JQuery and JavaScript together.

In the system CSS is used to format the menus, links and particular pages. Forms are manipulated into different shapes with the help of CSS in the system. The following is the way of linking CSS to HTML.

```
<head>
```

```
<link rel="stylesheet" type="text/CSS" href=""></head>
```

2.3 JavaScript

JavaScript is a very powerful scripting language that can be used to develop dynamics in the pages. It is the most commonly used as a part of web browsers. The script can be used to manipulate the contents of the CSS and HTML. The scripting language can be used in multiple types of applications. JavaScript has many names from the Java language. JavaScript can be embedded into the HTML pages by using `<script></script>` tags or can be linked to the external files by using links.

In the project JavaScript was used to develop many pages, the validation criteria was administrated by JavaScript and CSS. The calendar was made by JavaScript and implemented by using JQuery. The animations were purely made in JavaScript. Hide and seek of all the pages for user interface were administered by JavaScript, as well.

2.4 JQuery

JQuery is a very rich JavaScript library to develop web pages. It is an easy to handle library. The HTML contents can be manipulated very simply. Multiple browsers support it, and it can be considered a multitude browser and cross platform library. JQuery is free and accessible to everybody.

The JQuery library can be added by downloading a local copy or directly link over the web.

```
<script src="http://code.jquery.com/jquery-1.11.min.js"></script>
```

JQuery is a small, fast and feature rich JavaScript library. Event handling and animations can be done easily by using the library. Reference to /4/.

2.5 PHP

PHP stands for preprocessor hypertext. PHP is a server scripting language and is a powerful tool for making dynamic and interactive web pages quickly. PHP is widely-used, free and efficient. PHP commands can be embedded directly into the HTML source. Much of its syntax is borrowed from C, Java and Perl. PHP is embedded in between `<? PHP ?>` tags.

In the project the file upload and download were done with PHP. Visitors counter is also running with PHP files. Reference to /5/.

2.6 XML (Extensible Markup Language)

XML is a language meant for a readable format, it is meant for both human and machine. Its design is very simple and according to XML 1.0 Specifications. These specifications are produced by World Wide Web Consortium. Its usability is high over the Internet. The format support is Unicode which is computing industry standard for consistent encoding. XML is widely used for the web services. Hundreds of applications have been developed including RSS, Atom, SOAP and XHTML. Many productivity tools are using XML formats. In the real life it is introduced as data interchange over the Internet. RFC 3023 gives rules for the construction of Internet Media Types for use when sending XML. The characters making up an XML document are divided into markup and content. Reference to /7/.

2.7 Arduino

Arduino is an open-source electronics for end users. It is very flexible and easy to use hardware. The software can be downloaded for free. It was designed to give students an inexpensive and easy way to program interactive objects. An Arduino microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. Reference to /8/.

2.8 Arduino Software

Arduino Software, an Arduino integrated development environment is a cross platform application written in Java. The software is designed to introduce programming to artists and other newcomers unfamiliar with software development. It is very handy and easy to use IDE. Arduino programs are written in C or C++. Arduino IDE comes with a software library called "Wiring". Reference to /9/.

- setup(): initialize settings
- loop(): runs repeatedly

2.9 UART (Universal Asynchronous Receiver/Transmitter)

UARTs are commonly used in conjunction with communication standards. UARTs are commonly included in microcontrollers. UART takes bytes of data and transmits the individual bits in a sequential fashion. Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms. Communications may be simplex, half duplex or full duplex.

Table 1. UART data bits

Bit No	1	2	3	4	5	6	7	8	9	10	11
	Start bit	Data Bits								Stop bit	
	Start	0	1	2	3	4	5	6	7	Stop	

After data is written to the shift register, UART hardware pops up with starts the bit and writes the data bits to the line and appends parity with stop the bit. UART hardware is controlled by a clock signal which runs at a multiple of the data rate.

Reference to /1/.

2.10 USB (Universal Serial Bus)

Digital Equipment Corporation, IBM, Intel, Microsoft, NEC and Nortel are the designers of the Bus and designed in 1996.

The maximum voltage is 5 +/- 0,25 - 0,55 V and the maximum current 0,5 – 0,9 A (general). The bitrate is 1.5/12/480/5000/10000 Mbit/s (depending on the mode).

It can serve 127 devices maximum by using serial protocol. Reference to /2/.

Table 2. USB pins configuration

Pin 1	Vcc (+5 V red wire)
Pin 2	Data (white wire)
Pin 3	Data (green wire)
Pin 4	Ground (black wire)

2.11 Java Server Pages

JSP is the technology that mixes static web pages and dynamic web pages. Compared to the ASP Microsoft specific technology, JSP is more powerful as it is Java based and portable to other platforms. If we compare JSP and JavaScript, JavaScript is more dynamic but there are many things that JavaScript cannot handle such as submission and HTTP. JavaScript cannot access server side resources whereas JSP can access server side resources and it is built for that. The main advantages of JSP are efficiency, convenience, powerfulness, portability and inexpensiveness. In JSP JVM stays up and each request is handled by lightweight Java thread. JSP automatically parse HTML data.

2.12 Java Servlet

Servlets are Java based and supported by all web servers. The most significant part is that they are built for developing dynamic web pages in other way; we can say they perform work of application engine. Three methods are important for the life cycle of servlet.

- service()
- init()
- destroy()

These above mentioned three methods are implemented by all servlets and are invoked at a specific time. To understand the scenario let us take an example. The user enters URL and sends an HTTP request. On the other side the server receives request, now the server maps the request to a particular servlet, the servlet is loaded to the server. The servlet invokes `init()` and it is called once in the beginning, next Servlet invokes the `service()` method and it is invoked on every HTTP request. At the end the server may decide to unload the Servlet from the memory.

2.13 Java Beans

Java Bean makes the software more user-friendly and its usability increases. The reason is that we can be reused the bean without any hindrance or confusions.

2.14 MySQL

MySQL is database software that actually does all the work of storing, retrieving, managing and manipulating data. Phpmyadmin is used to manage the MySQL database. The usability of MySQL is very high, mainly because of its simplicity, cost effectiveness and fastness. It is trusted and is already in use of prestigious organizations and significantly for the performance as it is very fast in response.

```

root@ubuntu:/home/raw# cat /etc/mysql/my.cnf

The MySQL database server configuration file.

You can copy this to one of:
- "/etc/mysql/my.cnf" to set global options,
- "~/.my.cnf" to set user-specific options.

One can use all long options that the program supports.
Run program with --help to get a list of available options and with
--print-defaults to see which it would actually understand and use.

For explanations see
http://dev.mysql.com/doc/mysql/en/server-system-variables.html

This will be passed to all mysql clients
It has been reported that passwords should be enclosed with ticks/quotes
especially if they contain "#" chars...
Remember to edit /etc/mysql/debian.cnf when changing the socket location.
client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

```

Figure 2. MySQL my.cnf file for port address

2.15 Apache Server

Apache is used for PHP pages integrated with JSP. The Apache server can be configured by using file php.ini, Reference to /14/.

```

root@ubuntu:/home/raw# apache2 -v
Server version: Apache/2.2.22 (Ubuntu)
Server built:   Mar 19 2014 21:11:10
root@ubuntu:/home/raw# █

```

Figure 3. Apache Version


```

root@ubuntu:/home/raw# cd /etc/apache2
root@ubuntu:/etc/apache2# ls -F
apache2.conf  envvars      magic        mods-enabled/  ports.conf-   sites-enabled/
conf.d/       httpd.conf  mods-available/  ports.conf     sites-available/  ssl/

```

Figure 4. Apache folder contains

-apache2.conf

The main file for the Apache server. Everything can be configured from this file

-ports.conf

Virtual host ports can be defined in this file.

-conf.d/

SSL can be configured in this file.

-sites-available/

All the virtual host files are included in this.

-Timeout

This is by default set to a particular value that means the server has that time to serve the request.

-KeepAlive

The value set to “on” will allow each connection to remain open, to handle multiple requests.

-Enabling Sites and Modules in Apache

```
sudo a2ensite
```

```
sudo service apache2 reload
```

These two commands are used to enable the sites and reload for proper work.

2.16 Tomcat Server

In the thesis port 8089 was used for HTTP. The Tomcat server was used for Java Server Pages and servlets. The Tomcat server can be easily configured by configuring server.xml file.

Port Name	Port Number
Tomcat admin port	8029
HTTP/1.1	8089
SSL	8443
...	...

