ABSTRACT

This project is aimed at developing an Online Banking for customer. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of Oracle 10g and all the user interfaces have been designed using the JAVA. The database connectivity is planned using the “Database” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as Oracle 10g.The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The Oracle 10g was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the HTML 5. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The database connectivity was planned using the latest “ Database connection” technology provided by Oracle. The authentication and authorization was crosschecked at all the relevant stages. The user level accessibility has been restricted into two zones namely.

Table of Contents

CHAPTER- 1

Introduction

1.1 Overview

Internet Banking is all about knowing our customer need and provide them with the right service at the right time through right channel 24*7 day a week.

Being “electronic”, it not only provides its customers with faster and better facilities, it even reduces the manual overhead of accounts maintenance.

1.2 ABOUT THE PROJECT

APANA-BANK C.P. is one of the most prestigious BANKs in India. Founded as a Public BANK in 1972 in New Delhi, it is a private institution run by the Delhi Public BANK Society.

APANA-BANK, C.P. is affiliated to the Central Board of Bank (CBB), which is the largest educational board in the country. It is recognized by the Department of Education, Govt. of NCT Delhi and the Ministry of HRD, Govt. of India. Over 5000 BANKs in India, with over 80,000 students, are members of the Board.

The BANK is also affiliated to the Indian Public BANKs' Conference (IPSC), and the National Progressive BANKs' Conference (NPSC). The members of these organizations include some of the premier BANKs in the country.

Life at DPSRKP centers on a shared commitment to academic excellence, intellectual growth, art, athletics, high standards of ethical awareness, sportsmanship, and community service. The BANK's traditions and accessibility to a broad curriculum add depth to each student’s life. The BANK upholds the founders' commitment to excellence in all fields, with emphasis on its motto Service Before Self.

1.3 BANK PROFILE:

APANA-BANK, C.P. is a co-educational day-cum-boarding BANK, with approximately 9,500 customer on its rolls. These children, in the Junior and Senior branches, study in the three different campuses at East of Kailash, Vasant Vihar and C.P.

The BANK is among the most distinguished members of the Ravi Public BANK, C.P.. It is a path breaker in the pursuit of excellence. Its endeavor of integrating quality with quantity is reflected in the pivotal role it has played in the setting up of DPS Vasant Kunj, DPS Faridabad and DPS Manali at the national level. It has also promoted three BANKs abroad in Kuwait, Nepal and Indonesia. As their Linking BANK it also co-ordinates their activities.

The BANK has also extended its expertise further and in collaboration with the Government of Haryana, has taken up 3 BANKs in the under-privileged area of Mewat, to augment and enhance their standards and make them more conducive to teaming.

The BANK considers education to be a life-long process which should have a strong foundation. The goal of the BANK is to inculcate in the customer a love for learning and a desire to excel at every level. The BANK also aims at equipping the customer with the intellectual and practical skills that are necessary to meet the challenges in the future.

To sum up, the mission of APANA-BANK, C.P. "to open doors and open minds" and prepare the ground for the future of the nation.

1.4 OUR CULTURE:

In the portals of APANA-BANK, C.P., C.P. customer discover their own talents, and get an opportunity to develop them to the fullest. The BANK provides an invigorating and competitive atmosphere, created by excellent facilities and guidance provided by a highly qualified and dedicated faculty.

The values, which are ingrained help to promote confidence, direction, and critical thinking skills, leading to the development of well-adjusted, adaptable and integrated personalities. In other words, APANA-BANK, C.P. offers comprehensive and holistic education.

Besides being committed to academic excellence and providing education for all round development, another special characteristic of DPS R.K. Puram is the appreciation of the worth of the each student. The BANK is equally committed to the under-represented and less-privileged segments of the population, such as gifted applicants whose parents could not attend BANK, and children with high potential facing difficult financial circumstances.

A major landmark development has been the inclusion of the physically and mentally handicapped children into the mainstream of BANK life. This contributes to a strong sense of community life, so characteristic of the BANK. In other words, children belonging to every strata of society are given the opportunity to study here. The BANK, does not in any way, discriminate on the basis of race, color, religion, sex, caste or creed, in the implementation of its admission policy.

These qualities have placed APANA-BANK, C.P. on the forefront. There has been a continuity of purpose, underlying the change and growth of the BANK. Over the years, APANA-BANK, C.P. has steadily reflected a spirit of innovation in response to need, and has broadened its educational mission, by creating an academic environment that fosters close association and the exchange of ideas with some of the top BANKs in the nation and the world.

Its membership with the IPSC has brought it into regular interaction with BANKs of national standing such as Mayo College, Ajmer; Scindia BANK, Gwalior; The Doon BANK, Dehradun;

Bishop Cotton, Simla Hills; St. Xavier's and La-Martiniere at Calcutta; which has further inculcated a healthy spirit of competition and strong bonds of brotherhood, conducive to national integration.

All the academic programs and activities at APANA-BANK, C.P. work towards one purpose - to help coustemer develop lives of significance for themselves and for others, true to the traditions of the BANK Motto "Service Before Self".

1.5 Purpose:

The Online Banking suite provides a global accounting foundation that provides the all private banks with electronic banking facilities. It allows client of private banks to carry out their day to day banking transactions.

1.6 Scope:

The Online Banking project is widely applicable with private banks. It can even be used in industries for their personal transactions (working).

1.7 Functional components of the project:

Following are the functional needs of the software:-

1. Customer must have a valid user ID and password to login to the system.

2. After the valid user logs in, the system shows the present balance in that particular account number.

3. Customer can perform transactions like deposit and withdrawal from his account.

4. Proper help to be provided as and when requested by the customer.

1.8 More functionality can be added to “enhance the project”:

1. By adding new modules of different accounts like saving A/C, current A/C etc. to facilitate new customers/users.

2. By the use of electronic media, “Digital Signature” on the card can be provided with the customer to make it secure and efficient.

CHAPTER- 2

System Analysis

PRESENT SYSTEM

The developed system is an innovation in the area of private banking. In the existing system the no. of staff required for completing the work is more, while the new system requires lesser staffs generally.

The data entry process requires the data on the paper, which is then feed into the application by the operator while doing so; the data entry operator has to look into the paper again &again and thus the chances of in accuracies in the typed contents increases. Also the process includes higher transportation cost, increased handling cost, more time delays, low accuracy, more usage of resources like registers, books, papers, etc.

PROPOSED SYSTEM

“Why an Automated Private Banking System?”

Almost 60% of today’s information is still paper based.

30% of all office time is spent finding documents.

The average time to manage a single document is 12 minutes,

9 minutes to re-file and 3 minutes to process.

