Green Bank

Project Report On

Online Banking

Developed by:

Sanskriti Shukla

Tools &Technologies used

Java, JDBC JSP

Servlets Eclipse IDE

MySQL

Table Of Contents

|  |  |  |
| --- | --- | --- |
| Synopsis | | |
| Part I INTRODUCTION | | |
| 1. | Proposed System | 7 |
| 2. | Definition of Problem | 8 |
| 3. | Theoretical Background | 10 |
| 4. | Objective and Scope of the Project | 11 |
| Part II System Analysis | | |
| 5. | Requirement Analysis | 13 |
| 6. | Feasibility study | 14 |
|  | Part III System Requirements | |
| 7. | Hardware / Software specification | 1 17 |
|  | Part IV System Design | |
| 8. | Database design | 18 |
| 9. | E-R diagram | 20 |
| 10. | Dataflow Diagram | 21 |
| 11. | Use Case Diagram | 23 |
| 12. | Activity Diagrams | 24 |
| 13. | Deployment Environment Diagram | 26 |
| 14. | Process Description for customer | 27 |
|  | Part V Tools &Technologies used | |
| 15. | Java Server Pages | 28 |
| 16. | Servlets | 34 |
| 17. | MVC Architecture Model2 | 36 |
| 18. | Apache Tomcat | 37 |
| Part VI Software Development | | | |
| 19. | Input/Output Screens with Sample Data |  |
|  | Part VII Conclusion |  |
| 20. | Limitations |  |
| 21. | Future Enhancements |  |
| 22. | Bibliography |  |

# Synopsis - Mini Project

Green Bank - Online Banking System

# Introduction

Created by: Sanskriti Shukla

Class: - CSE(G) IVth Semester

RollNo.18100BTCSICS03038

The project is intended to provide online banking facilities such as checking accounts, paying mobile bills, getting pay-in slips and making transfers over World Wide Web.

Customer is supposed to provide login id and password to access his bank account online. With proper authorization the customer will be allowed to perform various banking activities. In addition to conventional banking services the customer is also facilitated with service like recharging mobile accounts for Airtel customers

There are two major ends in this project the customer end and the administrator end.

* Customer end is the end where customers make requests to view their account information update information and do transactions or pay certain bills.
* Administrator end is responsible for validating the user i.e. not allowing any unauthorized person to access the account. In event of a customer forgetting the Login password, provision has been made to provide password through a secure passage. Administrator is responsible for adding customers without affecting existing customers.

# Objective and Scope of the project

The main purpose of Online Banking Solution is to provide customers with an ability to make transactions online over a very user friendly interface.

Facilities provided

* View Account: Customer is authorized to check his account.
* Transfer Money: Customer can transfer money online over a secure passage to another account.
* Credit Card Facility: Customer can login using the credit card number. A customer can also apply for a credit card.
* Application Forms: Different forms such as new cheque book request, change of address etc can be downloaded.
* Customer Help: A document enabling customers to understand Online Banking.
* Bill Payment: Bills pertaining electricity, mobile can be paid online.
* Mobile Recharge: In collaboration with Airtel, users can get their account recharged.
* Generate Account Statements



# Architecture of Online Banking (Green Bank)

The Online Banking Application is based on 3-tiered model. The Enterprise architecture for

Online Banking Green Bank).

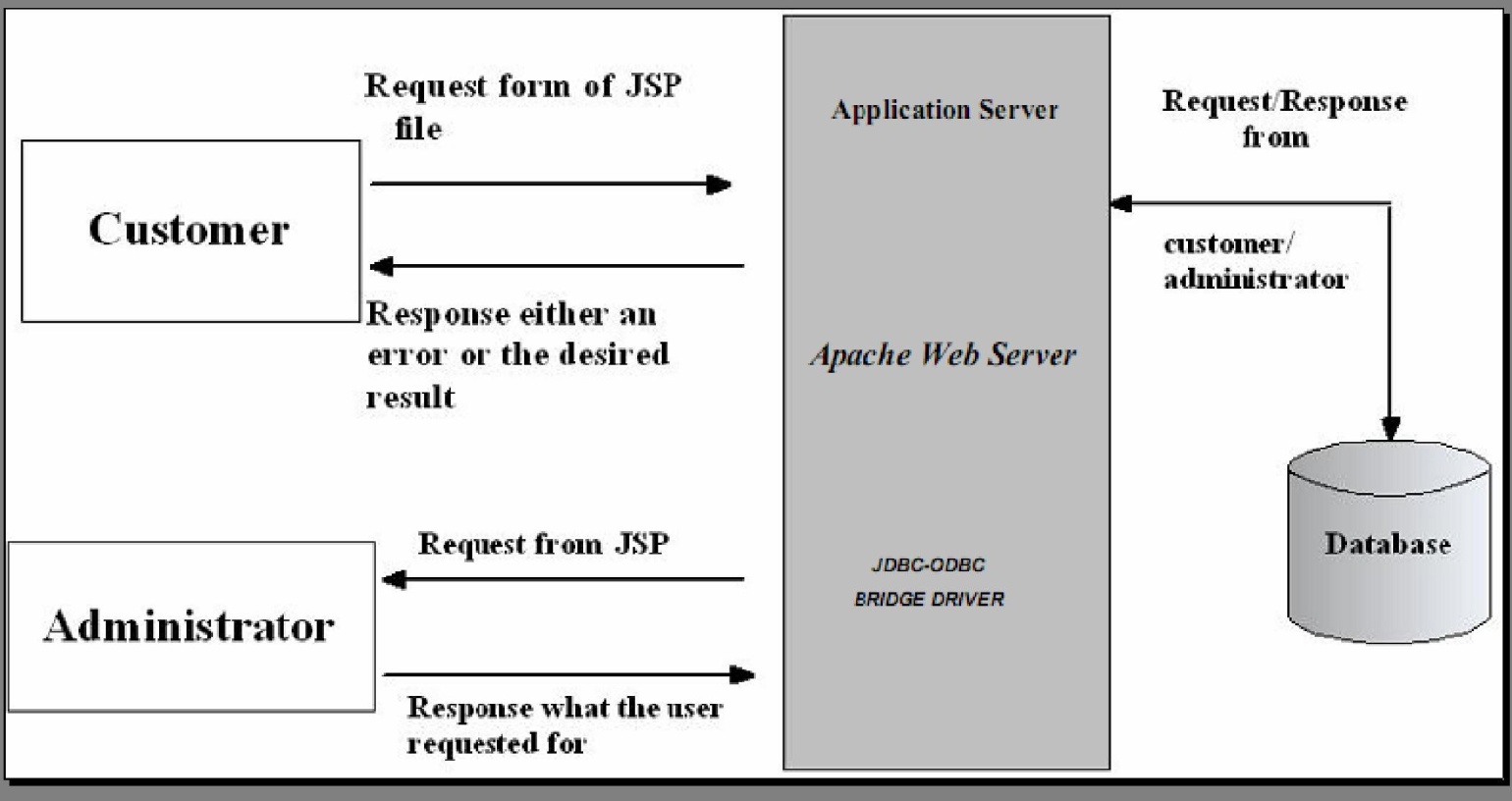


Figure1: Architecture of Online Banking (Green Bank)

The 3-tiered architecture shown above has the following major components:

* Client: There will be two clients for the application. One will be a web-based user-friendly client called bank customers. The other will be for administration purposes.
* Application Server: It takes care of the server script, takes care of JDBC-ODBC driver, and checks for the ODBC connectivity for mapping to the database in order to fulfill client and administrator’s request.
* Database: Database Servers will stores customer’s and bank data.

# Software Development Methodology

Analysis

The table below lists the functionalities to be included in the Online Banking Application, as well as certain features that will not be supported. This list is a tentative, since it may be discovered during development that additional features are required or some existing features may prove to be unworkable or impractical due to time limitations. The application will be written Java Server Pages (JSP), as it is easier to write and maintain pages. Table 1 represents the features supported by Online Banking

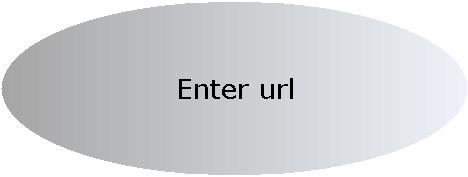
|  |  |
| --- | --- |
| Feature | Support |
| Support 3-tier architecture(Client, Server,  Database) | yes |
| Easy means of navigation through pages containing proper session tracking. | yes |
| Easy means of navigation | Yes |
| Support of application server | Tomcat |
| Support JDBC-ODBC connections | yes |
| Menus created with the help of flash | yes |
| Database support | Yes(Oracle 8.1.7) |
| Establish Database Connection | yes |
| Generation of Use Case and Dataflow diagrams | yes |
| Programming language support | yes |
| Operating system | Windows 98, 2000,Xp,Vista |

Table1 Functionalities of Online Banking









# Software and Hardware Requirements

* Hardware requirements: P4 (2.4 GHz), 512 mb DDR ram
* Operating System: Windows Xp(Professional) SP2
* Software Requirements: Eclipse IDE, Apache tomcat 6.0.18, JavaScript, HTML, JSP and Servlets, Oracle 8.1.7

# Limitations

* The project can be best viewed on Firefox.
* It provides banking facilities to personal banking.
* Enterprise Java Beans have not been implemented.
* SSL is not implemented.
* Loan facilities, credit card login is not being provided.

# Conclusion

This project shows, how Online applications like online Banking, can easily be developed using Enterprises Java Technologies and its distributed architectures \[Page-Centric,

Page-View, and Model 1]. A more efficient approach would be to utilize Model View Controller 2 architecture, which makes application reusable, robust, and more object- oriented the approach followed in this project.

# Proposed System

Those that do are sometimes referred to as "brick-to-click" banks, both to distinguish them from brick-and-mortar banks that have yet to offer online banking, as well as from online or "virtual" banks that have no physical branches or tellers whatsoever.

The challenge for the banking industry has been to design this new service channel in such a way that its customers will readily learn to use and trust it. After all, banks have spent generations earning our trust; they aren't about to risk that on a Web site that is frustrating, confusing or less than secure.

The proposed system will show the Online using Servlets and JSP and Oracle8i as the backend. This project makes use of flash navigation menu bars which makes it easier to navigate through the site and the project is build around MVC architecture which makes it more robust system.The vision is simple and powerful: customers will be able to perform banking functionalities in an easy manner .

# Definition of Problem

The purpose of this project is to develop an on-line banking system that provides customers with the facility to check their accounts and do transactions on-line. The system will provide all the banks facilities to its customers when their authentications [user id and password] matches, including viewing account information, performing fund transfers, giving the customer an option of changing address, password retrieval, performing transactions, viewing transactions and locate bank branches. The system allows and should allow customers to view their personnel accounts and to pay bills online from their account.