Figure 5. Tomcat port address

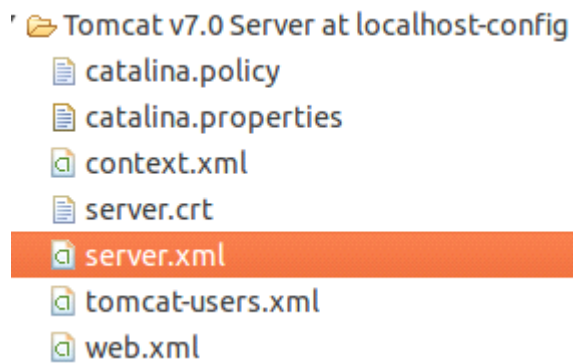
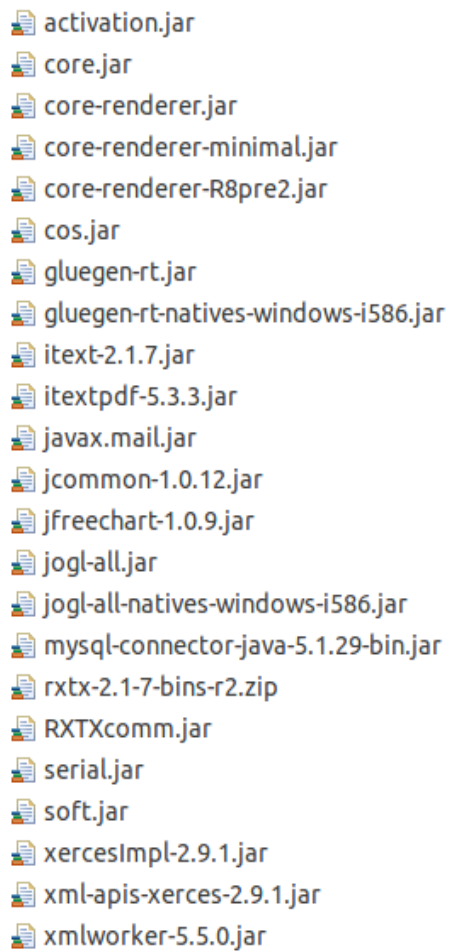


Figure 6. Tomcat folder contains

2.17 List of External Libraries



activation.jar
core.jar
core-renderer.jar
core-renderer-minimal.jar
core-renderer-R8pre2.jar
cos.jar
gluegen-rt.jar
gluegen-rt-natives-windows-i586.jar
itext-2.1.7.jar
itextpdf-5.3.3.jar
javax.mail.jar
jcommon-1.0.12.jar
jfreechart-1.0.9.jar
jogl-all.jar
jogl-all-natives-windows-i586.jar
mysql-connector-java-5.1.29-bin.jar
rxtx-2.1-7-bins-r2.zip
RXTXcomm.jar
serial.jar
soft.jar
xercesImpl-2.9.1.jar
xml-apis-xerces-2.9.1.jar
xmlworker-5.5.0.jar

Figure 7. External libraries used for development

2.18 Tools and Platforms

Tools and platform used were:

+Operating Systems

- Ubuntu 12.04 with Eclipse for Development
- Windows 7 for Documentation

+Browsers

Application engine is Java based so its platform independent and support all browsers.

+Servers

- MySQL with port 3306
- Apache with port 80
- Tomcat with port 8089

3 SOFTWARE DESIGN

3.1 Agile Software Development

Agile is a set of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

The most significant of the Agile software development is the Agile Manifesto which explains twelve principles. During the development of the thesis I learnt Agile to be best as it is flexible in developing and welcome manipulations on the run which proved its effectiveness. Scrum is the main part of it and it is very famous for its simplicity and flexibility.

Scrum has three roles, Product owner, Scrum Master and Team. In the thesis all the roles have been performed individually.

Agile is very famous to track the project and it is kind of watchdog throughout the project. The most favorable part of Agile is that, during the requirement gathering the team can develop the project as well. This is the delicacy of the Agile software development. Reference to /3/.

3.2 Backlog

Backlog is the list of all the requirements asked by the customer.

Story ID	Priority	Description	Story Points	Cumulative story points
1	3	Maintain the user's transactions	5	5
2	2	Maintain the user's profile	3	8
3	4	define user search criteria	5	13
4	1	Admin could modify, enter and delete the users.	5	18
5	5	new bank promotions invok to home page	3	21
				21
		Memos, trainings, meetings and official tender notices records		
6	8	maintenance and access on web by users	3	24
7	7	Bond draws results and record draws in database	3	27
8	6	Loan offers and loan calculator	2	29
				29
		Available bank certificates mean securities and new promotion		
9	9	relatively	3	32
10	10	e-banking facility to users	5	37
11	11	staff could download common shared files	5	42
12	14	users can buy certificates	5	47
13	12	Complain box to the customers	5	52
14	13	staff share files to common folder	5	57
				57
		Extras		57
15	15	Feedback facility for particular department	2	59
16	16	show the transection average in graphs	3	62
17	17	google search api for google search	2	64
18	18	google news api for news updates	2	66
19	20	facebook api for advertisement	2	68
20	19	Graphs for the admin and management statistic overview	3	71
21	22	database replication for database mirroring and risk management	3	74

Figure 8. Backlog for product

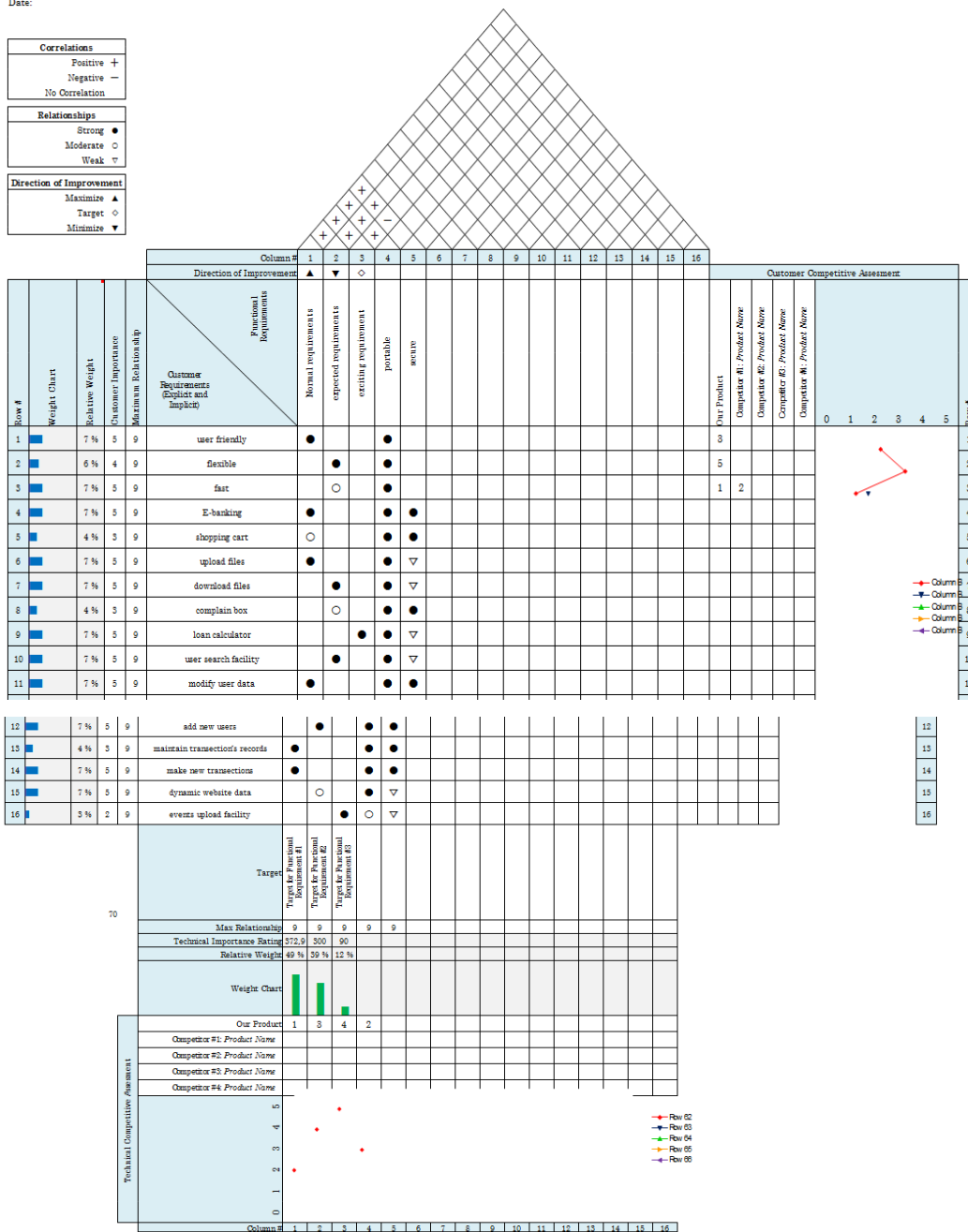
3.3 QFD

QFD is a method to convert user requirements to quantitative parameters; it transforms the needs into engineering traits. QFD is applied into variety of services. QFD is called as road map of design. Reference to /13/.

In short we have customer requirements will be built into a QFD matrix.

QFD: House of Quality
 Project: ATM and Web Bank
 Revision: 01.01.00
 Date:

Correlations	
Positive	+
Negative	-
No Correlation	
Relationships	
Strong	●
Moderate	○
Weak	▽
Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼



















Row #	Weight Chart	Relative Weight	Customer Importance	Maximum Relationship	Customer Requirements (Explicit and Implicit)	Functional Requirements	Normal requirements	expected requirements	exciting requirement	portable	secure
1		7 %	5	9	user friendly		●			●	
2		6 %	4	9	flexible			●		●	
3		7 %	5	9	fast			○		●	
4		7 %	5	9	E-banking		●			●	●
5		4 %	3	9	shopping cart		○			●	●
6		7 %	5	9	upload files		●			●	▽
7		7 %	5	9	download files			●		●	▽
8		4 %	3	9	complain box			○		●	●
9		7 %	5	9	loan calculator				●	●	▽
10		7 %	5	9	user search facility			●		●	▽
11		7 %	5	9	modify user data		●			●	●
12		7 %	5	9	add new users			●		●	●
13		4 %	3	9	maintain transection's records		●			●	●
14		7 %	5	9	make new transections		●			●	●
15		7 %	5	9	dynamic website data			○		●	▽
16		3 %	2	9	events upload facility				●	○	▽

Figure 9. QFD relative weight Quality house

The Quality matrix template idea is taken from the Internet and it shows relative weight of the customer requirements and developer functions. In the rows, there are sixteen customer's requirements are mentioned where as in the columns, there are five developer's functions mentioned. Three kinds of symbols are used to show the relation of customer's requirements and developers functions.