Hence the requirement is to develop a system that minimizes allthese overheads included while giving the maximum output for theorganization.

The basis for the project is to develop a fullyautomated banking system that includes depositing of amount,

withdrawal of amount and exporting the outcome back to the clientwhile considering all the tools and facilities than a client may needfor efficient and effective output.

Benefits of the system

- Quick, authenticated access to accounts via the desktop.
- Easily scalable to grow with changing system requirement.
- Enterprise wide access to information.
- Improved information security, restricting unauthorized access.
- Minimize Storage Space

In manual system, much storage space for data files is required so to overcome this problem, on automated well managed

database is developed for saving storage space. This s/w saves space and stores information efficiently. It ends the burden of having large manual filing storage system.

Banking System can be used extensively

- Withdrawal of amount by the client.
- Deposition of amount by the client.
- Faster balance enquiry.

CHAPTER- 3

FEASIBILITY REPORT

3.1 Understanding Feasibility

Feasibility study means the analysis of problem to determine if It can be solved effectively. In other words it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources.

Three aspects in which the system has to be feasible are:-

3.2 ECONOMICAL FEASIBILITY:

The economical analysis checks for the high investment incurred on the system. It evaluates development &

implementing charges for the proposed "Banking Project". The S/W used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

3.3 TECHNICAL FEASIBILITY:

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus the automated solution leads to the need for a technical feasibility study.

The focus on the platform used database management &users for that S/W.

The proposed system doesn't require an in depth technical knowledge as the system development is simple and easy to understand. The S/W (VB.NET) used makes the system user friendly (GUI). The result obtain should be true in the real time conditions.

3.4 BEHAVIOURAL FEASIBILITY:

Behavioral feasibility deals with the runtime performance of the S/W the proposed system must score higher than the present in the behavioral study. The S/W should have end user in mind when the system is designed while designing s/w the programmer should be aware of the conditions user's Knowledge input, output, calculations etc.

The s/w contains only a minimum no. of bugs. Care should be also taken to avoid non-working means &t buttons.

CHAPTER- 4

Software Requirement & Specification

Software Required:

The project is implemented in Core Java as it provides the implementation of Socket and Server Socket classes that are used to connect distinct applications, hence the software's required in the creation and execution of the project are j2sdk1.7 or Eclipse .As we know JAVA is a platform independent language so this software runs with JRE environment on any desired platform i.e. Linux ,windows 9x, XP, or 2000 or any operating system.

Hardware Required:

As the project does not involve any database, its hardware requirements are minimal. Any System with Pentium P2 or above processor, 32MB RAM, 1GB Hard Disk, a LAN Card, and a CDROM is sufficient. Its network based software so computers connected with any kind of mode (wireless, LAN connected etc) will suit its requirements. . . . It can also be run on a single machine for its demo use.

Best suited in laboratory where we can run its server on any machine and many clients can use it simultaneously.

Software Analysis Report

About java: **Features JDK 1.7**

Platform Independent:

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

Simple:

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we can not say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory allocation and de-allocation system.

Object Oriented:

To be an Object Oriented language, any language must follow at least the four characteristics.

- Inheritance : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.
- Encapsulation: It is the mechanism of combining the information and providing the abstraction.
- Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
- Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfills the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java, it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

Robust:

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features makes the java language robust.

Distributed:

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

Portable:

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

Dynamic:

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

Secure:

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The byte code Verifier checks the classes after loading.

Performance:

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

Multithreaded:

Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer.
Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

Interpreted:

we all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code.
The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program.
The versatility of being **platform independent** makes Java to outshine from other languages. The source code to be written and distributed is platform independent.
Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

Architecture Neutral:

The term architectural neutral seems to be weird, but yes Java is an architectural neutral language as well. The growing popularity of networks makes developers think distributed. In the world of network it is essential that the applications must be able to migrate easily to different computer systems. Not only to computer systems but to a wide variety of hardware

architecture and operating system architectures as well. The Java compiler does this by generating byte code instructions, to be easily interpreted on any machine and to be easily translated into native machine code on the fly. The compiler generates an architecture-neutral object file format to enable a Java application to execute anywhere on the network and then the compiled code is executed on many processors, given the presence of the Java runtime system. Hence Java was designed to support applications on network. This feature of Java has thrived the programming language.

ABOUT : JDK:

The **Java Development Kit** (**JDK**) is a Sun Microsystems product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java SDK. On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007[1] and the source code was contributed to the OpenJDK.

The primary components of the JDK are a selection of programming tools, including:

- java – The loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, is no longer provided with Sun JDK.
- javac – The compiler, which converts source code into Java bytecode
- jar – The archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
- javadoc – The documentation generator, which automatically generates documentation from source code comments
- jdb – The debugger
- javap – The class file disassembler
- appletviewer – This tool can be used to run and debug Java applets without a web browser.
- javah – The C header and stub generator, used to write native methods
- extcheck – This utility can detect JAR-file conflicts.
- apt – The annotation processing tool
- jhat – (Experimental) Java heap analysis tool
- jstack – (Experimental) This utility prints Java stack traces of Java threads.
- jstat – (Experimental) Java Virtual Machine statistics monitoring tool
- jstatd – (Experimental) jstat daemon
- jinfo – (Experimental) This utility gets configuration information from a running Java process or crash dump.
- jmap – (Experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
- idlj – The IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
- policy tool – The policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
- VisualVM – visual tool integrating several command line JDK tools and lightweight performance and memory profiling capabilities

The JDK also comes with a complete Java Runtime Environment, usually called a *private* runtime. It consists of a Java Virtual Machine and all of the class libraries that will be present in the production environment, as well as additional libraries only useful to developers, such as the internationalization libraries and the IDL libraries.

Also included are a wide selection of example programs demonstrating the use of almost all portions of the Java API.

Technologies and Requriments

IDE:

My Eclipse

Front End:

JSP, JDBC, Javascript, AJAX

Programming Language:

JAVA

Back End:

Oracle 10g

CHAPTER- 5

System Design

5.1 INTRODUCTION:

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

5.2 NORMALIZATION:

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

FIRST NORMAL FORM:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1) Primary key is a not a composite primary key

2) No non key attributes are present
3) Every non key attribute is fully functionally dependent on full set of primary key.

THIRD NORMAL FORM:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

5.3 E – R DIAGRAMS:

- **The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.**
- **The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.**
- **The set of primary components that are identified by the ERD are**

  - ◆ **Data object**
  - ◆ **Relationships**
  - ◆ **Attributes**
  - ◆ **Various types of indicators.**

The primary purpose of the ERD is to represent data objects and their relationships.