The system should assign a unique transaction number to every transaction that a user makes.. The Administrator will administer both normal bank account. The administrator have the ability to perform various operations like creating a normal bank account for the customer and performing functions like transfers, withdrawals and deposits when the customers want teller transactions. The administrator also has the privilege to close the customer’s account on the request of the bank customer. The customer should be able to access his/her account from anywhere just by inputting the correct user-id and password.

# Advantages of online banking

* Convenience: Unlike your corner bank, online banking sites never close; they're available 24 hours a day, seven days a week, and they're only a mouse click away.
* Ubiquity: If you're out of state or even out of the country when a money problem arises, you can log on instantly to your online bank and take care of business, 24/7.
* Transaction speed: Online bank sites generally execute and confirm transactions at or quicker than ATM processing speeds.
* Efficiency: You can access and manage all of your bank accounts, including IRAs, CDs, even securities, from one secure site.
* Effectiveness: Many online banking sites now offer sophisticated tools, including account aggregation, stock quotes, rate alerts and portfolio managing programs to help you manage all of your assets more effectively. Most are also compatible with money managing programs such as Quicken and Microsoft Money.

# Disadvantages of online banking

* Start-up may take time: In order to register for your bank's online program, you will probably have to provide ID and sign a form at a bank branch. If you and your spouse wish to view and manage your assets together online, one of you may have to sign a durable power of attorney before the bank will display all of your holdings together.
* Learning curve: Banking sites can be difficult to navigate at first. Plan to invest some time and/or read the tutorials in order to become comfortable in your virtual lobby.
* Bank site changes: Even the largest banks periodically upgrade their online programs, adding new features in unfamiliar places. In some cases, you may have to re-enter account information.
* The trust thing: For many people, the biggest hurdle to online banking is learning to trust it. Did my transaction go through? Did I push the transfer button once or twice? Best bet: always print the transaction receipt and keep it with your bank records until it shows up on your personal site and/or your bank statement.

# Theoretical Background

Similar banking applications are available in the market. A considerable amount of research has been done in past few months on this project. Many banks had migrated from paper based banking system to electronic online banking. Each bank had its own, user friendly interface, which helps its customers to interact with their account at their ease. A wide variety of Online Banking applications are available in the market which in turn help the bank to function smoothly. All banks which are using online banking application uses the same basic principle.

Axis Bank

has excellent features, which allows customers to check their accounts and view their statements. The best thing about this bank’s system allows us to schedule payments and do online transactions. The security feature is the best; it gains the trust of the customer and allows them to do their transitions in an efficient and secure manner. However, the interface is very complicated for novice users. The interface for credit card customer is confusing

Canara Bank

One of the finest bank of India , who perform their transactions online. The security issues are wonderful, and it allows the customer to view their transactions, pay bills online, ATM/branch locator and provide

# Project Scope

The Online banking Application project will be divided into 2 modules namely:

1. Bank Account
2. Bank Account Administrator Module 1

In this module the customer is allowed to logon to the website and can access his/her account by getting user name and password which will be verified with the server and the database. Once he/she gets verified then they are allowed to view their personal account and perform operations such as change of address, paying bills online, viewing transactions and transferring money into other accounts. Once the customer finishes the task the update information instantly gets stored into the database. The customer is then allowed to sign out from his/her account.

Module 2

In this module the administrator is allowed to log on to the website and can access his/her administrative account by using the user name and password which will then be verified with the database. Once he/she gets verified the administrative interface will be displayed, where the administrator can perform operations for both new customers and existing customers. Administrator will help a new customer in opening their account by taking complete information from them. Administrator provides services like withdrawal, deposit, transfer and deleting customer during the time of closing the account. In this module administrator provides great customer service to the customers who want to do phone banking or teller banking. The interface for administrator will be both very user friendly and efficient. The data gets stored in the database instantly when the administrator hits the submit button.

# Facilities provided

* View Account: Customer is authorized to check his account.
* Transfer Money: Customer can transfer money online over a secure passage to another account.
* Credit Card Facility: Customer can login using the credit card number. A customer can also apply for a credit card.
* Application Forms: Different forms such as new cheque book request, change of address etc can be downloaded.
* Customer Help: A document enabling customers to understand Online Banking.
* Mobile Recharge: In collaboration with Airtel, users can get their account recharged.
* Generate Account Statements

# Requirements Analysis

The requirements can be defined as “A complete understanding of the software development effort. No matter how well designed or well coded, a poorly analyzed and specified program will disappoint the user and bring grief to the developer.”

System requirements should set out what the system should do rather than how this is done. A requirement may be a functional requirement , that is, it describes a system service or function. Alternatively, it may be a non-functional requirement . A non-functional requirement is a constraint placed on the system (for instance, the required response time) or on the development process (such as the use of a specific language standard).

The Design process involves developing several models of the system at different levels of abstraction. As a design is decomposed, errors and omissions in earlier stages are discovered. Design is a creative process. Although methods and guidelines are helpful, judgment and flair are still required to design a software system.

Requirements may be either:

* Functional, describing the facilities of the system
* Non-functional, describing constraints which may impact later development phases

# Feasibility Study

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is known as feasibility. It is very necessary to evaluate the feasibility of a project. In the development of the present project, no such limitations were imposed that were not feasible.

All the projects are feasible given unlimited resources and infinite time. Unfortunately the development of a computer-based system is more likely to be plagued by a scarcity of resources and difficult delivery date. Therefore it is both necessary and prudent to evaluate the feasibility of a project. Hence this is the most important aspect in analysis part of any system. The feasibility study enables us to know whether it is viable to go for computerization or not.

This is the most important aspect in the analysis part of the system. Designing phase is generally dependent upon the feasibility study. The feasibility study enables us to determine whether it is feasible to go for computerization. So keeping this aspect in view the new computerized system has also undergone many checks and then only it was decided to develop the system.

There are different modes of feasibility study.

* + Technical Feasibility The technical feasibility is concerned with specifying the equipments and software that will successfully satisfy the user requirements. A system is technically feasible if it can be designed and implemented with the limitations of available resources.

The technical requirement i.e. the hardware and software required by the new system is of low cost and needing very less maintenance cost. There is no need to buy anything extra for the system. The projected resources are enough for the system. So the development of the new system is technically feasible.

The proposed system is technically feasible as the resources that are needed for the proposed system are available.

* + Economic Feasibility

Economic feasibility analysis is most frequently used technique for evaluating the effectiveness of the proposed system. It is also known as Cost/Benefit Analysis. The proposed computerized system is economically feasible, if the benefit in the long run compensates the cost incurred in the designing and implementation of the system.

The cost factor of the system involves the Manpower cost, Hardware cost, and the Software cost.

Manpower Cost : There is no need to employ any extra person to use this system as its very user friendly and very easy to learn too.

Hardware Cost : The hardware cost is very less. So this system is economically regarding hardware cost.

Software Cost: The software needed for the development of the project is nil as we are using Free and Open Source Software (FOSS) like Java.

The development of the proposed system does not result in too much extra expenditure. The existing hardware and software will fulfill the requirement of the proposed system. Thus, the proposed system is economically feasible.

* Social Feasibility

Social feasibility is a determination of whether a proposed system will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed change in system.

The proposed system helps to reduce the time of the user. So, the proposed system is accepted by the user. Thus, the proposed system is socially feasible.

* Legal Feasibility

Legal feasibility is a determination of whether the proposed system infringes on any known Acts or pending legislations. We are using Free and Open Source Software (FOSS) so the proposed system does not infringe on any legal Acts. Thus, it is legally feasible.

* Time Feasibility

Time feasibility determines whether the proposed project can be implemented fully within the stipulated time frame.

The proposed system was allotted two months time for being implemented. The time allotted for the proposed system is quite sufficient. Thus, the proposed system is time feasible.

Thus, the proposed system passes all the feasibility tests.

# Hardware/Software Specifications

SOFTWARE REQUIREMENTS

* Front End Tools: Servlets and JSP.
* Back End Tools: Oracle8i
* Server: Apache Tomcat Server

HARDWARE REQUIREMENTS

* Pentium III 600 MHz and above.
* RAM 64 MB and above.

# Database Design

Customer table

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| Customer\_id (PK) | NOT NULL | INTEGER |
| Cust\_first\_name |  | VARCHAR2(20) |
| Cust\_last\_name |  | VARCHAR2(20) |
| DOB |  | VARCHAR2(20) |
| Gender |  | VARCHAR2(2) |

Customer Detail table

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| Customer\_id (FK) | NOT NULL | INTEGER |
| City |  | VARCHAR2(20) |
| State |  | VARCHAR2(20) |
| Zip |  | VARCHAR2(20) |
| Phone Number |  | NUMBER(10) |
| Email id |  | VARCHAR2(20) |

Login table

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| Customer\_id (FK) | NOT NULL | INTEGER |
| Password |  | VARCHAR2(30) |
| Username |  | VARCHAR2(30) |

Account table

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Null? | | Type |
| Account Number (PK) |  | | NUMBER(8) |
| Customer\_id (FK) | NOT NULL | | INTEGER |
| Min\_Balance |  | | NUMBER(8) |
| Current balance |  | | NUMBER(8) |
| Recommended by |  | | VARCHAR2(20) |
| Nominee |  |  | VARCHAR2(20) |
| Type\_of\_account |  | | VARCHAR2(20) |
| Date\_of\_opening |  | | VARCHAR2(20) |
| Date\_of\_access |  | | VARCHAR2(20) |