3.4 System Architecture

The main architecture of the web server and ATM is explained with the help following two figures.

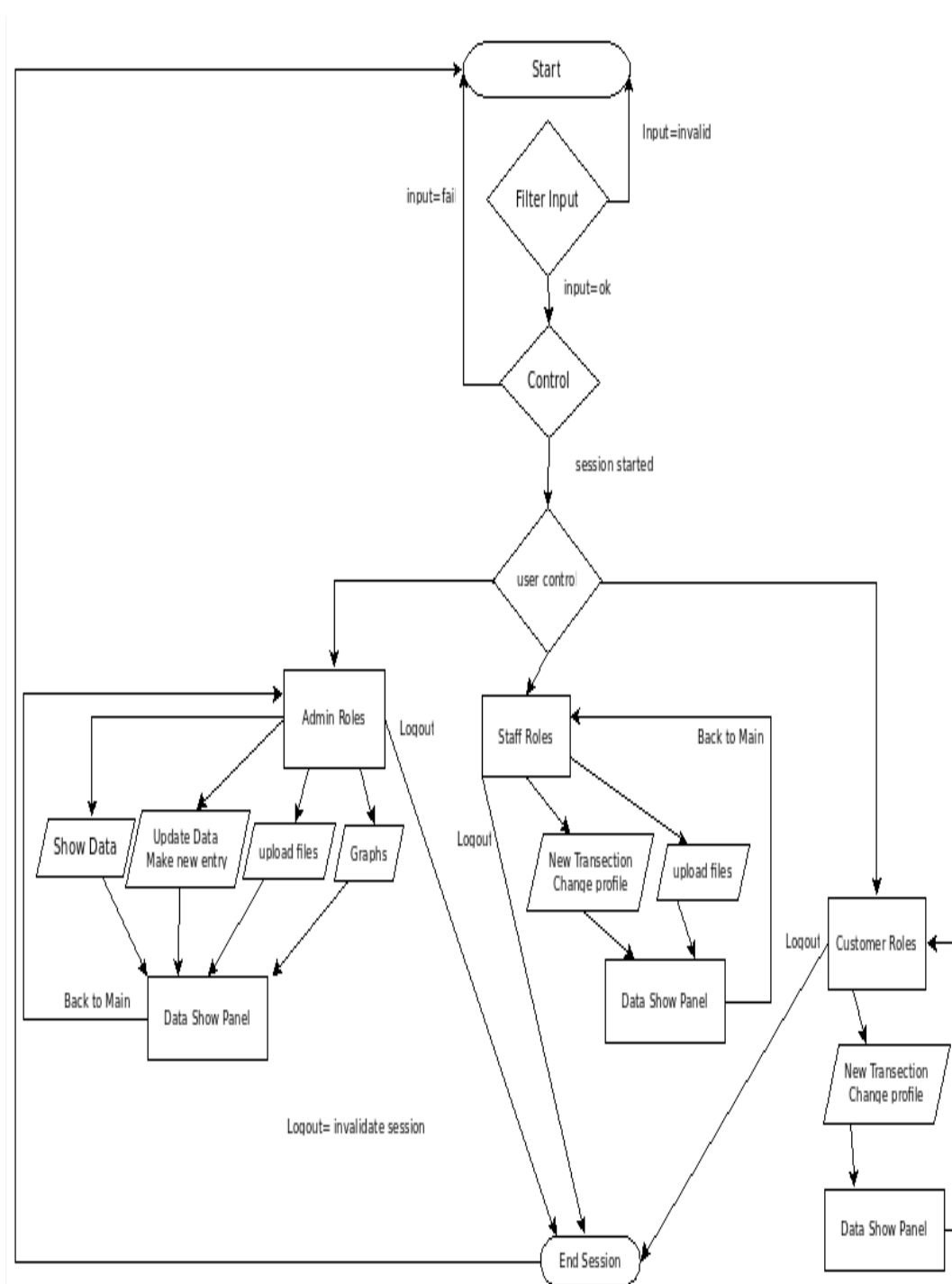


Figure 10. Web Bank Architecture

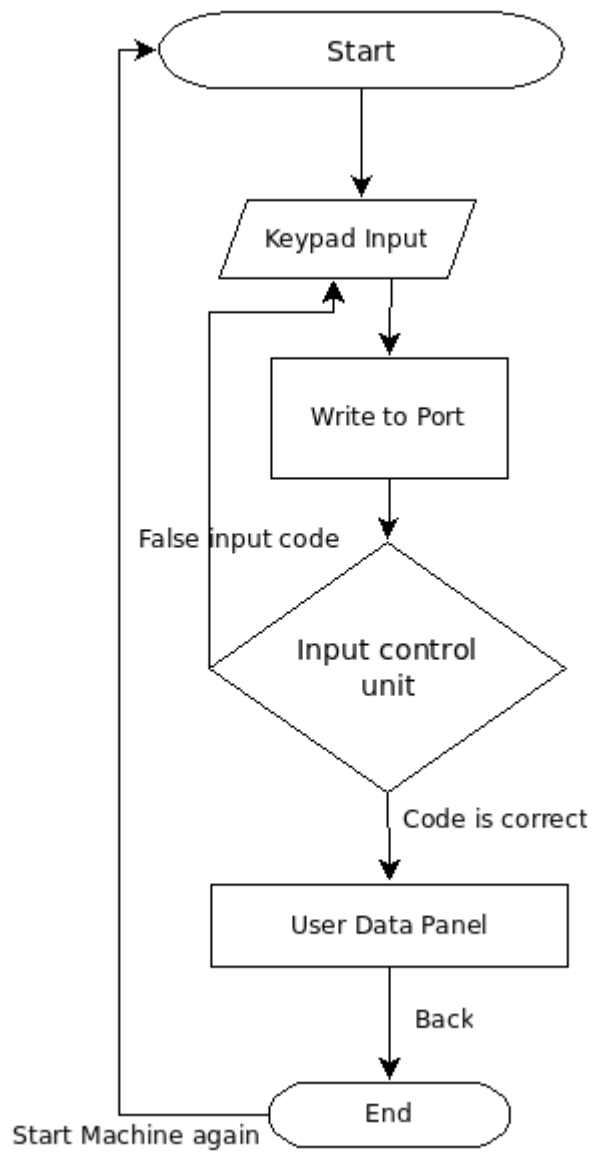


Figure 11. ATM Architecture

3.5 Use Cases

Figure 12 shows the use case for the development of web server

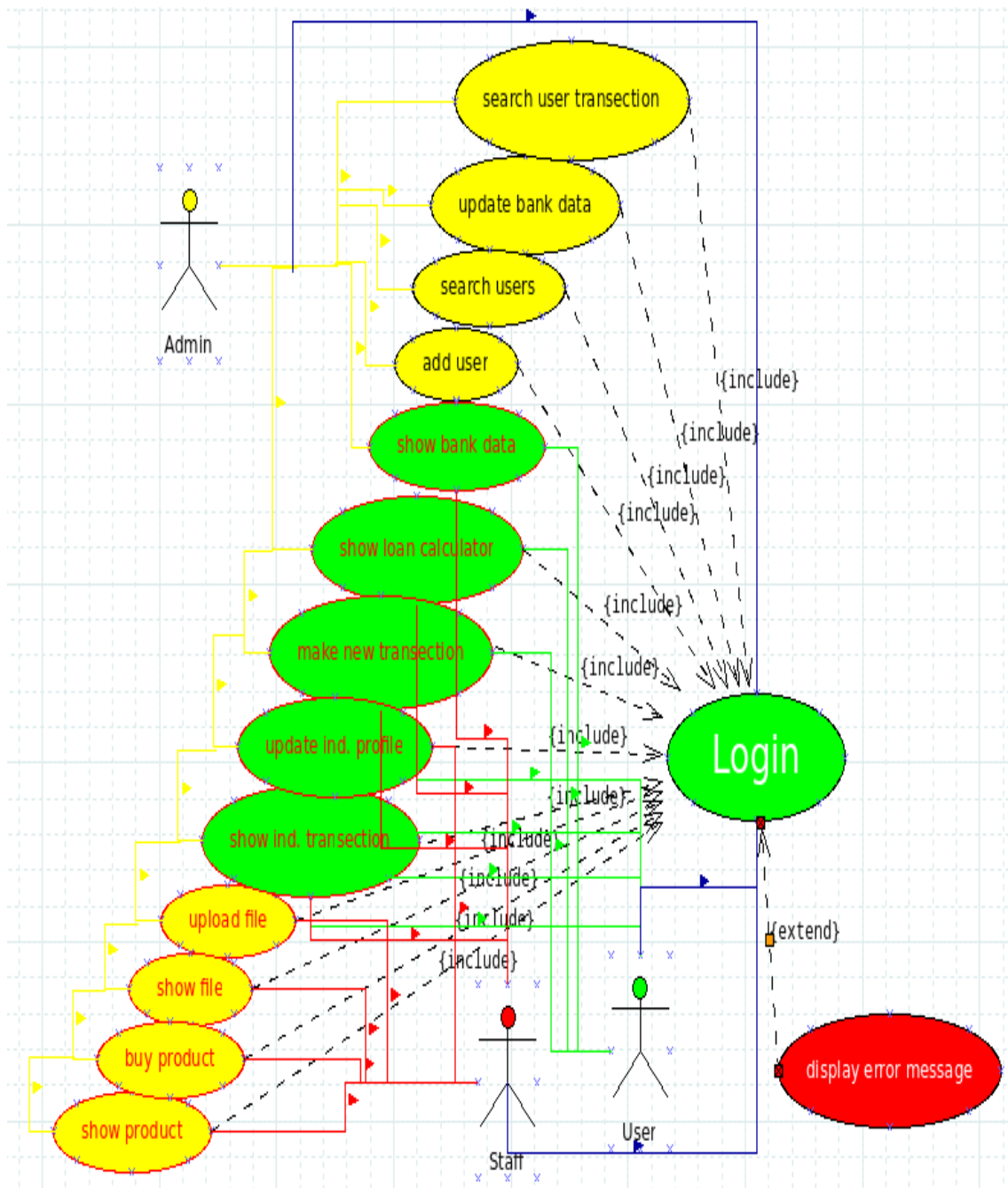


Figure 12. Development of web server

The use case for the development of ATM can be seen in the next figure.