5.4 DATA FLOW DIAGRAMS:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

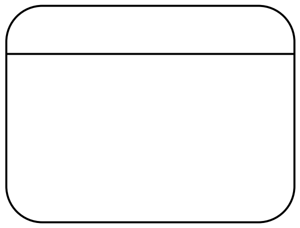
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

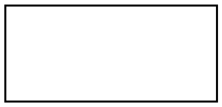
DFD SYMBOLS:

In the DFD, there are four symbols

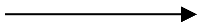
1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



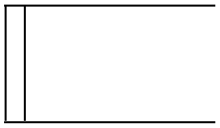
Process that transforms data flow.



Source or Destination of data



Data flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SAILENT FEATURES OF DFD'S

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In Current Physical DFD 26rocess label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects at the system are removed as mush as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with he user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S

PROCESS

1) No process can have only outputs.
2) No process can have only inputs. If an object has only inputs than it must be a sink.
3) A process has a verb phrase label.

DATA STORE

1) Data cannot move directly from one data store to another data store, a process must move data.
2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
3) A data store has a noun phrase label.

SOURCE OR SINK

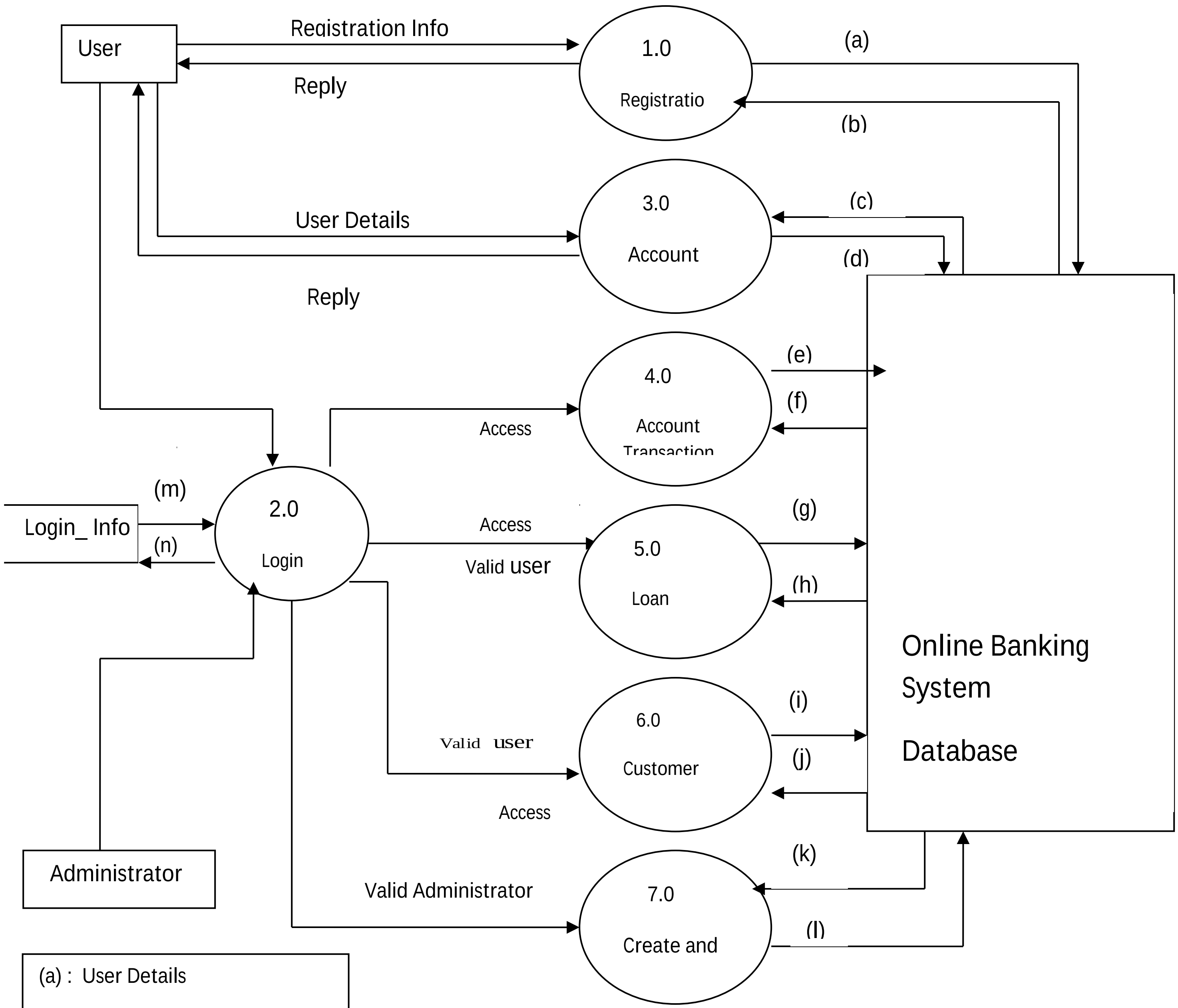
The origin and /or destination of data.

1) Data cannot move direly from a source to sink it must be moved by a process
2) A source and /or sink has a noun phrase land

DATA FLOW

1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3) A data flow cannot go directly back to the same process it leads. There must be atleast one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4) A Data flow to a data store means update (delete or change).
5) A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.