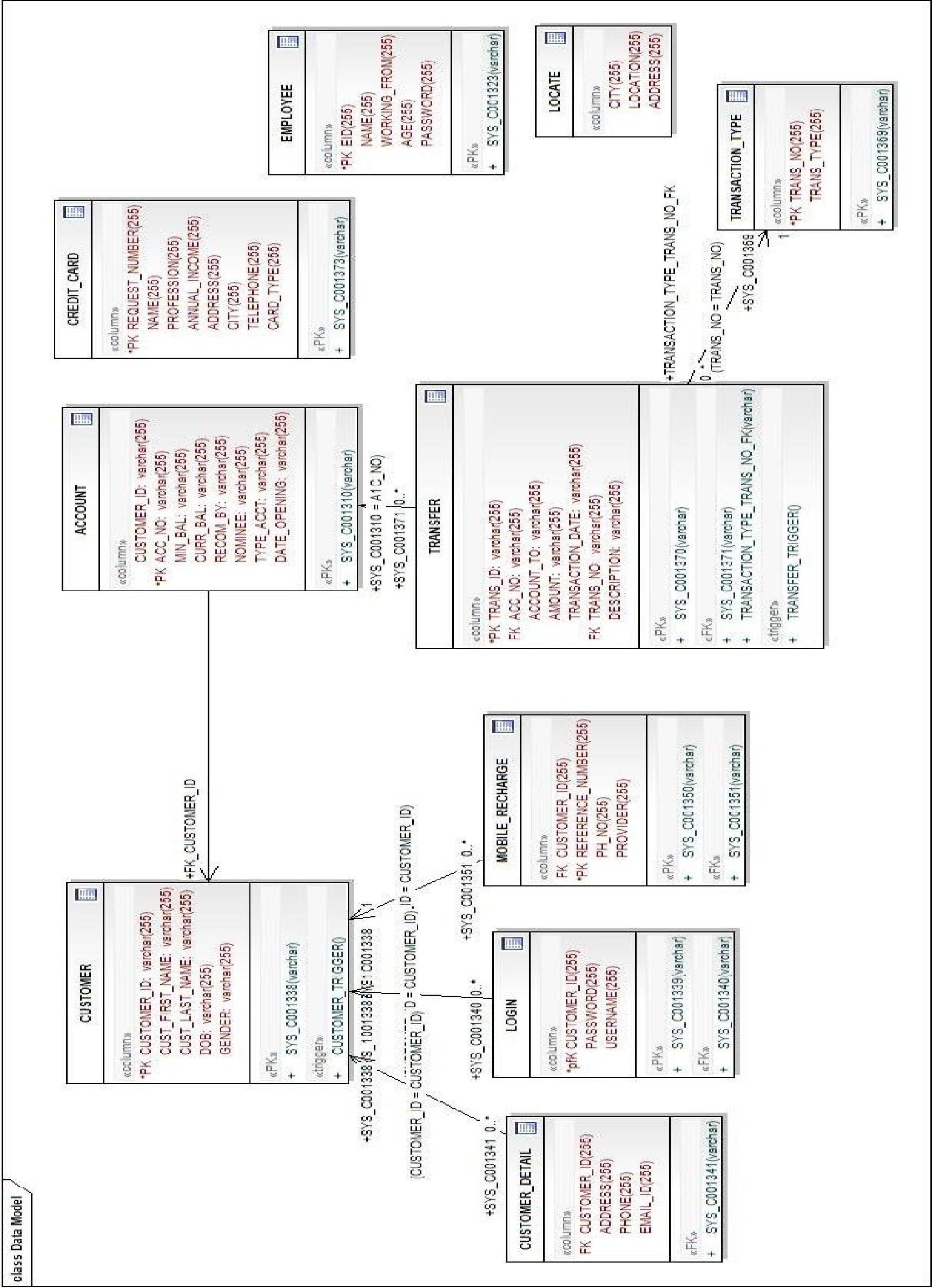
Transaction(transfer-funds) table

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| Trans\_id | NOT NULL | NUMBER(10) |
| Acc\_no |  | NUMBER(10) |
| Account to |  | NUMBER(10) |
| Amount |  | NUMBER(10) |
| Transaction\_date |  | VARCHAR2(20) |
| Trans\_no |  | INTEGER |
| description |  | VARCHAR2(30) |

Transaction type table

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| Transaction Number (PK) |  | INTEGER |
| Account Number (FK) |  | INTEGER |

# Entity Relationship Diagram



Data Flow Diagrams

Context Level Diagram



dfd Data Model

Onlie banking

Web Solution

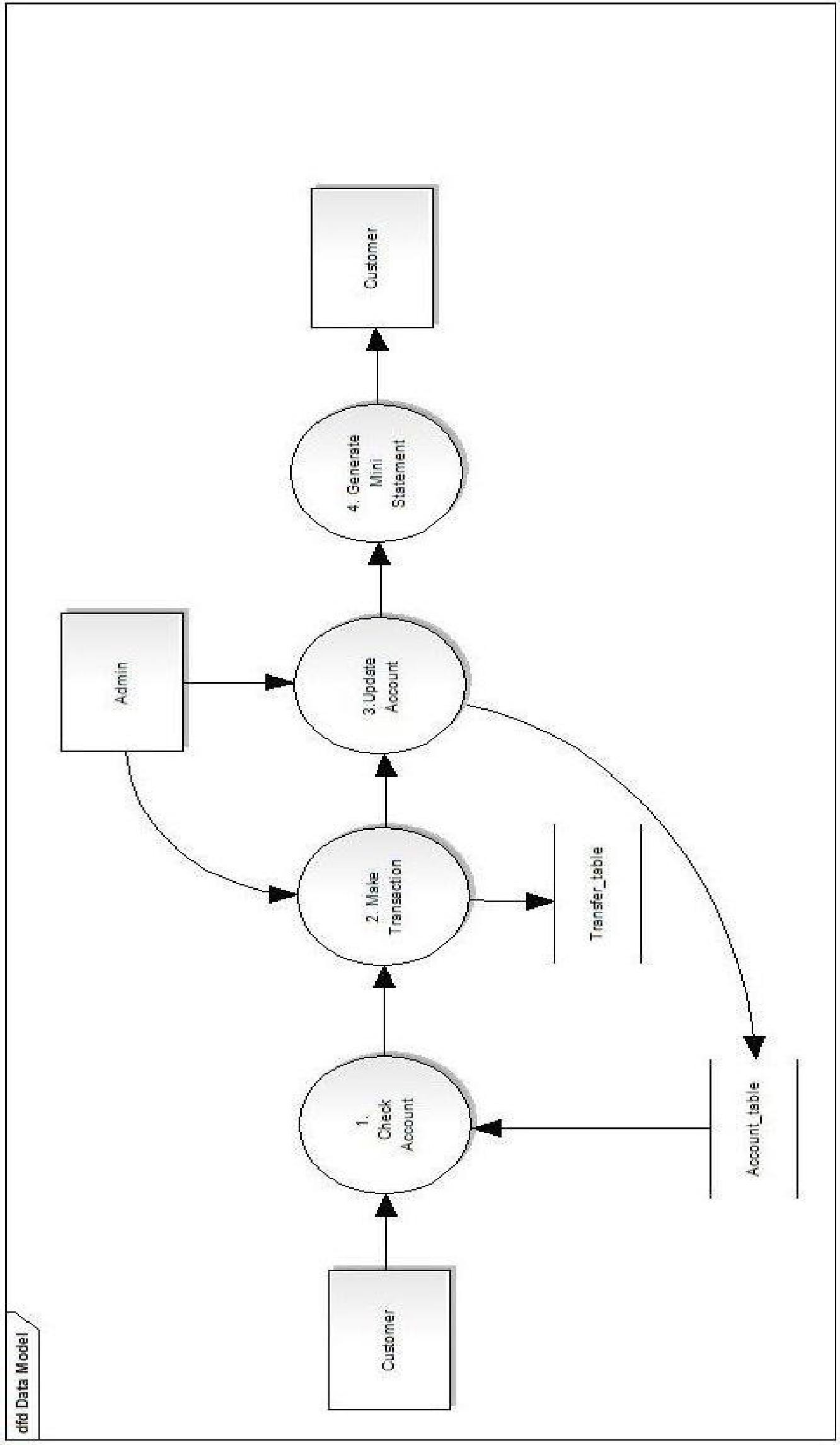
Administrator

Administrator services

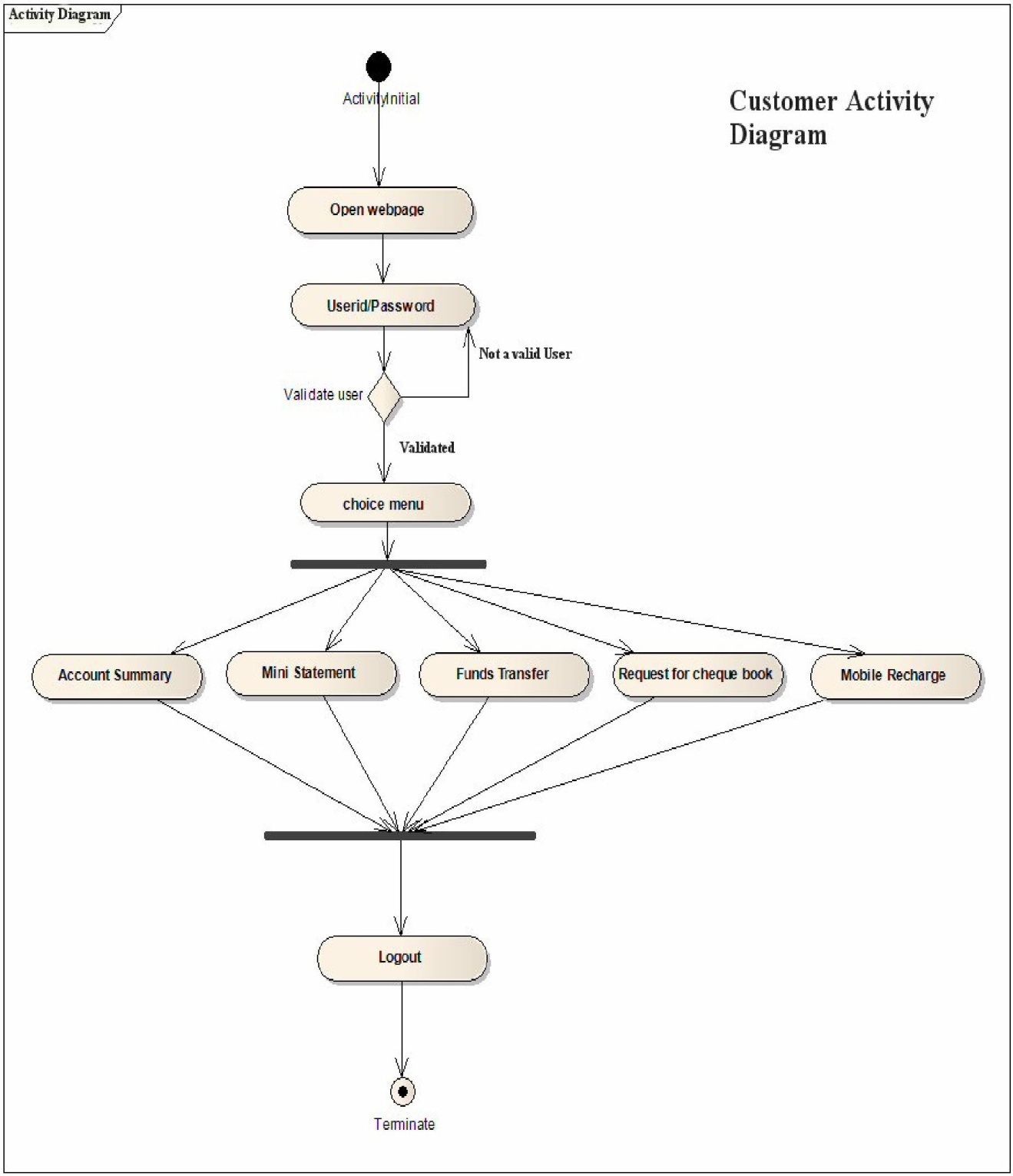
customer

customer services

Level 1 Diagram



Activity Diagram



# Comparison between JSP and other technologies

JSP vs. Active Server Pages(ASP)

ASP is a similar technology from Microsoft. The advantages of JSP are Saturday, March 13 2004 two fold. First, the dynamic parts written in java , not Visual Basic or other MS-specific language, so its more powerful and easier to use. Second, it is portable to other Operating systems and non-Microsoft Web servers.