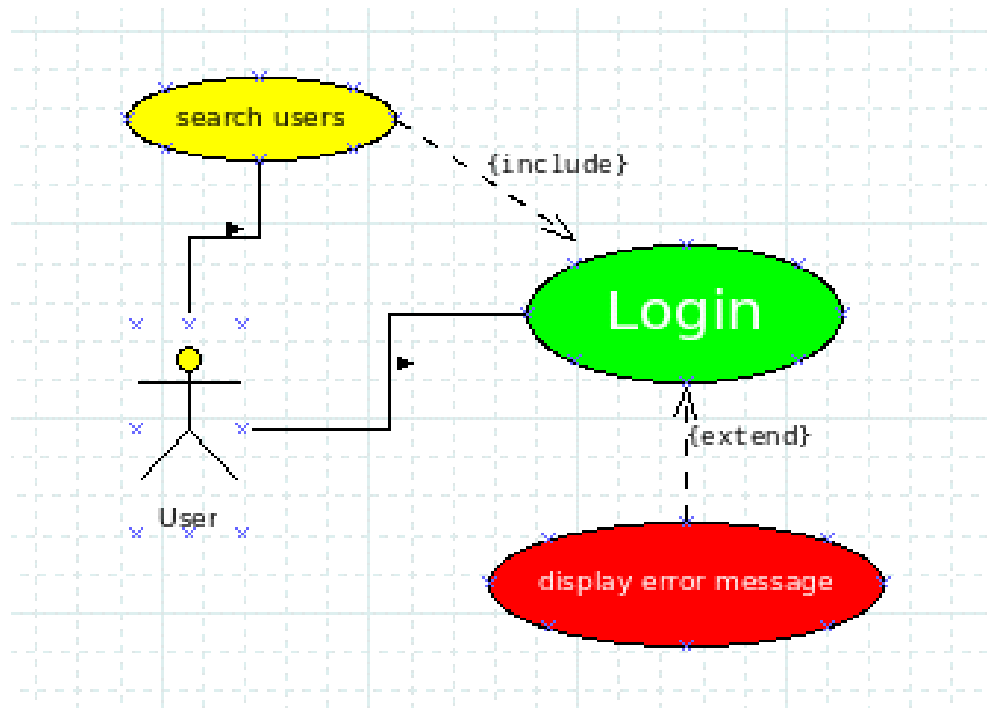


Figure 13. Development of ATM

3.6 ERD

There are many tables used in the project and diagram shows the contents.

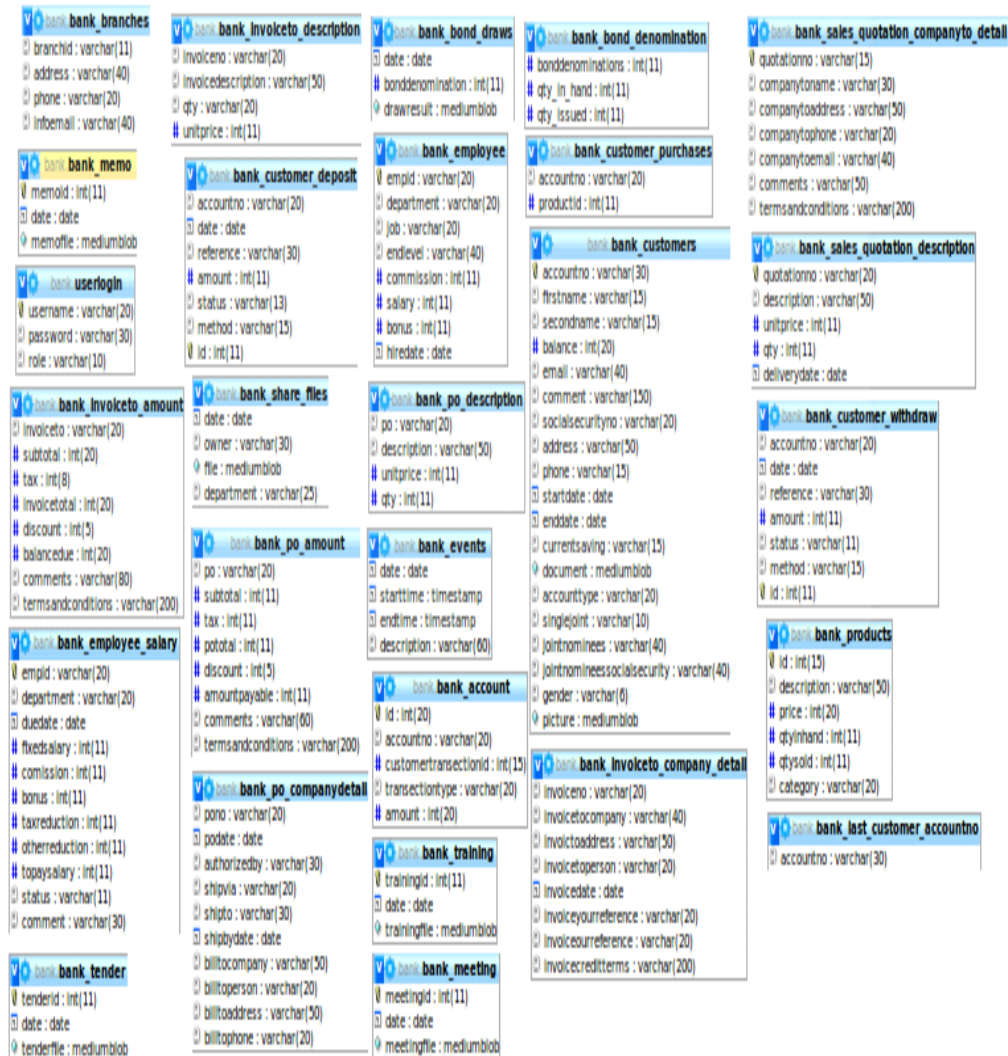


Figure 14. ERD for database

4 IMPLEMENTATION

4.1 Login User Form and Session

There are three kinds of users in the pages: the users with the full control with role administrators, staff member with limited control and users with customer control. The administrators can make new users, change profile of all users with limited editable values. By using graphs and transactions load can be monitored. Moreover, the session persists among specified pages. The validation criteria are defined to filter the input from the user on the run.

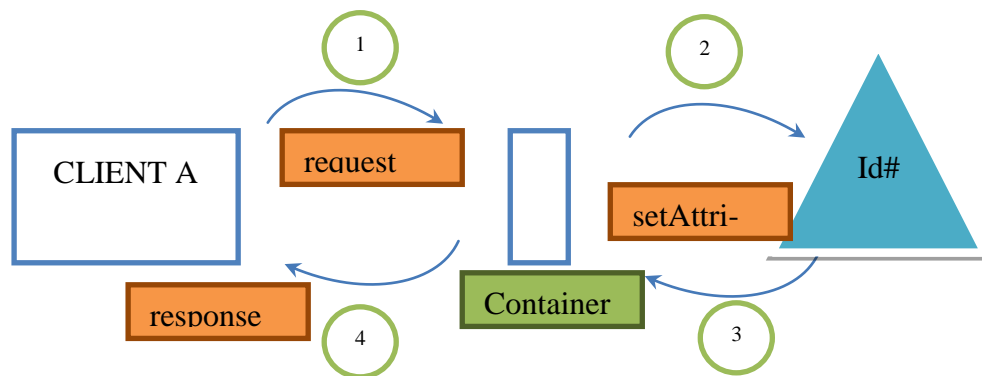


Figure 15. Client session request and response

4.2 New User Form

The functionality of the specified page is to generate a new user and the main part is to distinguish three kinds of users: customers, staff members and administrators. The password is generated on the run using some technical algorithm. The validation criteria are defined in a background to filter the input from the users. The page pops up with the session and the session has a username unique so in case there are multi-administrators, they can insert data at the same time with a different user's login.

4.3 Search User Form

The user search form has a filter criteria, the criteria is predefined up to limited search. It is divided into all users, outside customers, staff and can be filtered for

individual user. The multi user facility makes the form very dynamic. The administrator can change the editable contents of specific users and can monitor all the transactions of made by particular user. The statistic part adds more dynamics into the administrator and that is a graph. The administrator can check the trend of the user in the graphical mode and can find the average of transactions made by the user in everyday process.