(a) : User Details

(b) : Response

(c) : Personal details

(d) : Reply

(e) : Account Transaction entry

(f) : Transaction Details

(g) : Loan Application

Fig. Level 1 DFD

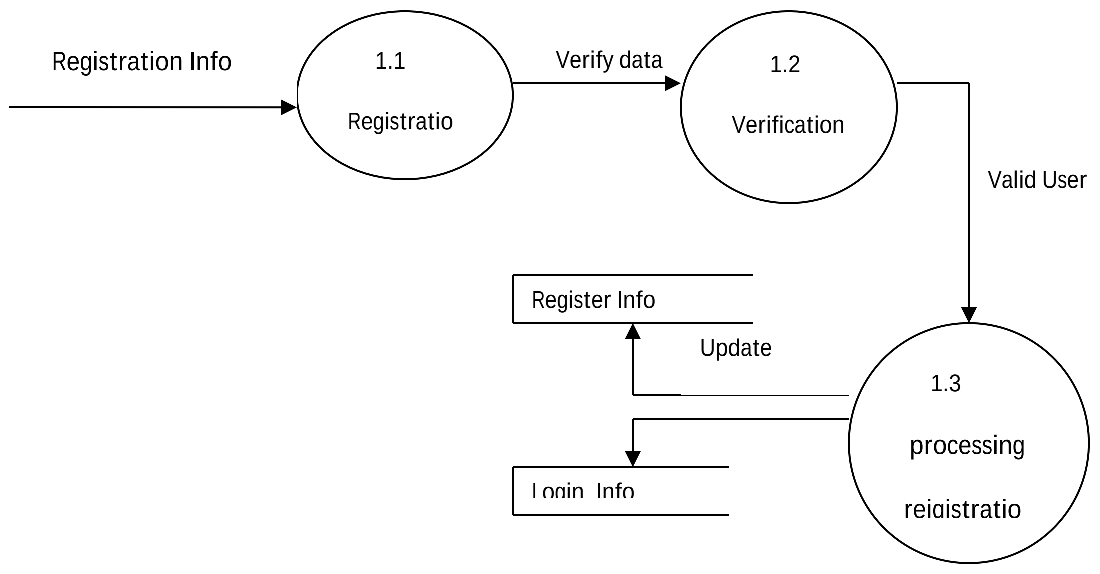


Fig . Level 2 DFD process-1

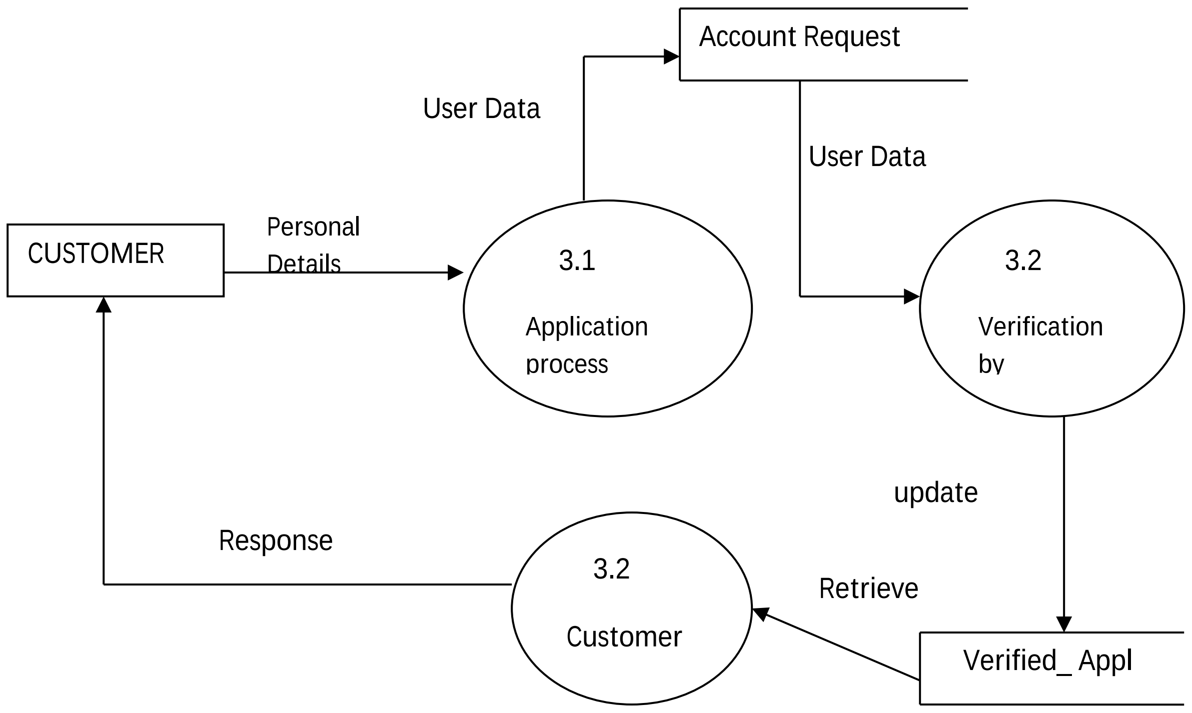


Fig . Level 2 DFD process-2

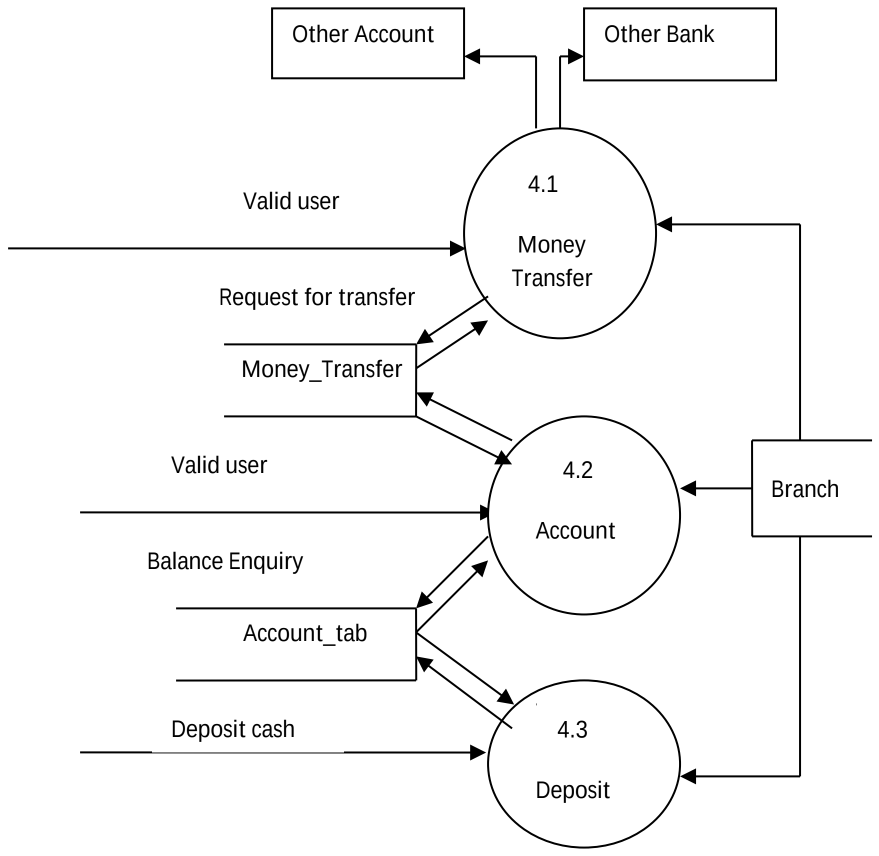


Fig . Level 2 DFD process-3

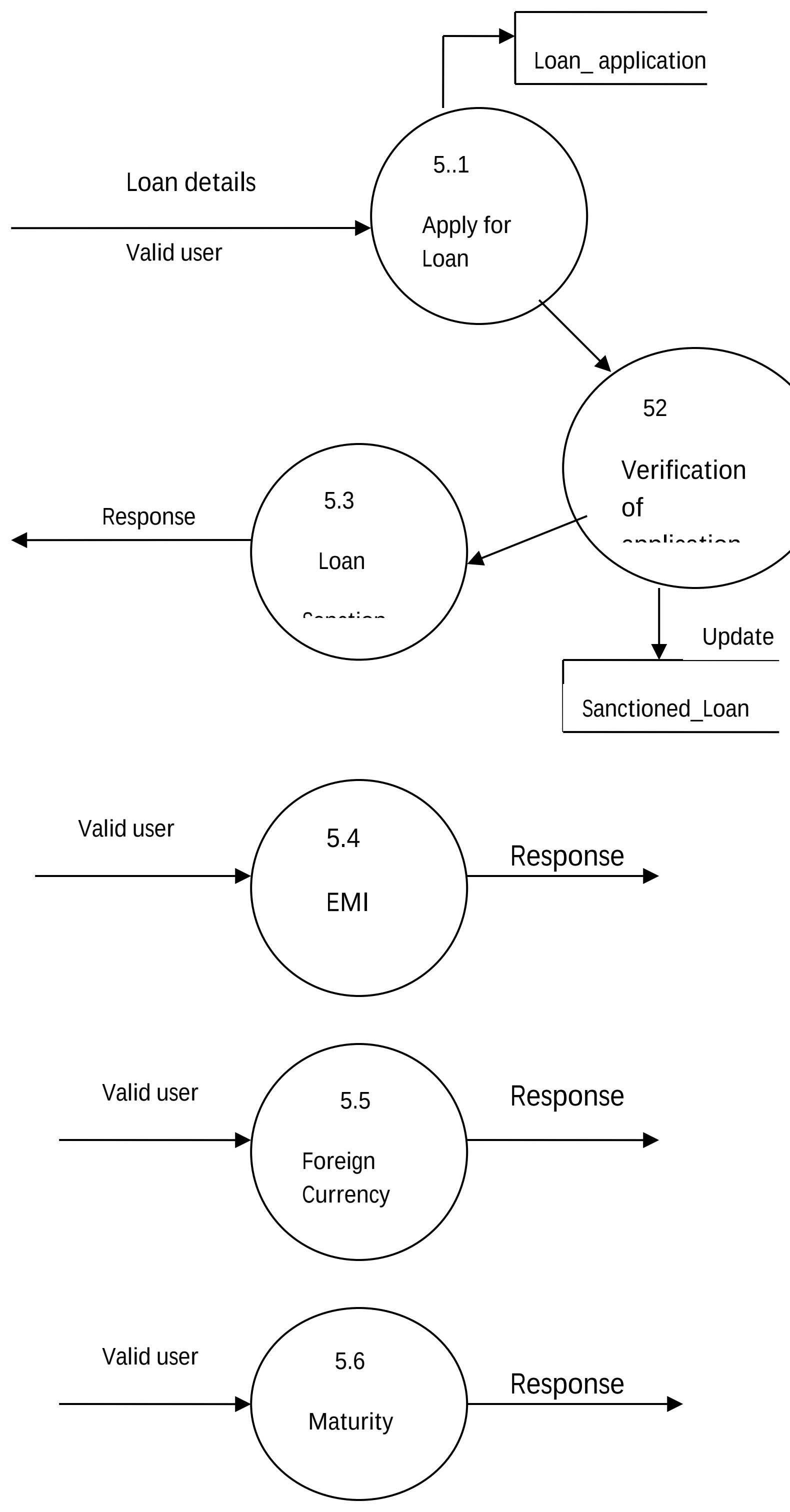


Fig . Level 2 DFD process-4

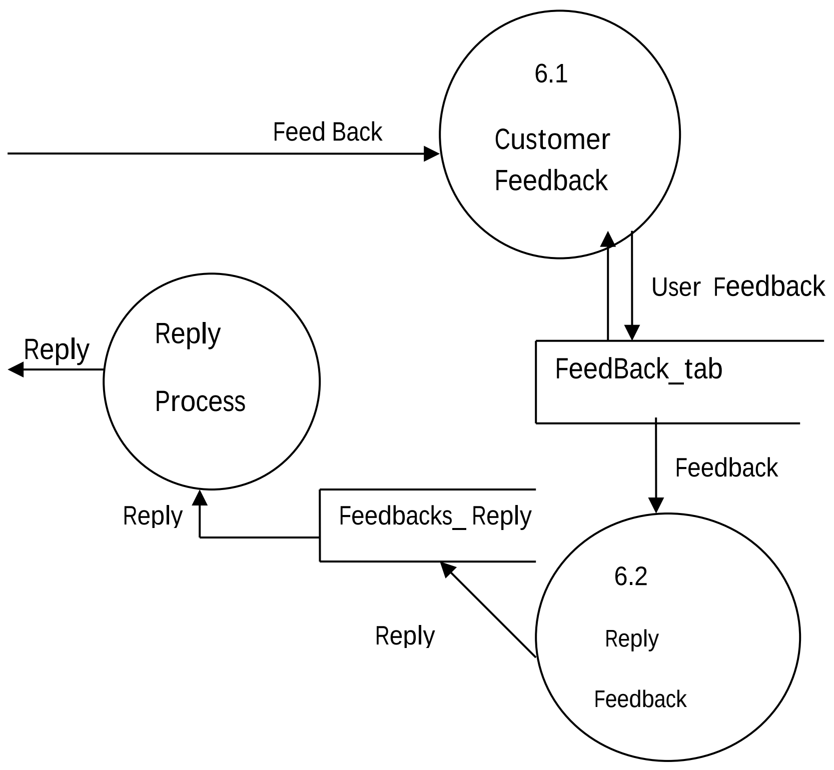


Fig . Level 2 DFD process-5

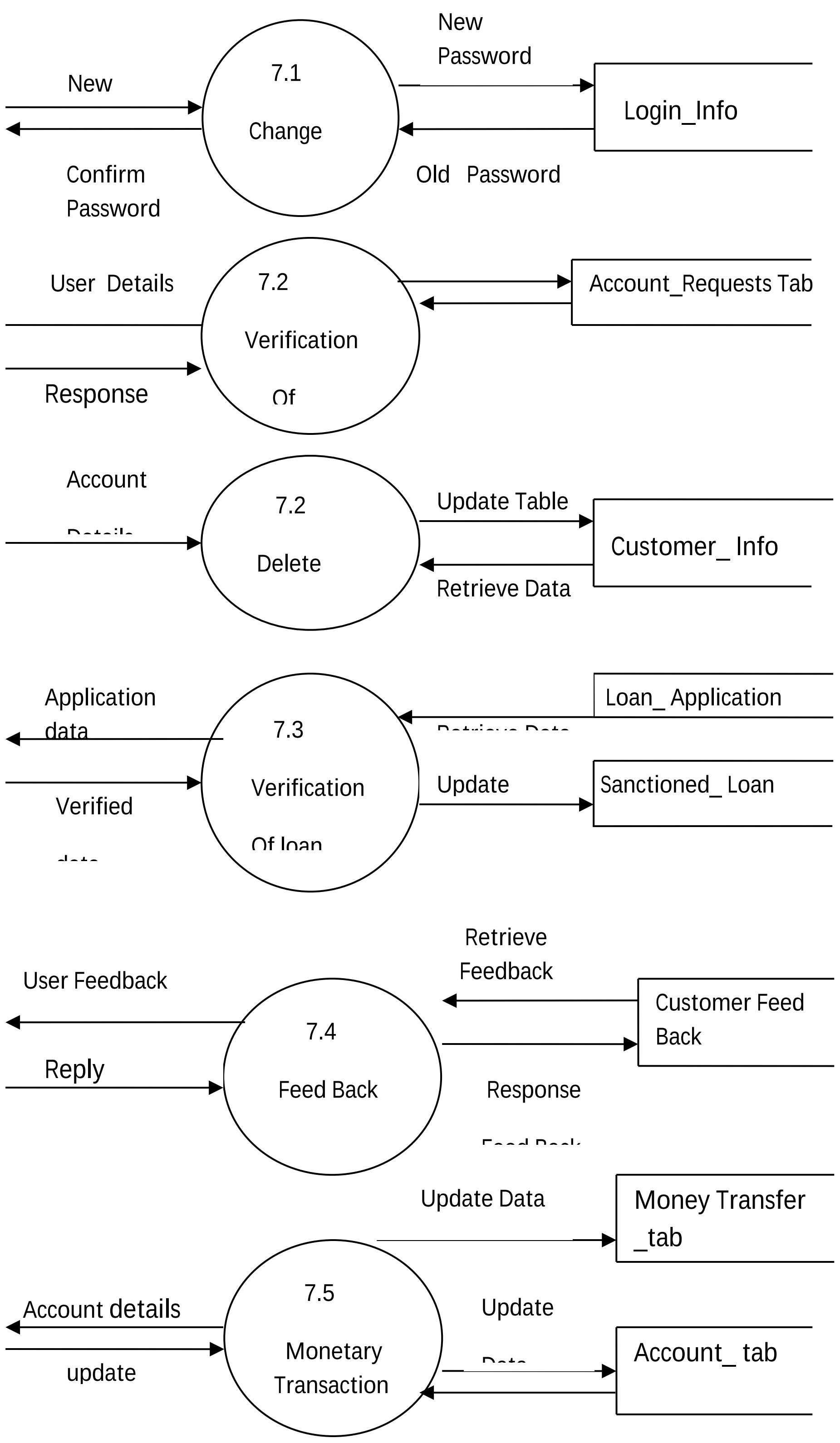


Fig . Level 2 DFD process-6

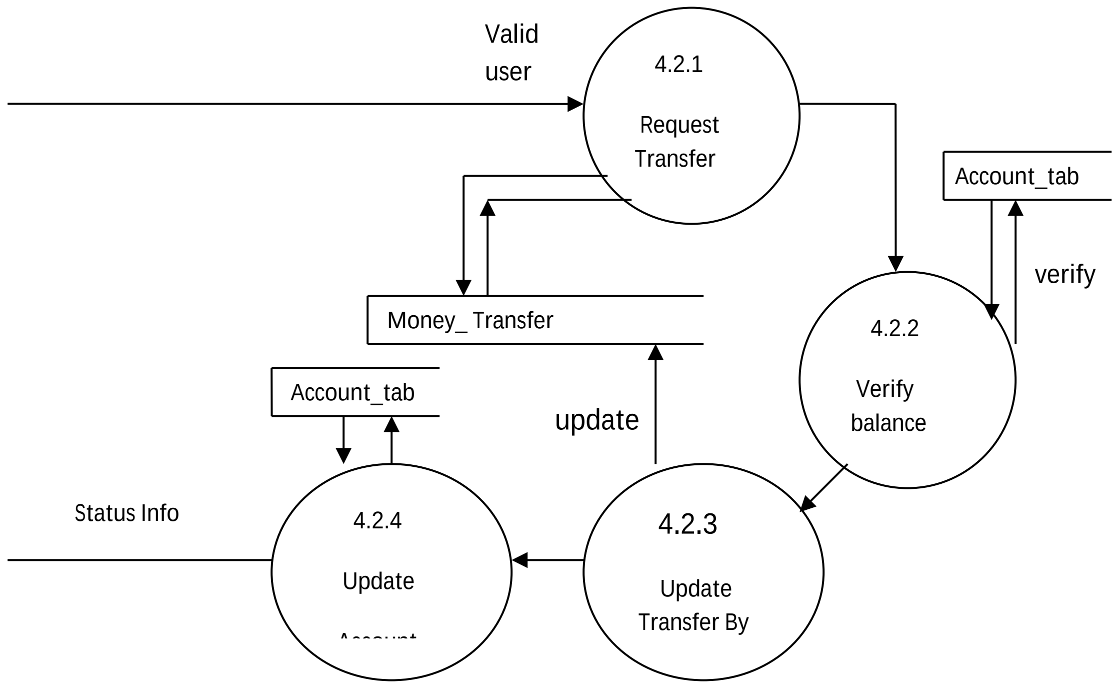


Fig . Level 3 DFD

5.5 DATABASE TABLE:

Column name	Data type	Nullable	Primary key
ACCOUNTINFO	Number	No	Yes
USERNAME	Varchar2	Yes	No
PASSWORD	Varchar2	Yes	No
AMOUNT	Varchar2	Yes	No
ADDRESS	Varchar2	Yes	No
PHONE	Varchar2	Yes	No