JSP vs. JavaScript

Client-side scripting with JavaScript is certainly handy and useful, but it does present several problems, including the following: You must count on the customer's browser to have scripting enabled, which, of course, you can t.

* + Different customers may use different browsers. And coding client-side scripts that work on different browsers can be a headache.
  + Scripting languages used on the client side cannot match the strength and versatility of Java.
  + Client-side scripting languages have very limited access to server-side resources, such as databases. JavaServer pages have access to all server- side resources within the well-defined architecture of J2EE.
  + You have the usual problems of maintaining software on the client that caused

Advantages of using JSP over competing technologies are as follows:

* + - JSP enables a clean separation of business logic from presentation.
    - JSP, by using Java as the scripting language, is not limited to a specific vendor s

platform.

* + - JSP, as an integral part of the J2EE architecture, has full access to server-side resources.
* Because JSP pages execute on the server, you need not require the client to use a particular browser or have a fixed configuration.

JSP vs. Pure Servlets

JSP doesn’t give anything that you couldn't in principle do with a servlet. But it is more convenient to write and modify regular HTML than have million of println statements

# Java Server Pages

While there are numerous technologies for building web applications that serve dynamic content, the one that has really caught the attention of the development community is JavaServer Pages (JSP). And not without ample reason either. JSP not only enjoys cross-platform and cross-Web-server support, but effectively melds the power of server-side Java technology with the WYSIWYG features of static HTML pages.

JSP pages typically comprise of:

* Static HTML/XML components.
* Special JSP tags
* Optionally, snippets of code written in the Java programming language called "scriptlets."

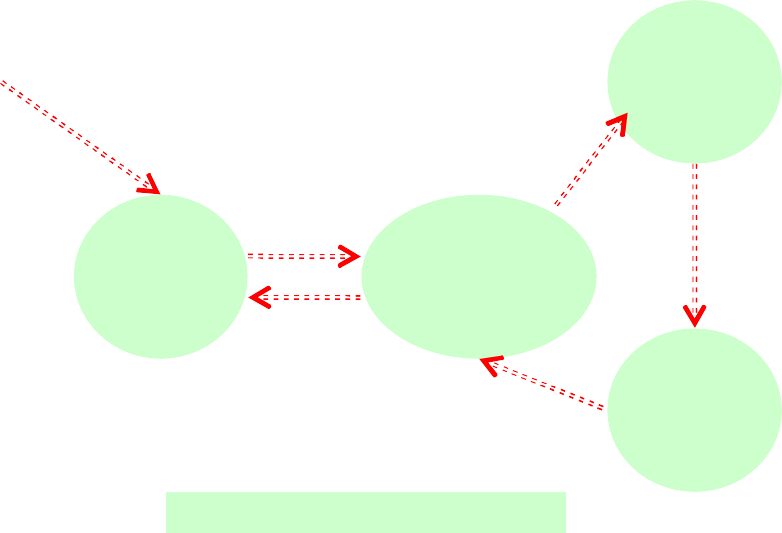
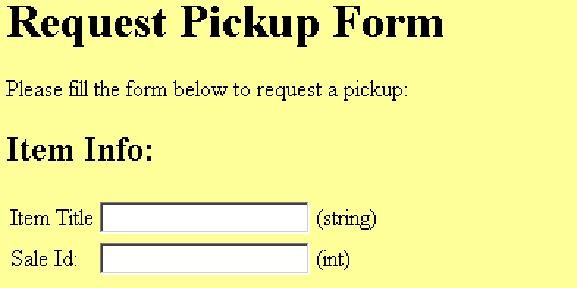
Consequently, you can create and maintain JSP pages by conventional HTML/XML tools.

It is important to note that the JSP specification is a standard extension defined on top of the Servlet API. Thus, it leverages all of your experience with servlets.

There are significant differences between JSP and servlet technology. Unlike servlets, which is a programmatic technology requiring significant developer expertise, JSP appeals to a much wider audience. It can be used not only by developers, but also by page designers, who can now play a more direct role in the development life cycle.

Another advantage of JSP is the inherent separation of presentation from content facilitated by the technology, due its reliance upon reusable component technologies like the JavaBeans component architecture and Enterprise JavaBeans technology.

JSP Architecture



JSP

Engine

Web Server

Server Extension

Servlet Engine

Client

Server-Side

JSP Advantages

* Separation of static from dynamic content
* Write Once Run Anywhere
* Dynamic content can be served in a variety of formats
* Recommended Web access layer for n-tier architecture.
* Completely leverages the Servlet API.

JSP Syntax Basics

JSP syntax is fairly straightforward, and can be classified into directives, scripting elements, and standard actions.

Directives

JSP directives are messages for the JSP engine. They do not directly produce any visible output, but tell the engine what to do with the rest of the JSP page. JSP directives are always enclosed within the <%@ ... %> tag. The two primary directives are page and include.

Page Directive

Typically, the page directive is found at the top of almost all of your JSP pages. There can be any number of page directives within a JSP page, although the attribute/value pair must be unique. Unrecognized attributes or values result in a translation error. For example,

<%@ page import="java.util.\*, com.foo.\*" buffer="16k" %>

makes available the types declared within the included packages for scripting and sets the page buffering to 16K.

Include Directive

The include directive lets you separate your content into more manageable elements, such as those for including a common page header or footer. The page included can be a static HTML page or more JSP content. For example, the directive:

<%@ include file="copyright.html" %>

can be used to include the contents of the indicated file at any location within the JSP page.

Declarations

JSP declarations let you define page-level variables to save information or define supporting methods that the rest of a JSP page may need. While it is easy to get led away and have a lot of code within your JSP page, this move will eventually turn out to be a maintenance nightmare. For that reason, and to improve reusability, it is best that logic-intensive processing is encapsulated as JavaBean components.

Declarations are found within the <%! ... %> tag. Always end variable declarations with a semicolon, as any content must be valid Java statements:

<%! int i=0; %>

You can also declare methods. For example, you can override the initialization event in the JSP life cycle by declaring:

<%! public void jspInit() {

//some initialization code

}

%>

Expressions

With expressions in JSP, the results of evaluating the expression are converted to a string and directly included within the output page. Typically expressions are used to display simple values of variables or return values by invoking a bean's getter methods. JSP expressions begin within <%= ... %> tags and do not include semicolons:

<%= fooVariable %>

<%= fooBean.getName() %>

Scriptlets

JSP code fragments or scriptlets are embedded within <% ... %> tags. This Java code is run when the request is serviced by the JSP page. You can have just about any valid Java code within a scriptlet, and is not limited to one line of source code. For example, the following displays the string "Hello" within H1, H2, H3, and H4 tags, combining the use of expressions and scriptlets:

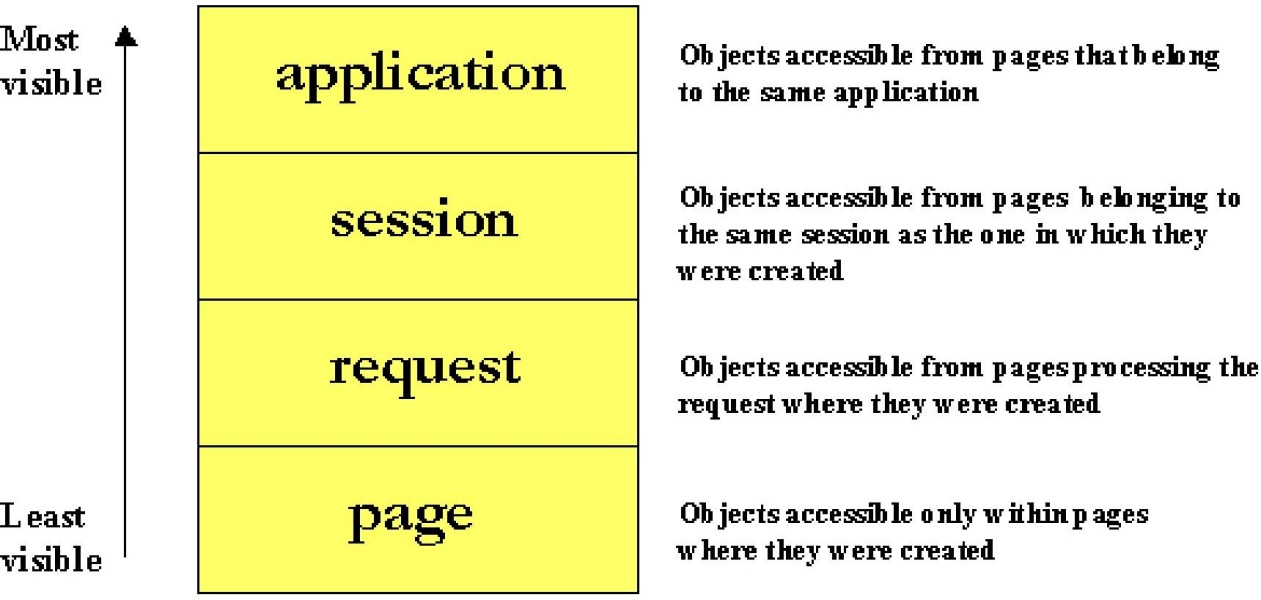
<% for (int i=1; i<=4; i++) { %>

<H<%=i%>>Hello</H<%=i%>>

<% } %>

Object Scopes

Before we look at JSP syntax and semantics, it is important to understand the scope or visibility of Java objects within JSP pages that are processing a request. Objects may be created implicitly using JSP directives, explicitly through actions, or, in rare cases, directly using scripting code. The instantiated objects can be associated with a scope attribute defining where there is a reference to the object and when that reference is removed. The following diagram indicates the various scopes that can be associated with a newly created object:



JSP Implicit Objects

As a convenience feature, the JSP container makes available implicit objects that can be used within scriptlets and expressions, without the page author first having to create them. These objects act as wrappers around underlying Java classes or interfaces typically defined within the Servlet API. The nine implicit objects:

1. request: represents the HttpServletRequest triggering the service invocation. Request scope.
2. response: represents HttpServletResponse to the request. Not used often by page authors. Page scope.
3. pageContext: encapsulates implementation-dependent features in PageContext. Page scope.
4. application: represents the ServletContext obtained from servlet configuration object. Application scope.
5. out: a JspWriter object that writes into the output stream. Page scope.
6. config: represents the ServletConfig for the JSP. Page scope.
7. page: synonym for the "this" operator, as an HttpJspPage. Not used often by page authors. Page scope.
8. session: An HttpSession. Session scope.
9. exception: the uncaught Throwable object that resulted in the error page being invoked. Page scope.