4.4 Upload Files to Database

To upload the pdf files from anywhere is the main trait of this implementation. In general the users can share files a common sharing place. This part of the system makes things more user-friendly.

4.5 Download Files from Database

To download the uploaded files we have options where we can download the pdf files, the concept is very vivid and simple. There is a common place where there are many files uploaded for common sharing with the staff members. Downloading from the specified place is easy.

4.6 Shopping Cart

All registered users of the system can easily buy specified products available in the cart and can delete from the cart as well. The idea behind is online shopping. The concept is online products offered by the company can be bought and sold to registered users. The each transaction affects the user balance and the transaction is recorded for the particular usage, affected transaction debits the bank account as well.

4.7 Concept of Transactions and Graph

Each transaction has a unique identity and reference number which distinguishes each transaction from others. There are some records which will be affected after each transaction, such as deposit, withdrawal, balance and bank account. These all

set of records are affected on each transaction. For example, user A makes payment and transfer money from an account to user B's account. In this transaction, money from user A's account will be withdrawn and deposited to the user B's account. According to this transaction balances of both accounts will take effect and a particular amount of money will be charged by the bank on each transaction. The second type of transaction is online shopping, in this case the money will be credited from the user's account, balance will take effect and debited to the bank account.

4.8 Concept of Bank Account

According to the concept discussed in earlier, on each transaction bank will charge some amount from the customer and transaction deposits a small amount to the bank account according to a specified percentage. Moreover, on every purchase, the user deposits the particular amount to the bank account with specific identity that distinguishes the transaction from other kind of transactions.

4.9 Loan calculator and technologies

The idea about the loan calculator is very similar to the real life; the concept was taken to generate the XML file on the runtime. Mainly the concept was about the conversion from XML to pdf. There has been runtime generation of data which makes the XML file and by using a specified XSL file, it represent the data in HTML and then pdf afterwards.

4.10 Update the Text on the Web

To make the web pages more dynamic the concept of text files was used. In this, the administrator can manipulate the text files and save data back which can be accessed by the users.

4.11 Feedback Form and Visitor Counter

Companies need feedback on the runtime to make it easier to make decisions for future plans. Therefore a feedback form was introduced. The second part, the visitors counter, is a very simple concept to find out how many visitors access the internet every day. By using this data, the administrator can plan for the server life.

4.12 Arduino Uno Board

The Arduino Uno is based on ATmega328. It has 14 digital I/O pins and 16MHz ceramic resonator, USB connection, a power jack, an ICSP header and a reset button. The operating voltage is 5V. There are six analog input pins.

4.13 Arduino Communication

In the project UART TTL, 5V serial communication was used which is available on digital pins 0 (RX) and 1 (TX) used.

4.14 USB Overcurrent Protection

The Arduino Uno has a resettable poly-fuse that protects the USB of the computer from short-circuits and overcurrent. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the overload or short-circuit is removed.

4.15 4 x 4 Keypads

A sixteen buttons keypad is used to insert the pin code at the user login and all pins are attached to the corresponding pins on the board.

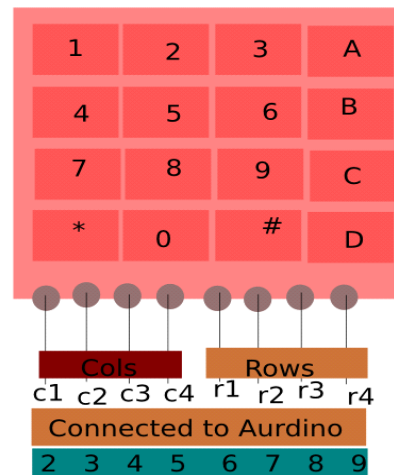


Figure 16. Keypad with Arduino pin configuration

4.16 Arduino I/O and JSP output

Arduino input and output is attached to the electronic components as discussed, the input is attached to the corresponding pins on the Arduino Uno and output of Arduino Uno is attached to the LED which is to indicate whether the input is correct or not. The LED will turn on in case the input is ok and turns off if the input is wrong. The Java Server Pages are used to communicate with the text file and timer is used to refresh the contents of the JSP, the specified file is attached with the serial port ttyACMX via shell scripting. The shell program reads the serial port and writes data to the text file. JSP retrieves the data from the text file and check the database for the particular record.

4.17 Facebook API

The Facebook Platform is a software environment provided by the social networking service Facebook for third-party developers. The platform offers a set of programming interfaces. Reference to 11/.

4.18 Google API

Google APIs is a set of JavaScript APIs developed by Google that allows interaction with Google Services and interaction of rich, multimedia, search or feed-based internet content into web applications. Reference to /12/.

5 TEST CASE

System meets specifications. There could be many test cases for the one requirement. Test case validates one or more system requirements and generates pass or fail. One test case could be for multiple requirements. The terminology domain used in this, it is set of values that the input may take and the range is the set of values that output may show.

This test case explains the login of user with the high priority. The test case was to check the functionality of the login procedure.

ID	1	
Title	Log in	
Priority	High	
Execution result	working	
Purpose	Verify input with Valid data	
	Description	Expected Result
Step 1	Open browser	The browser open
Step 2	Goto login	Log in page open
Step 3	Enter username and password (1111 &admin1)	
Step 4	Click on login	System logs in as admin

Test case 1. User Login

This test case explains the user search with the high priority. The test case was to check the functionality of the user search engine.

ID	2
Title	User search
Priority	High

Execution result	working	
Purpose	Find the specific user	
	Description	Expected Result
Step 1	Login as admin	Log in Works
Step 2	Click on administration	Admin menu open
Step 3	Open update user	Search page opens
Step 4	Enter the search criteria ssid 151080-2437	Open particular data

Test case 2. User Search

This test case explains the new user form with the high priority. The test case was to check the functionality of enter new user process.

ID	3	
Title	Add new user	
Priority	High	
Execution result	Working	
Purpose	New entry	
	Description	Expected Result
Step 1	Login as admin	Log in Works
Step 2	Click on administration	Admin menu opens
Step 3	Click add new user	New user form opens
Step 4	Enter the particulars	
Step 5	Click save	Data is saved

Test case 3. Add new user

This test case explains the user modification data with the high priority. The test case was to check the functionality of the procedure.

ID	4	
Title	Modify user data	
Priority	High	
Execution result	Working	
Purpose	Change the other user's profile	
	Description	Expected Result
Step 1	Login as admin	Log in Works
Step 2	Click on administration	Admin menu opens
Step 3	Click update user	User search form opens
Step 4	Enter particular social id 151080-2437	Data opens
Step 5	Click on edit	Profile opens
Step 6	Make changes like workplussuccess1@yahoo.com	
Step 7	Click save	Data is saved

Test case 4. Modify user data

This test case explains the shopping cart with the high priority. The test case was to check the functionality of the cart.

ID	5	
Title	Buy products	
Priority	High	
Execution result	Working	
Purpose	Add to cart and confirm	
	Description	Expected Result

Step 1	Login as a user	Log in Works
Step 2	Click on buy product	Products page open
Step 3	Select product	Product selected
Step 4	Add to cart	Product added to cart
Step 5	Click done	Product is bought

Test case 5. Buy products

This test case explains the upload procedure with high priority.

ID	6	
Title	Upload files	
Priority	High	
Execution result	working	
Purpose	Upload to common share	
	Description	Expected Result
Step 1	Login as a user	Log in Works
Step 2	Open staff menu	Staff menu opens
Step 3	Click upload files	Upload file page opens
Step 4	Click browse	
Step 5	Click add	File is uploaded

Test case 6. Upload files

This test case explains the files download procedure with high priority.

ID	7
Title	Download files
Priority	High
Execution result	working

Purpose	Download from common	
	Description	Expected Result
Step 1	Login as a user	Log in Works
Step 2	Open staff menu	Staff menu opens
Step 3	Click download files	download page opens
Step 4	Click particular file	Invok save menu
Step 5	Click save	File is downloaded

Test case 7. Download files

This test case explains the loan calculator functionality with medium priority.

ID	8	
Title	Calculate loan	
Priority	medium	
Execution result	working	
Purpose	Calculate loan installment	
	Description	Expected Result
Step 1	Click on accounts	Account page opens
Step 2	Click house loan	Loan page opens
Step 3	Enter values(20000 5% 12%)	
Step 4	Click caalculate	Installment page appears

Test case 8. Calculate Loan

This test case explains the new payment with high priority.

ID	9
Title	Make payment
Priority	High

Execution result	working	
Purpose	Transfer money to other A/C	
	Description	Expected Result
Step 1	Login as a user	Log in Works
Step 2	Click new payment	Payment page opens
Step 3	Enter values (to A/C: 1112 ref: 2324 amt:200)	
Step 4	Click done	Payment made

Test case 9. Make payment

This test case explains update text procedure with high priority.

ID	10	
Title	Update website data	
Priority	High	
Execution result	working	
Purpose	Update text on website	
	Description	Expected Result
Step 1	Login as a admin	Log in Works
Step 2	Open administratio	Admin menu opens
Step 3	Click update text	Page appears
Step 4	Click read file (car insurance)	Shows file contents
Step 5	Make changes in file text	
Step 6	Save	Specific file saved

Test case 10. Update website data

This test case explains the users graph working properly with high priority.