CHAPTER- 6

CODING

<u>CreateServlet.java</u>

```
package g;

import java.io.IOException;

import java.io.PrintWriter;

import java.rmi.Naming;


import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;




public class CreateServlet extends HttpServlet {


      public   void   doGet(HttpServletRequest   request,   HttpServletResponse
response)
                  throws ServletException, IOException {


            response.setContentType("text/html");

            PrintWriter out=response.getWriter();


            String username=request.getParameter("username");

            String password=request.getParameter("password");

            String  repassword=request.getParameter("repassword");


            String amoun=request.getParameter("amount");
            double amount=Double.parseDouble(amoun);


            String adderess=request.getParameter("adderess");


            String ph=request.getParameter("phone");

            double phone=Double.parseDouble(ph);

            //double mname=Double.parseDouble(num);

            //String country=request.getParameter("country");


          int   status=RegisterUser.register(username,   password,   repassword,
amount, adderess,phone);




            if(status>0){

                  out.print("WELCOME! YOUR ACCOUNT HAS OPENED");

                  RequestDispatcher
rd=request.getRequestDispatcher("index.jsp");

                  rd.include(request, response);

            }

            else{

                  out.print("Sorry,Registration failed. please try later");

                  RequestDispatcher
rd=request.getRequestDispatcher("MyHtml.html");

                  rd.include(request, response);

            }


      out.close();

      }}
```