# Servlets

The Java Servlet API allows a software developer to add dynamic content to a Web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to non-Java dynamic Web content technologies such as PHP, CGI and ASP.NET. Servlets can maintain state across many server transactions by using HTTP cookies, session variables or URL rewriting.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of a Web container and a servlet. A Web container is essentially the component of a Web server that interacts with the servlets. The Web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

A Servlet is an object that receives a request and generates a response based on that request. The basic servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package javax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the Web server and a client. Servlets may be packaged in a WAR file as a Web application.

Servlets can be generated automatically by JavaServer Pages (JSP) compiler, or alternately by template engines such as WebMacro. Often servlets are used in conjunction with JSPs in a pattern called "Model 2", which is a flavor of the model-view-controller pattern.

The Servlet lifecycle consists of the following steps:

* 1. The Servlet class is loaded by the container during start-up.
  2. The container calls the init() method. This method initializes the servlet and must be called before the servlet can service any requests. In the entire life of a servlet, the init() method is called only once.
  3. After initialization, the servlet can service client-requests. Each request is serviced in its own separate thread. The container calls the service() method of the servlet for every request. The service() method determines the kind of request being made and dispatches it to an appropriate method to handle the request. The developer of the servlet must provide an implementation for these methods. If a request for a method that is not implemented by the servlet is made, the method of the parent class is called, typically resulting in an error being returned to the requester.
  4. Finally, the container calls the destroy() method which takes the servlet out of service. The destroy() method like init() is called only once in the lifecycle of a Servlet.

Servlet containers

A Servlet container is a specialized web server that supports Servlet execution. It combines the basic functionality of a web server with certain Java/Servlet specific optimizations and extensions – such as an integrated Java runtime environment, and the ability to automatically translate specific URLs into Servlet requests. Individual Servlets are registered with a Servlet container, providing the container with information about what functionality they provide, and what URL or other resource locator they will use to identify themselves. The Servlet container is then able to initialize the Servlet as necessary and deliver requests to the Servlet as they arrive. Many containers have the ability to dynamically add and remove Servlets from the system, allowing new Servlets to quickly be deployed or removed without affecting other Servlets running from the same container. Servlet containers are also referred to as web containers or web engines.

Like the other Java APIs, different vendors provide their own implementation of the Servlet container standard. For a list of some of the free and commercial web containers, see the list of Servlet containers. (Note that 'free' means that non- commercial use is free. Some of the commercial containers, e.g. Resin and Orion, are free to use in a server environment for non-profit organizations).

# MVC Architecture

The main aim of the MVC architecture is to separate the business logic and application data from the presentation data to the user.

Here are the reasons why we should use the MVC design pattern.

1. They are resuable : When the problems recurs, there is no need to invent a new solution, we just have to follow the pattern and adapt it as necessary.
2. They are expressive: By using the MVC design pattern our application becomes more expressive.
3. Model: The model object knows about all the data that need to be displayed. It is model who is aware about all the operations that can be applied to transform that object. It only represents the data of an application. The model represents enterprise data and the business rules that govern access to and updates of this data. Model is not aware about the presentation data and how that data will be displayed to the browser.
4. View : The view represents the presentation of the application. The view object refers to the model. It uses the query methods of the model to obtain the contents and renders it. The view is not dependent on the application logic. It remains same if there is any modification in the business logic. In other words, we can say that it is the responsibility of the of the view's to maintain the consistency in its presentation when the model changes.
5. Controller: Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. After the action has been taken on the data, the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often work very closely together.

Features of MVC2:

1. The MVC2 architecture removes the page centric property of MVC1 architecture by separating Presentation, control logic and the application state.
2. In MVC2 architecture there is only one controller which receives all the request for the application and is responsible for taking appropriate action in response to each request.

# Apache Tomcat

The Apache HTTP Server, commonly referred to simply as Apache, is a web server notable for playing a key role in the initial growth of the World Wide Web. Apache was the first viable alternative to the Netscape Communications Corporation web server (currently known as Sun Java System Web Server), and has since evolved to rival other Unix-based web servers in terms of functionality and performance.

Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The application is available for a wide variety of operating systems, including UNIX, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF, and eComStation. Released under the Apache License, Apache is characterized as free software and open source software.

Since April 1996 Apache has been the most popular HTTP server on the World Wide Web. As of June 2008[update] Apache served 49.12% of all websites.

Features

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support mod\_perl, mod\_python, Tcl, and PHP. Popular authentication modules include mod\_access, mod\_auth, and mod\_digest. A sample of other features include SSL and TLS support (mod\_ssl), a proxy module, a useful URL rewriter (also known as a rewrite engine, implemented under mod\_rewrite), custom log files (mod\_log\_config), and filtering support (mod\_include and mod\_ext\_filter).

Popular compression methods on Apache include the external extension module, mod\_gzip, implemented to help with reduction of the size (weight) of web pages served over HTTP. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different actual websites. For example, one machine, with one Apache installation could simultaneously serve [www.example.com,](http://www.example.com/) [www.test.com,](http://www.test.com/) test47.test- server.test.com, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs) which permit easier, more intuitive configuration of the server.

Usage

Apache is primarily used to serve both static content and dynamic Web pages on the World Wide Web. Many web applications are designed expecting the environment and features that Apache provides.

Apache is the web server component of the popular LAMP web server application stack, alongside MySQL, and the PHP/Perl/Python (and now also Ruby) programming languages.

Apache is redistributed as part of various proprietary software packages including the Oracle Database or the IBM WebSphere application server. Mac OS X integrates Apache as its built-in web server and as support for its WebObjects application server. It is also supported in some way by Borland in the Kylix and Delphi development tools. Apache is included with Novell NetWare 6.5, where it is the default web server.

Apache is used for many other tasks where content needs to be made available in a secure and reliable way. One example is sharing files from a personal computer over the Internet. A user who has Apache installed on their desktop can put arbitrary files in the Apache's document root which can then be shared.

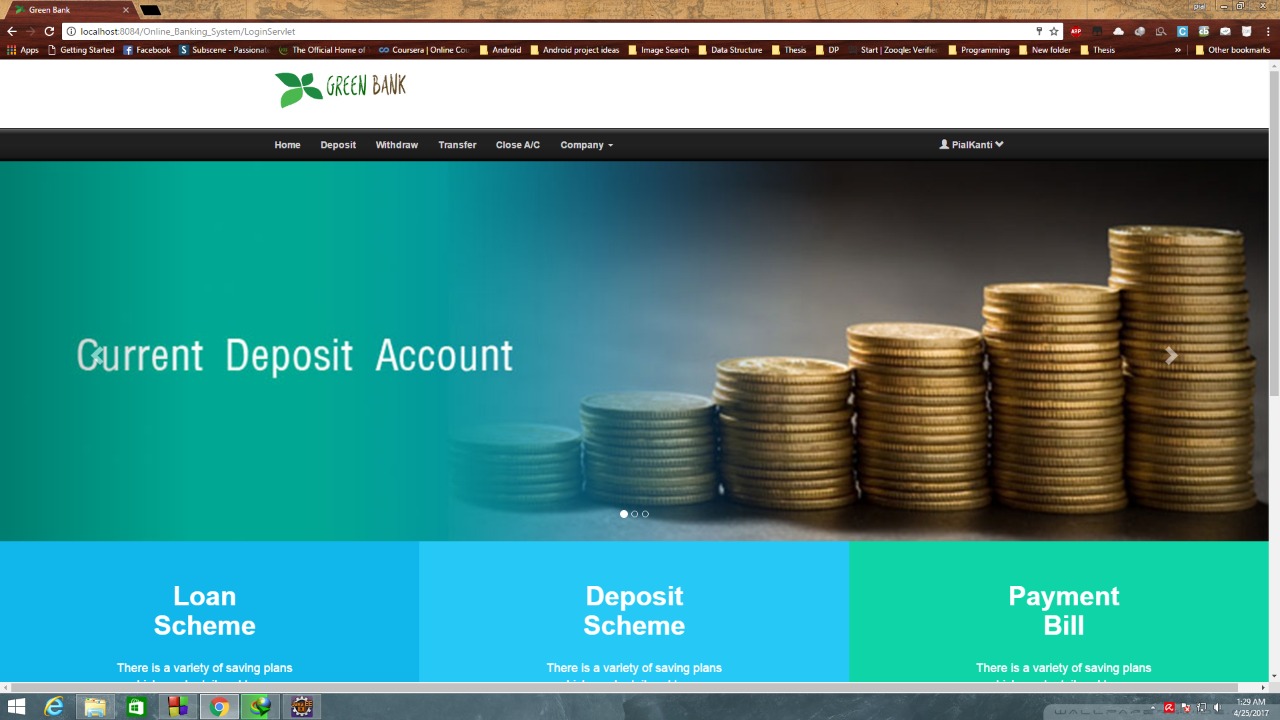
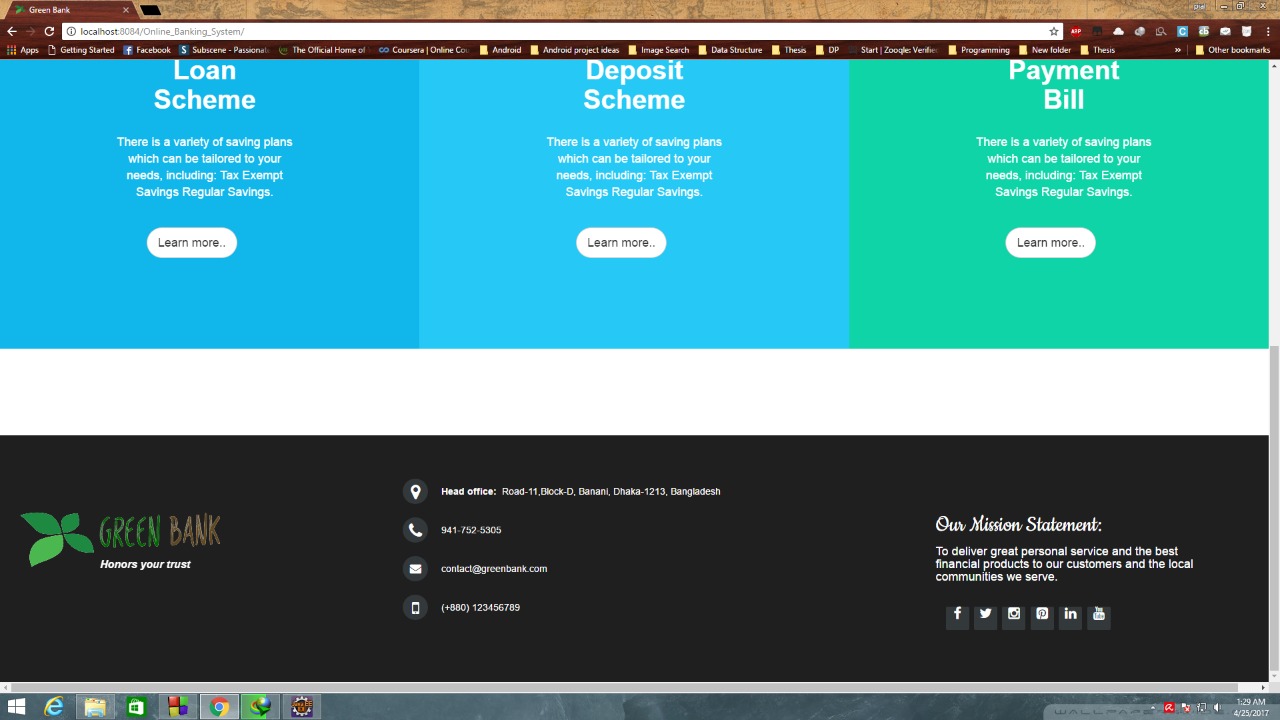
Programmers developing web applications often use a locally installed version of Apache in order to preview and test code as it is being developed.