ID	11	
Title	Show user graph	
Priority	High	
Execution result	working	
Purpose	Show user transaction statistics	
	Description	Expected Result
Step 1	Login as a admin	Log in Works
Step 2	Open administration	Admin menu opens
Step 3	Click update user	Search page opens
Step 4	Enter ssid 151080-2437	Data opens
Step 5	Click graph	Graph page opens
Step 6	Graph shows transaction data and average	Works

Test case 11. Show user graph

This test case explains the ATM working properly with high priority.

ID	12	
Title	ATM	
Priority	High	
Execution result	working	
Purpose	Opens user data with keypad	
	Description	Expected Result
Step 1	Open terminal run shell program	Shell program start
Step 2	Open browser website	opens
Step 3	Click ATM	Page opens
Step 4	Start ATM	
Step 5	Enter keys from keypad (456)	Wrong Key input
Step 6	Enter keys (4567)	Login success

Test case 12. ATM

6 CONCLUSIONS

This thesis introduces the concept of conjugation of multiple departments into one context and integration of multiple servers in confined place.

The approach to reach the concept was quite exponential. The reason for the exponential approach throughout the thesis includes predefined scheduling and multi technology usage. There are many third party applications used in the thesis with the Microsoft and Linux both platforms. Eclipse was the main developing tool used for the preparation of project and it proved to be very powerful tool. Suitable mathematical algorithms were used for random generation of limited number of password alphanumeric; a mathematical algorithm was applied while calculating the loan values on the run and the exposure of graphs with average values. Mathematical algorithms were used broadly in some contexts, such as in the evaluation of feedback input as well.

The project was quite lengthy and it was very challenging to sort out the untidy material. Therefore, the main part was to make the material into an understandable form. The multiplatform structure makes the thesis work more flexible for the end user.

Last but not least, in the implementation phase the flexible structure of the whole design has been shown. The main role of the robust structure of the Java server engine and PHP is to introduce the significance of the technology in the new era of information technology.

The thesis required a lot of knowledge and some technical experiences. This project idea of the conjugation of departments in confined place is flexible and worth full in its own context. I wish, VAMK uses the idea for further development and conjugation of departments.

REFERENCES

/1/ UART. Accessed 03.5.2014
http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

/2/ Universal serial bus. Accessed 03.5.2014
<http://en.wikipedia.org/wiki/USB> 3rd may 2014

/3/ Agile development. Accessed 04.5.2014
http://en.wikipedia.org/wiki/Agile_software_development

/4/ JQuery. Accessed 25.4.2014
<http://en.wikipedia.org/wiki/JQuery>

/5/ PHP. Accessed 25.4.2014
<http://en.wikipedia.org/wiki/PHP>

/6/ HTTP. Accessed 30.4.2014
http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

/7/ XML. Accessed 30.4.2014
<http://en.wikipedia.org/wiki/XML>

/8/ Arduino. Accessed 30.4.2014
<http://simple.wikipedia.org/wiki/Arduino>

/9/ Arduino. Accessed 30.4.2014
<http://en.wikipedia.org/wiki/Arduino>

/10/ Arduino Board. Accessed 03.5.2014
<http://arduino.cc/en/Main/arduinoBoardUno>

/11/ Facebook Platform. Accessed 03.5.2014
http://en.wikipedia.org/wiki/Facebook_Platform

/12/ Google API. Accessed 03.5.2014
http://en.wikipedia.org/wiki/Google_APIs

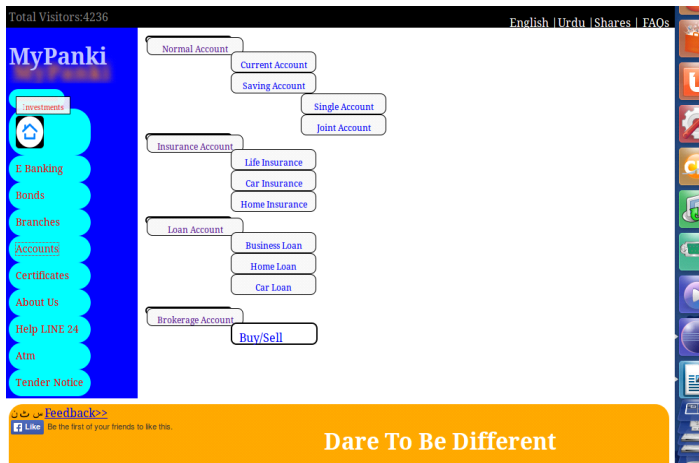
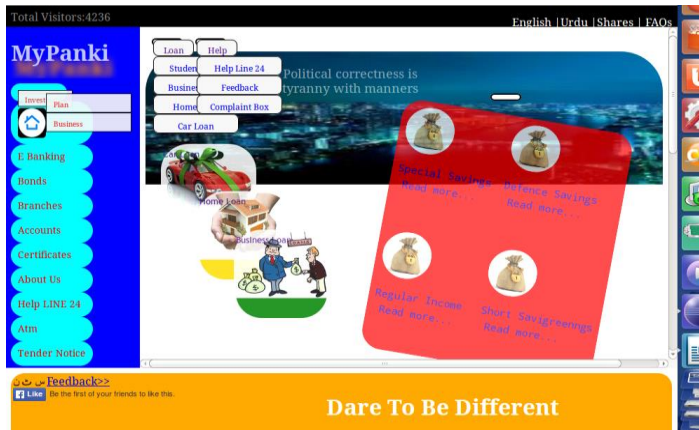
/13/ QFD. Accessed 05.5.2014
http://en.wikipedia.org/wiki/Quality_function_deployment

/14/ Apache configuration. Accessed 06.5.2014
<https://www.digitalocean.com/community/articles/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps>

APPENDICES

Appendix 1

Below are the screen shots of all the pages developed during the project work.



Total Visitors:4236

MyPanki

Investments

- Home
- E Banking
- Bonds
- Branches
- Accounts
- Certificates
- About Us
- Help LINE 24
- Atm
- Tender Notice

Bond Draws

- Current Account
- Current Account
- Saving Account
- Single Account
- Joint Account
- Joint Account
- Joint Account
- Joint Account
- Joint Account
- Joint Account
- Joint Account

Total Visitors:4236 English Urdu IShares

MyPanki

Investments

- Home
- E Banking
- Bonds
- Branches
- Accounts
- Certificates
- About Us
- Help LINE 24
- Atm
- Tender Notice

Login Here

User Name:

Password:

Feedback >>> Be the first of your friends to like this.


Dare To Be Different

MyPanki

Investments

- Home
- E Banking
- Bonds
- Branches
- Accounts
- Certificates
- About Us
- Help LINE 24
- Atm
- Tender Notice

Special Savings Defiance Savings
Read more... Read more...



Regular Income Short Savings
Read more... Read more...

Total Visitors:4236 English | Urdu | Shares | FAOs

MyPanki

Investments

Home

E Banking

Bonds

Branches

Accounts

Certificates

About Us

Help LINE 24

Atm

Tender Notice

About Us...

With a presence in 78 countries and nearly 190,000 employees, including more than 145,000 in Europe,

BNP Paribas is a leading bank in the euro zone and a first-class player on a global scale.

Total Visitors:4236 English | Urdu | Shares | FAOs

MyPanki

Investments

Home

E Banking

Bonds

Branches

Accounts

Certificates

Plan...

In business, a strategic business unit (SBU) is a profit center which focuses on product offering and market segment. SBUs typically have a discrete marketing plan, analysis of competition, and marketing campaign, even though they may be part of a larger business entity.

Total Visitors:4236 English | Urdu | Shares | FAOs

MyPanki

Investments

Home

E Banking

Bonds

Branches

Accounts

Certificates

About Us

Help LINE 24

Atm

Business...

Are you one of those who dream of owning your own business? You'll be your own boss, and the captain of your destiny??maybe even a captain of industry. Is it hard? Without a doubt. Is it challenging? Absolutely. Do you have to be wealthy and well-educated with a lengthy resumé? Not at all! Can you do it? As the magic 8-ball says, "All signs point to yes!" So how do you do it, and make it work? Plan, plan, plan! There are some tried and true ways head down the path of creating your own business, and there is no time like the present to get started!

Total Visitors:4236 English | Urdu

MyPanki

We expect your page rating:

Select Dept.:

Select Rating:

Invest Plan
 Business
 Banking
 Bonds
 Branches
 Accounts
 Certificates
 About Us
 Help LINE 24
 ATM
 Tender Notice

Feedbacks
 Like Be the first of your friends to like this.