DBIntializer.java

```
package g;


public interface DBIntializer {

String DRIVER="oracle.jdbc.driver.OracleDriver";

String CON_STRING="jdbc:oracle:thin:@localhost:1521:xe";

String USERNAME="system";

String PASSWORD="oracle";

}
```

Details.java

```
package g;

// THIS PROGRAM GBANK IS THE INTERFACE FOR THE MAIN PROGRAM



import java.rmi.*;


public interface Details extends Remote

{

    public int open(String username,String password,double amount,String adderess,double phone) throws RemoteException;

    public String withdraw(int acno,String uname,String pwd,int amt) throws RemoteException;

    public String deposit(int acno,String uname,String pwd,int amt) throws RemoteException;

    public String transfer(int acno,String uname,String pwd,int tacno,int amt) throws RemoteException;

    public String close(int acno,String uname,String pass)  throws RemoteException;
    public String balance(int acno,String uname,String pass) throws RemoteException;

}
```

GetCon.java

```
package g;

import java.sql.*;

public class GetCon {

private GetCon(){}


public static Connection con;

static{

      try {

            Class.forName(DBIntializer.DRIVER);


      con=DriverManager.getConnection(DBIntializer.CON_STRING,DBIntializer.USE
RNAME,DBIntializer.PASSWORD);

      } catch (ClassNotFoundException e) {


            e.printStackTrace();

      } catch (SQLException e) {


            System.out.println("Exception in GetCon");

      }


}

public static Connection getCon(){

      return con;
}




public static int getPrimaryKey(){

      int nextvalue=0;

      Connection con=GetCon.getCon();

      PreparedStatement ps2;

      try {


      ps2=con.prepareStatement("select    javatpointnewaccount.nextval    from
dual");


      ResultSet rs=ps2.executeQuery();

      rs.next();

      nextvalue=rs.getInt(1);




} catch (SQLException e) {


            e.printStackTrace();

      }

return nextvalue;


}

}
```

MyListener.java

```
package g;

import javax.servlet.*;

import java.sql.*;


public class MyListener implements ServletContextListener{


      public void contextInitialized(ServletContextEvent arg0) {

            int status=0;

            Connection con=null;


      try{

            con=GetCon.getCon();

            PreparedStatement    ps1=con.prepareStatement("Select    *    from
NEWACCOUNT");



      try{

            status=ps1.executeUpdate();

            }


      catch(Exception e)

            {e.printStackTrace();

             status=2;

             System.out.println("my staus is1111111"+status);

             }


       if(status==0)
```

```
            {System.out.println("your table name already exist"+status);}



       else if(status==2)

        {System.out.println("else if part table does not exist new table has
created"+status);

            PreparedStatement     ps3=con.prepareStatement("CREATE     SEQUENCE
javatpointnewaccount MINVALUE 1 MAXVALUE 999999999999 INCREMENT BY 1 START WITH
1 NOCACHE  NOORDER  NOCYCLE");

            ps3.executeUpdate();


            PreparedStatement       ps=con.prepareStatement("CREATE       TABLE
NEWACCOUNT(ACCOUNTNO          NUMBER,USERNAME          VARCHAR2(4000),PASSWORD
VARCHAR2(4000),REPASSWORD        VARCHAR2(4000),AMOUNT         NUMBER,ADDERESS
VARCHAR2(4000),PHONE NUMBER,PRIMARY KEY (ACCOUNTNO))");

            ps.executeUpdate();







            }




            else{System.out.println("else part "+status);

            }}

          catch(Exception e){e.printStackTrace();}

          }

          public void contextDestroyed(ServletContextEvent arg0) {

            System.out.println("project undeployed");
```

```
      }
}
```

RegisterUser.java

```
package g;
import java.sql.*;


import g.GetCon;
public class RegisterUser {
static int status=0;
//int accountno=1;
public static int register(String username,String password,String repassword,double amount,String adderess,double phone){
      //public static int register(String email,String password,String gender,String country,String name){


      Connection con=GetCon.getCon();
      PreparedStatement ps;
      try {
            ps = con.prepareStatement("Insert into NEWACCOUNT values(?,?,?,?,?,?,?)");
            int   nextvalue1=GetCon.getPrimaryKey();
            ps.setInt(1,nextvalue1);
            ps.setString(2,username);
            ps.setString(3,password);
            ps.setString(4,repassword);
            ps.setDouble(5,amount);
            ps.setString(6,adderess);
            ps.setDouble(7,phone);
            status=ps.executeUpdate();

      } catch (SQLException e) {

            e.printStackTrace();
      }
      return status;

}
}
```

verifyLogin1.java

```
package g;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class verifyLogin1 {

public static boolean checkLogin(int accountno,String username,String password){
      boolean status=false;
      Connection con=GetCon.getCon();
      try {
            //PreparedStatement ps=con.prepareStatement("Select * from MAILCASTINGUSER where EMAILADD = ? and PASSWORD =?");
            PreparedStatement ps=con.prepareStatement("Select * from NEWACCOUNT where accountno=? and username = ? and password =?");
            ps.setInt(1,accountno);
            ps.setString(2,username);
            ps.setString(3,password);

            ResultSet rs=ps.executeQuery();
            status=rs.next();

      } catch (SQLException e) {
            e.printStackTrace();
      }
      return status;
}
}
```

WEB.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
      xmlns="http://java.sun.com/xml/ns/javaee"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
      http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">