Microsoft Internet Information Services (IIS) is the main competitor to Apache, trailed by Sun Microsystems' Sun Java System Web Server and a host of other applications such as Zeus Web Server. Some of the biggest web sites in the world are run using Apache. Google's search engine front end is based on a modified version of Apache, named Google Web Server (GWS). Several Wikimedia

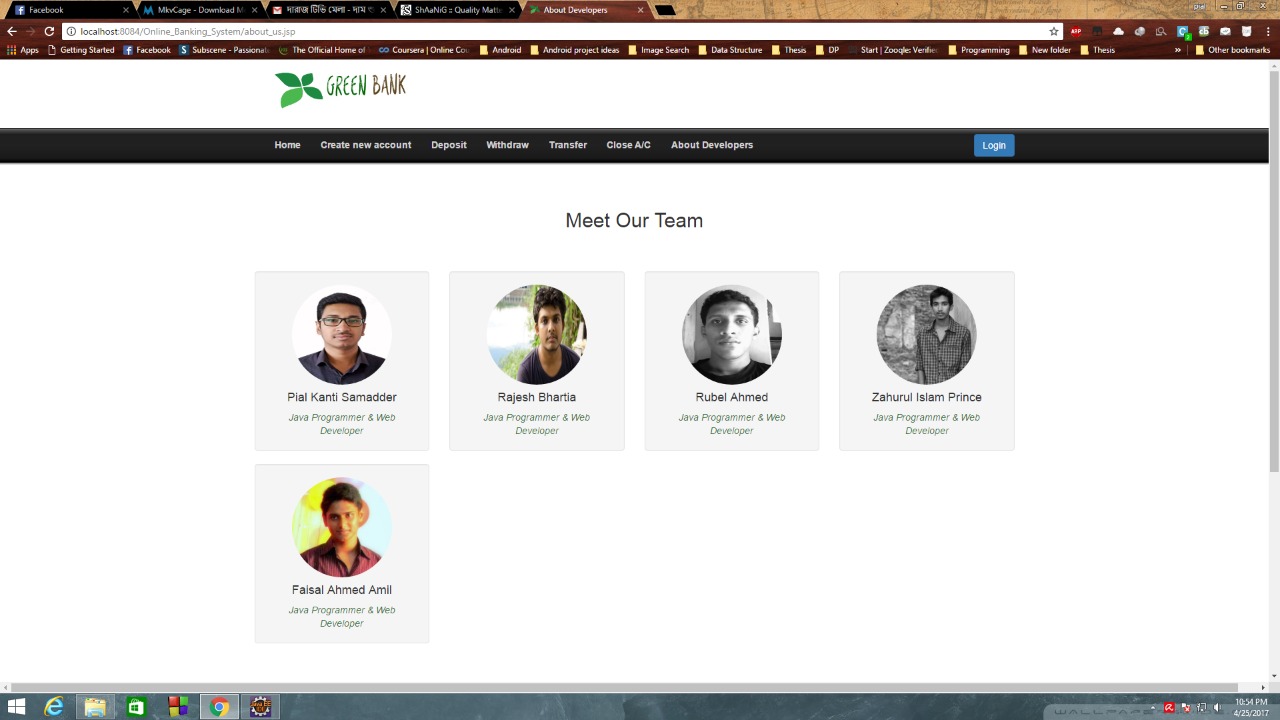
projects also run on Apache servers.