Dare To Be Different

018:4236 English | Urdu | Share

anki

Welcome 1111
[Contents...](#) [Logout Here..](#)

- New Payment
- Transactions
- Change Profile
- Administration
- Office Work
- Buy Products
- Sell Products
- Balance...2582

236 English | Urdu | Shares | FAQs

Welcome 1111

Main Customer Page

Make a New Payment:

Your Account	Benef. Account	Reference	Amount
1111	1111	00121	only int

Welcome 1111 English | Urdu

Main Customer Page

Account	Date	Reference	Amount	Status	Method	Id
1111	2014-04-15	21321	212	done	cheque	1
1111	2014-04-15	87	776	done	cheque	4
1111	2014-04-16	5245	50	done	cheque	12
1111	2014-04-16	4645	1002	done	cheque	13
1111	2014-04-16	545	500	done	cheque	14
1111	2014-04-16	45645	8000	done	cheque	15
1111	2014-04-16	1111	11	done	banktransfer	17
1111	2014-04-16	1111	11	done	banktransfer	18

New Entries
New Customer or Staff
New Meeting
New Tender Notice
New Training
New Memo
New Draw Result
Manage Bonds
Manage Branch Info
Update Entries
Update Customer or Staff
Update Meeting
Update Tender Notice
Update Training
Update Memo
Update Text
Show Graphs
Main Page

English | Urdu | Shares | FAQs

Enter Information Here

First Name	<input type="text" value="Tanveer"/>	Last Name	<input type="text" value="Altaf"/>
Email	<input type="text" value="himan1@yahoo.fi"/>	Social Sid	<input type="text" value="151080-2437"/>
User Name	<input type="text" value="1111"/>	Password	<input type="text" value="admin1"/>
Address	<input type="text" value="Helsinki"/>	Phone	<input type="text" value="04036810"/>
<input type="button" value="Save"/>			
Admin Main			
<input type="text" value="Purchase"/>	<input type="text" value="Manager"/>	<input type="text" value="Bachelor"/>	
<input type="text" value="100"/>	<input type="text" value="2000"/>	<input type="text" value="200"/>	

English Urdu States 7408

Enter Information Here

First Name	<input type="text"/>	Last Name	<input type="text"/>
Email	<input type="text"/>	Social Sid	<input type="text"/>
User Name	<input type="text"/>	Password	<input type="password" value="****"/>
Address	<input type="text"/>	Phone	<input type="text"/>
<input type="text"/>	<input type="text" value="Normal A/C"/>	<input type="text" value="Single"/>	<input type="text" value="Male"/>
Jointnominees(separate with (,))		Jointnominees SSID	
Comment	<input type="text"/>	Account No.FI2244- Done	<input type="text" value="4574"/>
Start Date: Today	End Date	<input type="text" value="0000-00-00"/>	
Attach File	<input type="button" value="Browse..."/> No file selected.	Balance	<input type="text" value="0"/>
<input type="button" value="Save"/>	<input type="button" value="Reset"/>		

Admin Main Staff Entry

You are Login as PurchaseManager

- [Contents...](#)
- [Mail Box](#)
- [Check Memo](#)
- [Training Info](#)
- [Meeting Info](#)
- [Events](#)
- [Upload To Share Folder](#)
- [Show Share Folder](#)
- [Deposit Customer](#)
- [Withdraw Customer](#)
- [Make Purchaseorder](#)
- [Raise Invoice](#)
- [New Quotation](#)
- [Main Page](#)

English | Urdu | Shares | FAQs

Main Staff Page

Select The File

aboutus

Are you one of those who dream of owning your own business? You'll be your own boss, and the captain of your destiny—maybe even a captain of industry. Is it hard? Without a doubt. Is it challenging? Absolutely. Do you have to be wealthy and well-educated with a lengthy resume? Not at all! Can you do it? As the magic 8-ball says, "All signs point to yes!" So how do you do it, and make it work? Plan, plan, plan! There are some tried and true ways head down the path of creating your own business, and there is no time like the present to get started!

Read Save

Please enter item to add or remove:
Add Item:

1 certificate special saving Price:100

add remove

Main Customer Page

Your Cart...

Sr. No	Product Number
1	Product id# 1
2	Product id# 1

Main Customer Page

Bank Customers

Admins	20.0%
Staff	26.0%
Outside Customers	53.0%

Customers Amount

A/c Balance Variation

Admin Main Staff Entry Admin

Department Job End Level

Fixed Commission Basic Salary Bonus Agree Pic. Browse

User Management
Admin Main

Choose Criteria **Xml** Enter(totalcustomers for All)[outsidecustomers,staff] ['ssid','ssid'...]

Enter Information Here

First Name Last Name

Email Social Sid

User Name Password

Address Phone

Admin Main

Purchase Manager Bachelor

User Management
Admin Main

Choose Criteria **Xml** Enter(totalcustomers for All)[outsidecustomers,staff] ['ssid','ssid'...]

ame	Balance	Social Id	Email	Gender	Address	A/C Type			
tanveer altaf	2582	151080-2437	himan1@yahoo.fi	male	Helsinki	normal	Edit	Show Transections	A/c Variations Graph

User Management
Admin Main

Choose Criteria **Xml** Enter(totalcustomers for All)[outsidecustomers,staff] ['ssid','ssid'...]

Welcome 151080-2437

Account	Date	Reference	Amount	Status	Method	Id
1111	2014-04-15	21321	212	done	cheque	1
1111	2014-04-15	87	776	done	cheque	4
1111	2014-04-16	5245	50	done	cheque	12
1111	2014-04-16	4645	1002	done	cheque	13
1111	2014-04-16	545	500	done	cheque	14
1111	2014-04-16	45645	8000	done	cheque	15
1111	2014-04-16	1111	11	done	banktransfer	17
1111	2014-04-16	1111	11	done	banktransfer	18
1111	2014-04-16	1111	11	done	banktransfer	19

No file selected.

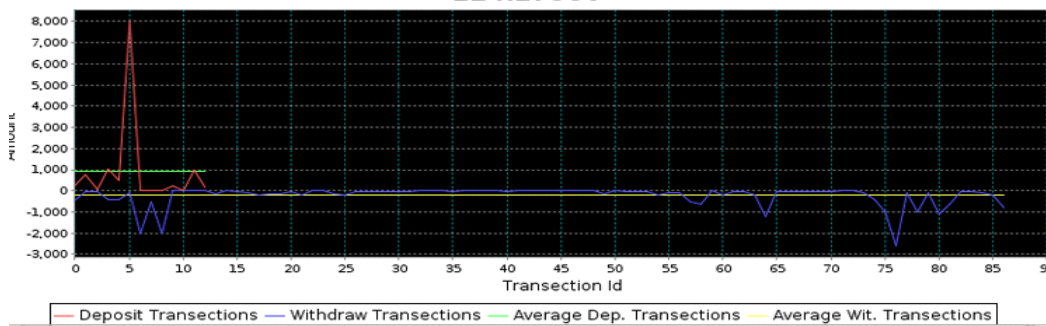
Owner

Dpt.

User Management
Admin Main

Choose Criteria Enter (totalcustomers for All)[outsidecustomers,staff] ['ssid','ssid'...]

Deposit/Withdraw Dcount:13 Wcount:87 Avg(D):921.7692 Avg(W):-224.27586



Common share files to download..

Date	Owner	Department	download
2014-04-23	tanveer	purchase	file here1
2014-04-25	Himan	sales	file here2
2014-12-12	Tanveer	sales	file here3
2014-04-24	Himan	administration	file here4

[<<< Back...](#)

Total Visitors:4361 Shares | FAQs

MyPanki

Investments

[Home](#)

[E Banking](#)

[Bonds](#)

[Branches](#)

[Accounts](#)

[Certificates](#)

[About Us](#)

[Help LINE 24](#)

[Atm](#)

[Tender Notice](#)

Years	Principal Amount	Interest	Abbreviation	Monthly Payment
1	10000	100.0	833	1038.0
2	9167	91.666664	833	929.6667
3	8334	83.333336	833	921.3333
4	7501	75.0	833	913.0
5	6668	66.666664	833	904.6667
6	5835	58.333332	833	896.3333
7	5002	50.0	833	888.0
8	4169	41.666668	833	879.6667
9	3336	33.333332	833	871.3333
10	2503	25.0	833	863.0
11	1670	16.666666	833	854.6667
12	837	8.333333	833	846.3333

Enter Loan Information:

1) Loan Amount

2) Annual %:

3) Perios(years):

Base Fee:

Monthly Fee:

Payment Information:

Feedback >>

Like Be the first of your friends to like this.

Dare To Be Different

Total Visitors:4361 Shares | FAQs

MyPanki

Investments

[Home](#)

[E Banking](#)

[Bonds](#)

[Branches](#)

[Accounts](#)

[Certificates](#)

[About Us](#)

[Help LINE 24](#)

[Atm](#)

[Tender Notice](#)

Main ATM Page

You entered Wrong code
5454

Feedback >>

Like Be the first of your friends to like this.