**SCREENSHOTS**

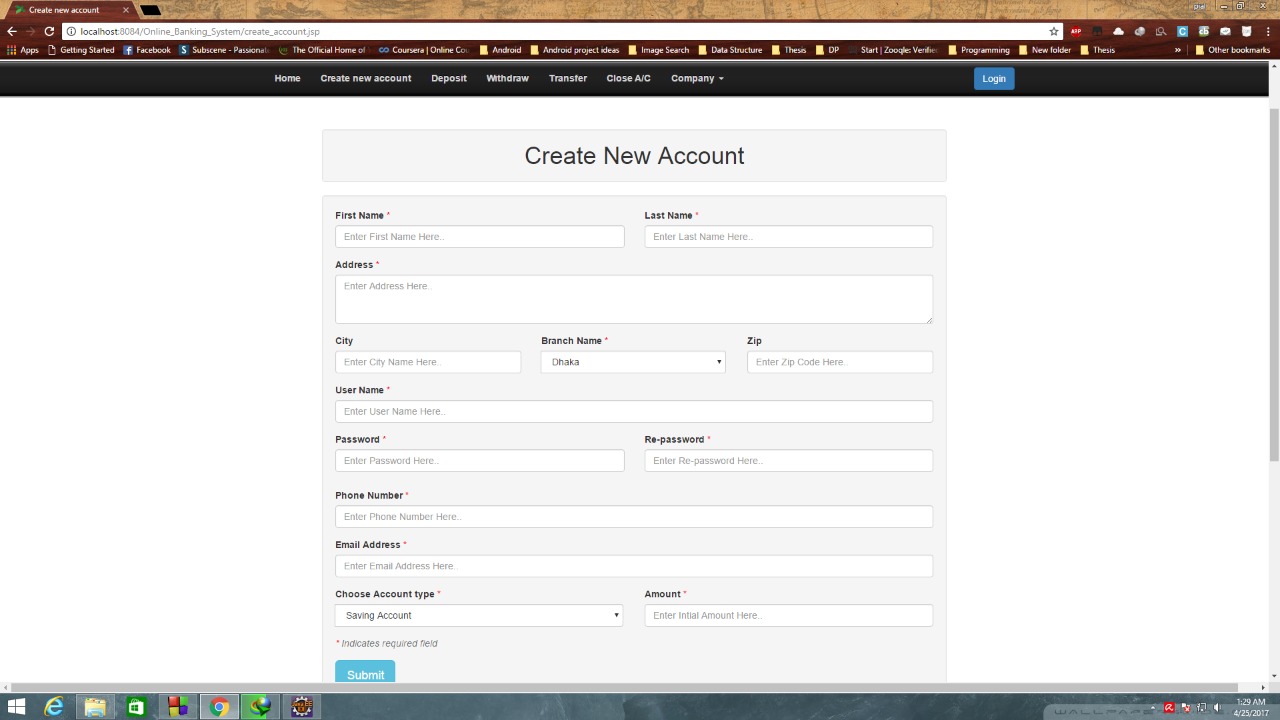
**Main page(HOME PAGE)**

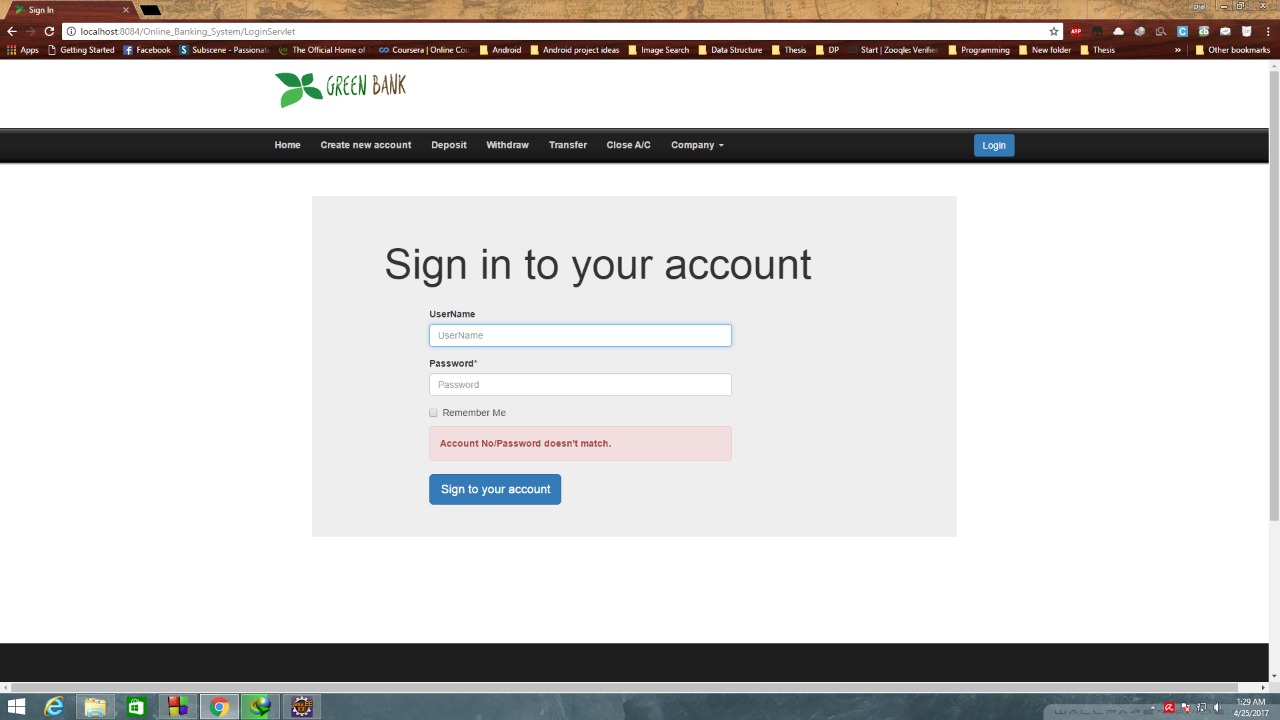
**ABOUT PAGE**



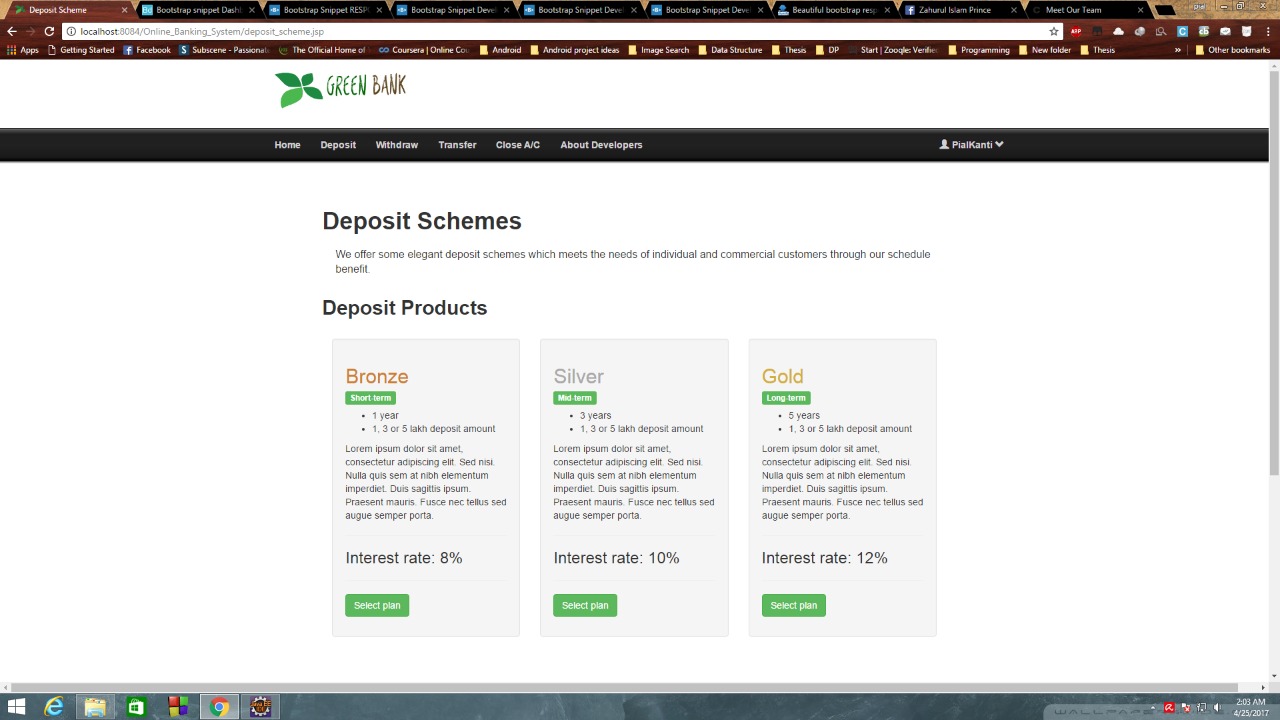
**CREATE ACCOUNT PAGE**



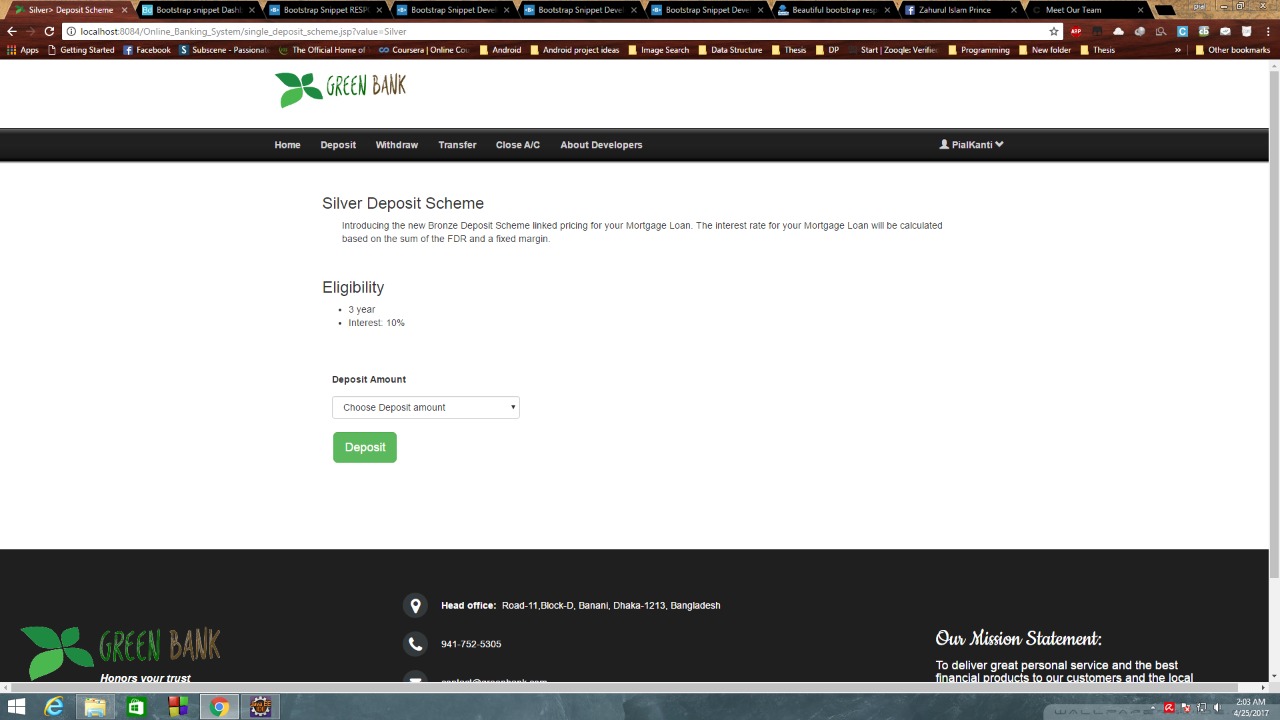
**LOGIN PAGE**



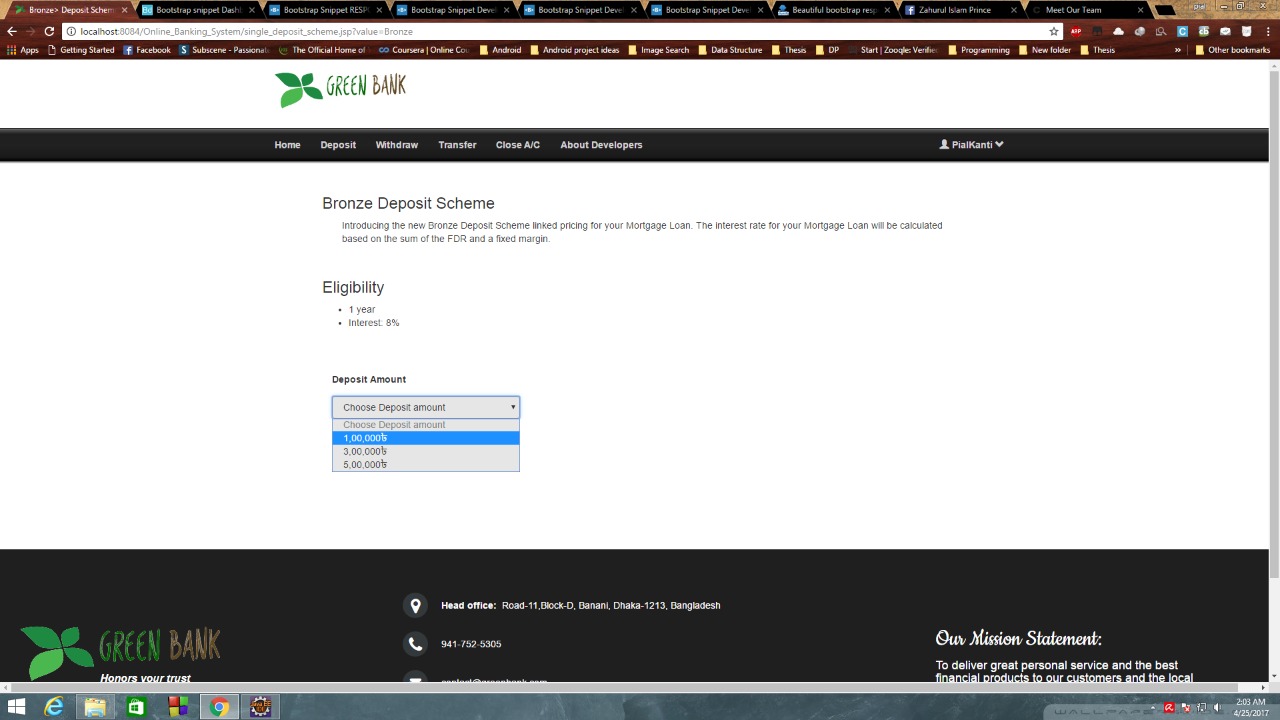
**DEPOSIT SCHEMES PAGE**



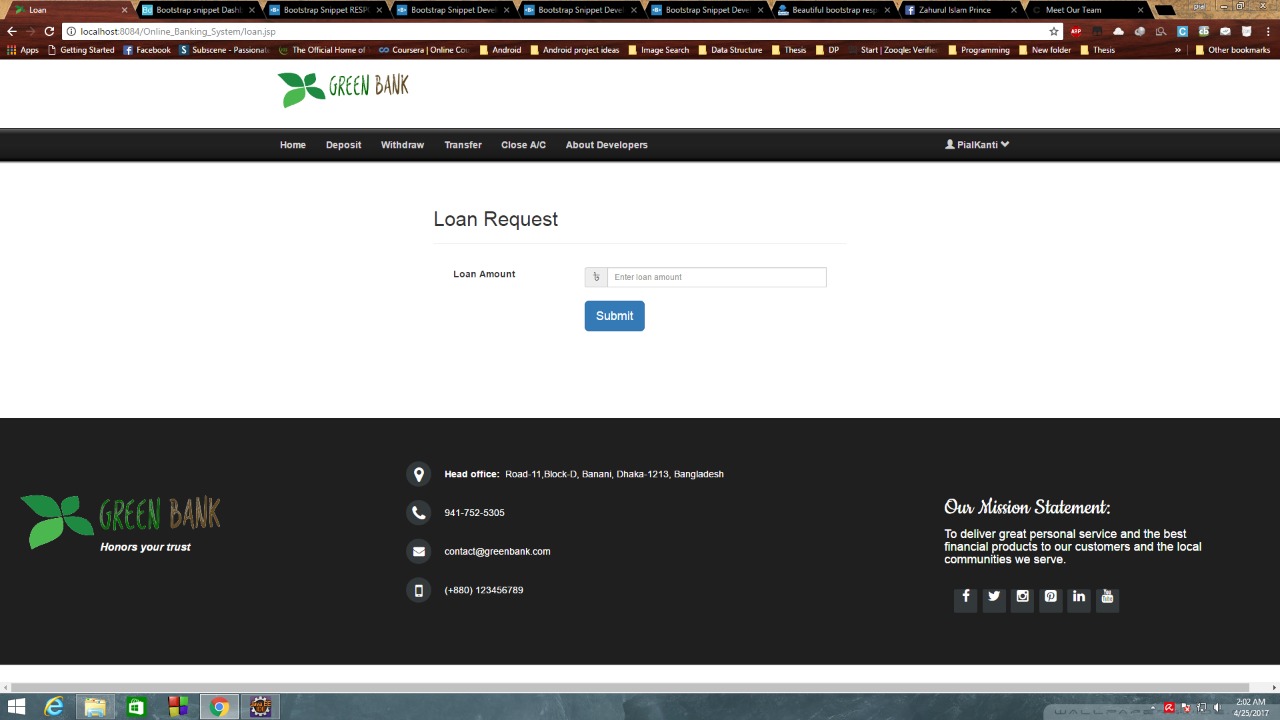
**SILVER SCHEME PAGE**



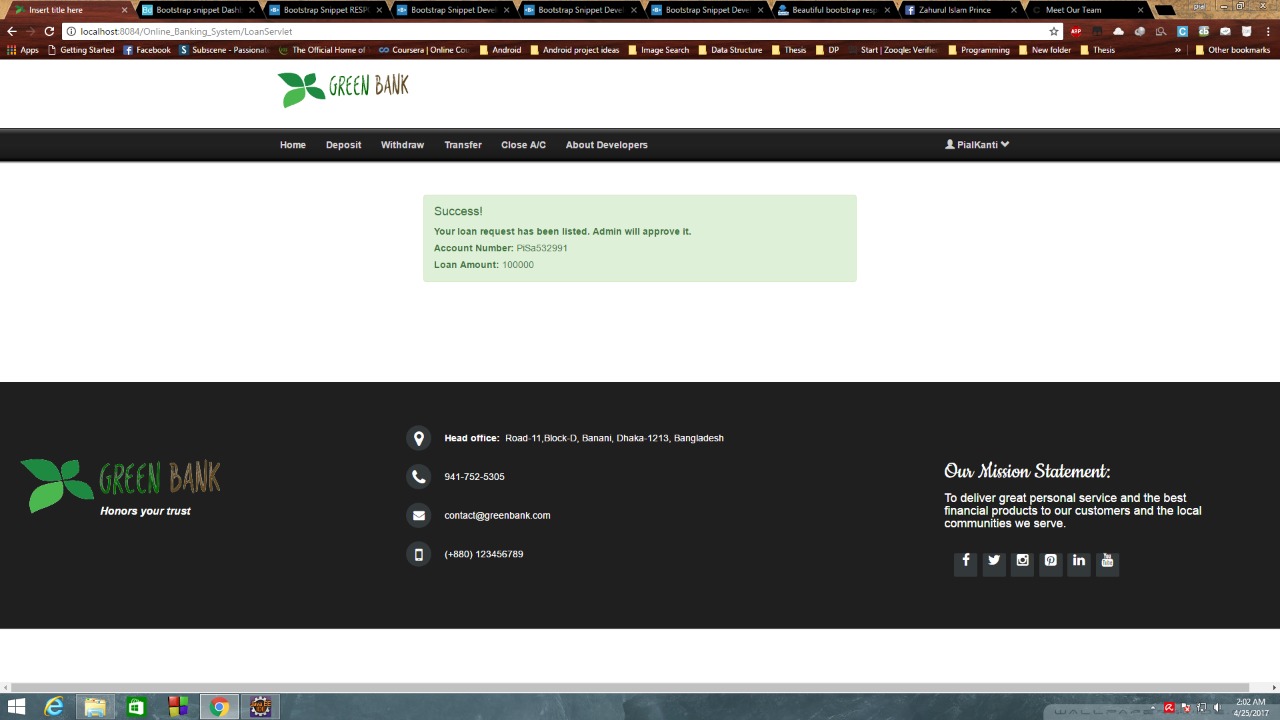
**BRONZE SCHEME PAGE**



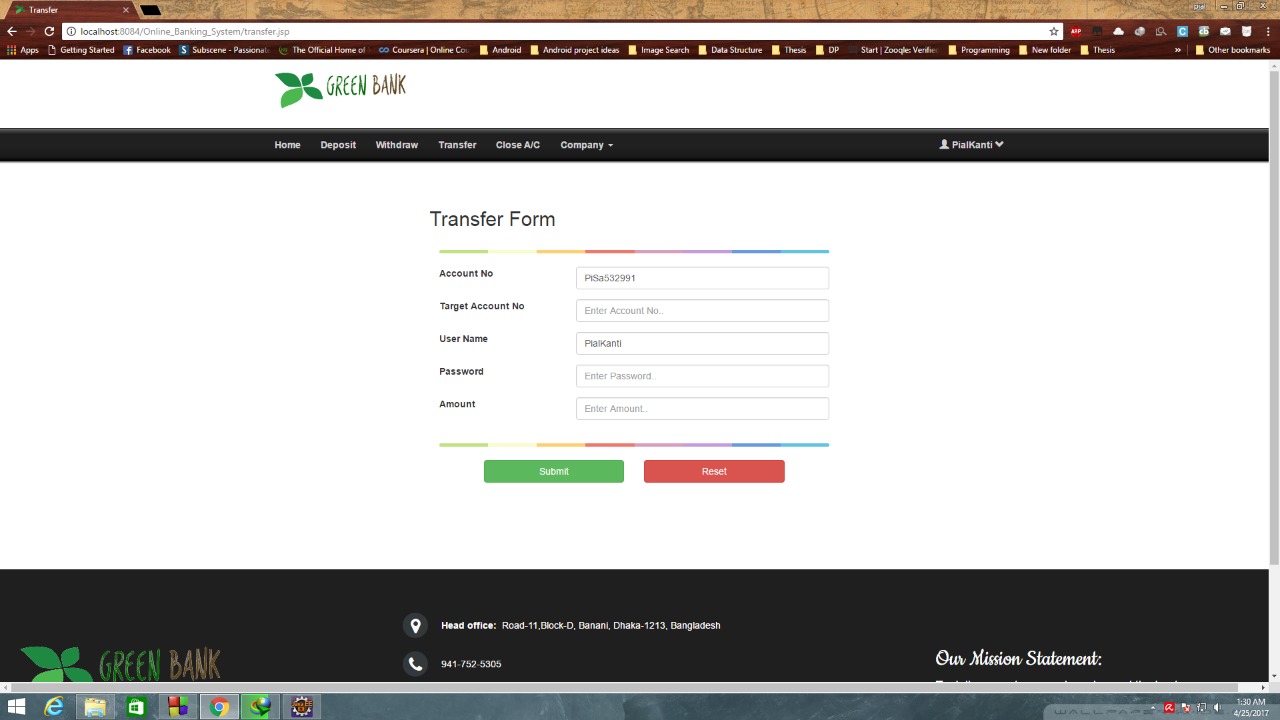
**LOAN REQUEST PAGE**



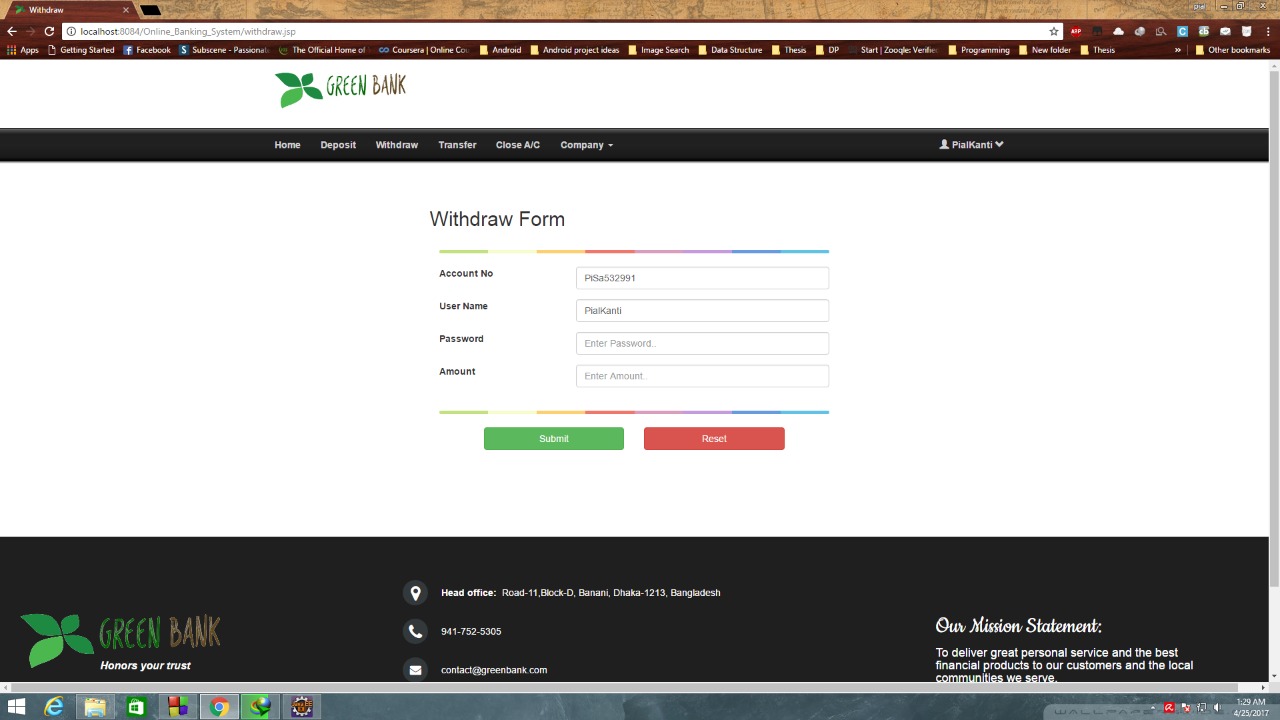
**CONFORMATION PAGE**



**TRANSFER PAGE**



**WITHDRAW PAGE**



# Limitations

* + The project can be best viewed on Firefox.
  + It provides banking facilities to personal banking.
  + Enterprise Java Beans have not been implemented.
  + SSL is not implemented.
  + Loan facilities, credit card login is not being provided.

# Future Enhancements

The Online Banking project can be enhanced in future, which will increase the credibility and reliability of the website. The enhancements could be done by adding credit checks generating credit card statements. The enhancements that could be done are as follows:

* + It can be implemented fully on the Internet by registering a domain and buying a web space.
  + It can provide the Credit card bill payment option.
  + It can provide user loan facilities with interest rates in accordance with bank to.
  + It can be merged with other e-commerce websites for providing a gateway to pay through credit cards .

# Bibliography

Books

* + Kathy Sierra & Bert Bates, Head First JSP and Servlets
  + Kathy Sierra & Bert Bates, Head First Java
  + Chip Dawes & Biju Thomas, Introduction To Oracle 9i
  + Lynn Beighley, Head First SQL
  + Roger Pressman, Software Engineering
  + Michael Morrison, Head First Java Script

Websites

* + [www.sourceforge.net](http://www.sourceforge.net/)
  + [www.javaranch.com](http://www.javaranch.com/)
  + [www.w3schools.com](http://www.w3schools.com/)
  + [www.webreference.com](http://www.webreference.com/)

References

* + [www.icicibank.com](http://www.icicibank.com/)
  + [www.hdfcbank.com](http://www.hdfcbank.com/)
  + [www.unitybanking.com](http://www.unitybanking.com/)