Dare To Be Different

Appendix 2

Some Sample Code

```
/**
 *
 */
```

The Sample code is for the xml file generation on the run time. The code was used to produce the xml file for loan calculation.

```
/**
 *
 */
```

```

    try {

        int                loaana-
        amount=Integer.parseInt(request.getParameter("principal"));

        int                inter-
        est=Integer.parseInt(request.getParameter("interest"));

        int years=Integer.parseInt(request.getParameter("years"));

        int                month-
        lyfee=Integer.parseInt(request.getParameter("monthlyfee"));

        int
        basefee=Integer.parseInt(request.getParameter("basefee"));

        months=years*12;

        DocumentBuilderFactory docFactory = Document-
        BuilderFactory.newInstance();

        DocumentBuilder    docBuilder    =    docFacto-
        ry.newDocumentBuilder();

        // root elements
        Document doc = docBuilder.newDocument();

```



```

Element          rootElement          =
doc.createElement("company");
    doc.appendChild(rootElement);
    //////////////////////////////////////
    remainamount=loanamount;
    abbreviation=loanamount/months;
    inter-
estpay=(float)((remainamount*interest)/100)/12;
    month-
lypaid=interestpay+(float)(abbreviation+basefee+monthlyfee);
    //////////////////////////////////////
    for(int i=1;i<=(years*12);i++){

        // staff elements
        Element staff = doc.createElement("customer");
        rootElement.appendChild(staff);

        // set attribute to staff element
        Attr attr = doc.createAttribute("id");
        attr.setValue(i+"");
        staff.setAttributeNode(attr);

        // shorten way
        // staff.setAttribute("id", "1");

        // firstname elements
        Element loan = doc.createElement("loanamount");

        loan.appendChild(doc.createTextNode((remainamount)+""));
        staff.appendChild(loan);

        // lastname elements

```

```

        Element          interestrate          =
doc.createElement("annualinterestrate");
        in-
terestrate.appendChild(doc.createTextNode(interestpay+""));
        staff.appendChild(interestrate);

// nickname elements

        Element period = doc.createElement("years");
        period.appendChild(doc.createTextNode(i+""));
        staff.appendChild(period);

        Element          fixedpermonth          =
doc.createElement("abbreviation");
        fixedper-
month.appendChild(doc.createTextNode(abbreviation+""));
        staff.appendChild(fixedpermonth);

        Element          paypermonth          =
doc.createElement("monthlypaid");
        payper-
month.appendChild(doc.createTextNode(monthlypaid+""));
        staff.appendChild(paypermonth);

        remainamount=remainamount-abbreviation;
        inter-
estpay=(float)((remainamount*interest)/100)/12;
        month-
lypaid=interestpay+(float)(abbreviation+monthlyfee);
    }

```

```

////////////////////////////////////
/*
    // salary elements
    Element salary = doc.createElement("salary");
    salary.appendChild(doc.createTextNode("100000"));
    staff.appendChild(salary);
*/
    // write the content into xml file
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(doc);
    StreamResult result = new StreamResult(new File(request.getSession().getServletContext().getRealPath("/")+"bankdata/Loan.xml"));

    // Output to console for testing
    // StreamResult result = new StreamResult(System.out);

    transformer.transform(source, result);

    System.out.println("File saved!");

}/******

```

The Sample code is for the bean class. The code was used to make the bean class for the shopping cart.

```

/*****

```

```
public class CartBean{

    private String itemID;

    ResultSet rs=null;
    Statement stm=null;
    Connection conn=null;
    public int a=2;
private Vector v=new Vector();
public CartBean(){

    itemID="";
}
public int Icartcount(){
    int aa=1;
    return aa;
}

public void add(Object obj){
    v.add(obj);
}
public void remove(Object obj){
    v.removeElement(obj);

}
public Vector showCart(){
    return v;
}
public void makeConnection(){
    try{
        Class.forName("com.mysql.jdbc.Driver");
```

```

        conn                =(Connection)                DriverManager.
er.getConnection("jdbc:mysql://localhost/bank","root","raw");

        }catch(Exception ex){

        }

}

public String getTitle(int id){
    String title="";
    String category="";
    try{
        String query="select description,category from bank_products
where id="+id;
        stm=(Statement) conn.createStatement();
        rs=stm.executeQuery(query);
        while(rs.next()){
            title=rs.getString("description");
            category=rs.getString("category");
        }

    }catch(Exception e){
        e.printStackTrace();
        stm=null;
        rs=null;
    }
    return title + " "+category;
}

public int getPrice(String id2){
    int price=0;
    int id1=Integer.parseInt(id2);

```

```
try{
    String query="select price from bank_products where id="+id1;
    stm=(Statement) conn.createStatement();
    rs=stm.executeQuery(query);
    while(rs.next()){
        price=rs.getInt("price");
    }

}catch(Exception e){
    e.printStackTrace();
    stm=null;
    rs=null;
}
return price;
}

public int getQuantityinhand(int id){
    int qinhand=0;
    try{
        String query="select qtyinhand from bank_products where
id="+id;
        stm=(Statement) conn.createStatement();
        rs=stm.executeQuery(query);
        while(rs.next()){
            qinhand=rs.getInt("qtyinhand");
        }

    }catch(Exception e){
        e.printStackTrace();
        stm=null;
    }
}
```

```

        rs=null;
    }
    return qinhand;
}

public void insertProduct(String account,String productid){
    int pid=Integer.parseInt(productid);
    try{
        Statement st_insertproduct=(Statement) conn.createStatement();
        st_insertproduct.executeUpdate("INSERT INTO
bank_customer_purchases (accountno,productid)
VALUES("+account+", "+pid+"");
    }catch(Exception e){
        e.printStackTrace();
    }
}

public void withDraw(int amount,String account){
    int qinhand=0;
    int newid_key=0;
    int newid_key1=0;
    int customertransactionid=0;

    try{
        ResultSet rs_transaction_insert_bankaccount=null;
        ResultSet rs_transaction1=null;
        ResultSet rs_transaction=null;

        PreparedStatement ps_customers = (PreparedStatement)
conn.prepareStatement("INSERT INTO bank_customer_withdraw
VALUES(?,?,?,?,?,?)");
        PreparedStatement ps_customers1= (PreparedStatement)
conn.prepareStatement("INSERT INTO bank_account VALUES(?,?,?,?,?)");

```

```

Statement stm=(Statement) conn.createStatement();

////////////////////////////////////
String query4="select Max(id) from bank_account";

rs_transaction_insert_bankaccount=stm.executeQuery(query4);

while(rs_transaction_insert_bankaccount.next()){

newid_key1=rs_transaction_insert_bankaccount.getInt("Max(id)");

}

// String amount="100";
// int amountnew=Integer.parseInt(amount.toString());

////////////////////////////////////INSERT INTO WITHDRAW TRANSECTIN////////////////////////////////////

String query1="select Max(id) from
bank_customer_withdraw";
rs_transaction1=stm.executeQuery(query1);

while(rs_transaction1.next()){

newid_key=rs_transaction1.getInt("Max(id)");

```



```

    }

    customertransactionid=newid_key+1;

    ps_customers.setString(1,account);
    ps_customers.setDate(2,new                               ja-
va.sql.Date(System.currentTimeMillis()));
    ps_customers.setString(3,account);
    ps_customers.setInt(4,-(amount));
    ps_customers.setString(5,"done");
    ps_customers.setString(6,"purchase");
    ps_customers.setInt(7,newid_key+1);

    ps_customers.executeUpdate();
    stm.executeUpdate("INSERT INTO bank_account
(id,accountno,customertransactionid,transactiontype,amount) VAL-
UES('+(newid_key1+1)+'','"+account+"','"+customertransactionid+'','purchase','"+
amount+"");
////////////////////////////////////uPDATE CUSTOMER AC-
COUNT////////////////////////////////////
    String query="select accountno,balance from
bank_customers";

    rs_transaction=stm.executeQuery(query);

    while(rs_transaction.next()){

```

```

        if((rs_transaction.getString("accountno").equalsIgnoreCase(account)) &&
(rs_transaction.getInt("balance")>=amount)){

                //ps_customers.executeUpdate();
                ///////////////////////////////////////////////////Deleting balance from bank_customers
table////////

                String query3 = "UPDATE bank_customers " +
"SET balance = (balance-"+amount+") WHERE accountno
="+account;

                stm.executeUpdate(query3);

                }

        }

        ///////////////////////////////////////////////////INSERT INTO BANK AC-
COUNT////////////////////////////////////

        /*
ps_customers1.setInt(1,3);//
ps_customers1.setString(2,"1111");
ps_customers1.setInt(3,183);
ps_customers1.setString(4,"purchase");
ps_customers1.setInt(5,100);

ps_customers1.executeUpdate();

        */

        rs_transaction.close();
        rs_transaction1.close();
        rs_transaction_insert_bankaccount.close();

    }catch(Exception e){
        e.printStackTrace();
        stm=null;
    }

```

```
        //rs=null;
    }

}

public void cleanup() throws Exception{
    //stm.close();
    conn.close();
}

public String getItemID() {
    return itemID;
}

public void setItemID(String itemID) {
    this.itemID = itemID;
}

@Override
public String toString() {
    return itemID;
}

public void clearVec(){
    v.clear();
}

}
```

```

/*****

```

The following sample code was used to invoke the bean class. It was meant to give some energy to the shopping cart to make it runnable.

```

/*****

```

```

    if(action.equals("add")){
        cb.add(item);
        session.setAttribute("cart",cb);
        price1=cb.getPrice(getrealidforprice);

        productId[iii].setItemID(getrealidforprice.toString());

        iii++;

    }else if(action.equals("remove")){
        cb=(CartBean) session.getAttribute("cart");
        cb.remove(item);
        price1=-cb.getPrice(getrealidforprice);
        session.setAttribute("cart",cb);

    }else if(action.equals("done")){
        /*    if(hideamount==0){
            hideamount=12;
        }*/
        int aa=Integer.parseInt(request.getParameter("aa"));
        //price1=0;
        //cb=(CartBean) session.getAttribute("cart");
        //cb.remove(item);
        //session.setAttribute("cart",cb);
        //cb.makeConnection();
        done=1;

```

```
orderconfirm="Your order is Confirmed Thanks for Purchase...";
    cb.withDraw(aa,accountnum);
    for(CartBean d : productId)
    {
//    if(productId[ci]!=null){
        cb.insertProduct(accountnum, "1");
//    }
    }
//    JOptionPane.showMessageDialog(null,aaaa+"");
//    response.sendRedirect("mysessioninvalidate.jsp");

    }
    pricefordatabase+=price1;
    re-
    sponse.sendRedirect("Carts.jsp?id="+item+"&price="+price1+"&counter="+counter+"&done="+done+"&orderconfirm="+orderconfirm);
    }
```