       <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
```

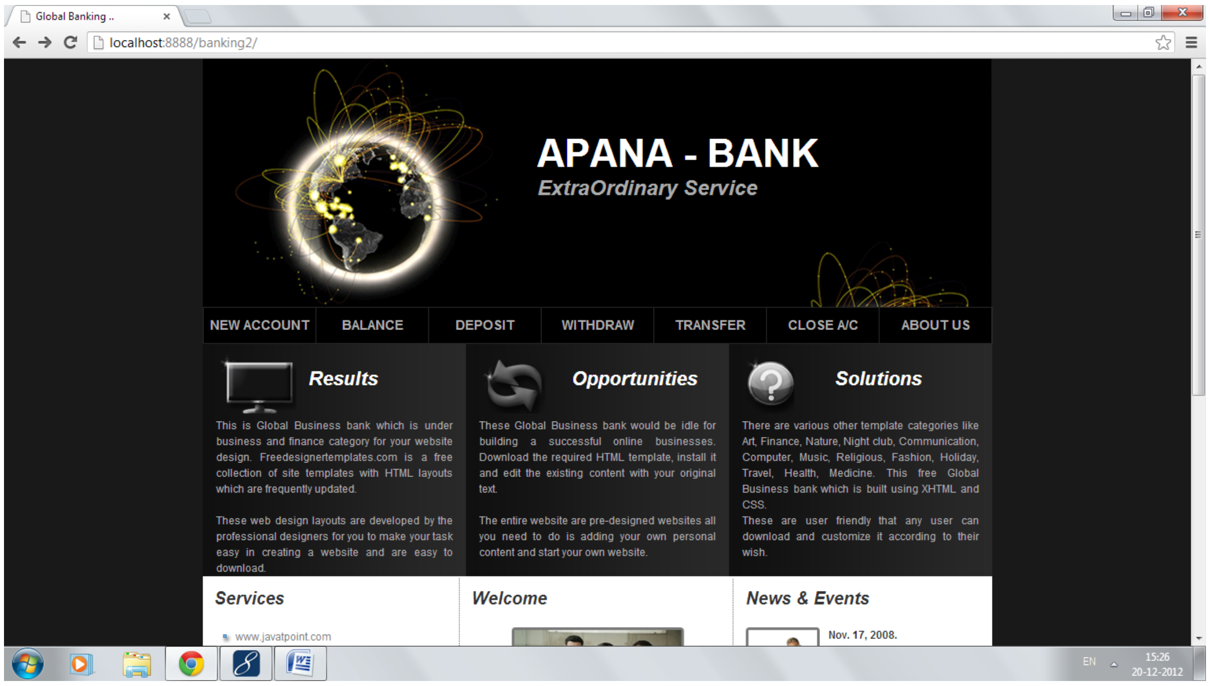
```
   <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>CreateServlet</servlet-name>
    <servlet-class>g.CreateServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>CreateServlet</servlet-name>
    <url-pattern>/CreateServlet</url-pattern>
  </servlet-mapping>

   <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>registereduserServlet</servlet-name>
    <servlet-class>g.registereduserServlet</servlet-class>
  </servlet>
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>verifyLogin</servlet-name>
    <servlet-class>verifyLogin</servlet-class>
  </servlet>
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>VerifyLogin1</servlet-name>
    <servlet-class>g.VerifyLogin1</servlet-class>
  </servlet>
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>verifyLogin1</servlet-name>
    <servlet-class>g.verifyLogin1</servlet-class>
  </servlet>
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>GetCon</servlet-name>
    <servlet-class>GetCon</servlet-class>
  </servlet>




  <servlet-mapping>
    <servlet-name>registereduserServlet</servlet-name>
    <url-pattern>/go</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>verifyLogin1</servlet-name>
    <url-pattern>/servlet/verifyLogin1</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
```

```
    <servlet-name>VerifyLogin1</servlet-name>
    <url-pattern>/servlet/VerifyLogin1</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>verifyLogin1</servlet-name>
    <url-pattern>/servlet/verifyLogin1</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>GetCon</servlet-name>
    <url-pattern>/servlet/GetCon</url-pattern>
  </servlet-mapping>



   <listener>
  <listener-class>g.MyListener</listener-class>
  </listener>

</web-app>
```

CHAPTER- 7

OUTPUT SCREENS

Welcome Page



This is Global Business bank which is under business and finance category for your website design. Freedesignertemplates.com is a free collection of site templates with HTML layouts which are frequently updated.

These web design layouts are developed by the professional designers for you to make your task easy in creating a website and are easy to download.

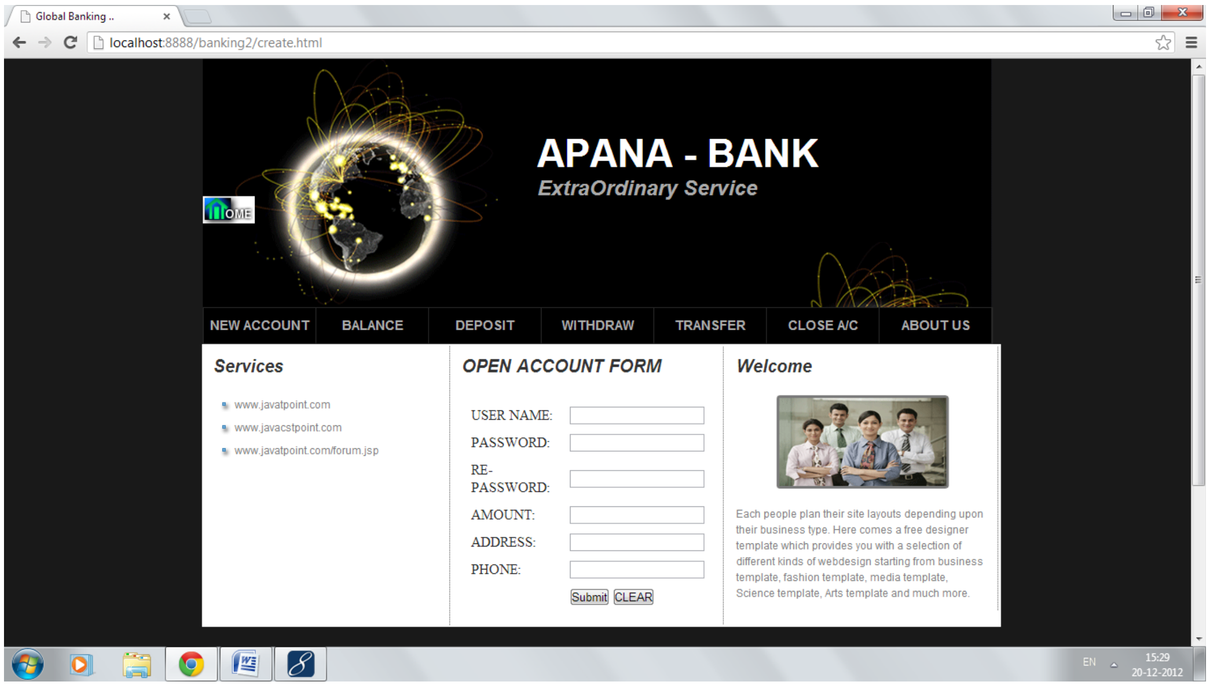
These Global Business bank would be idle for building a successful online businesses. Download the required HTML template, install it and edit the existing content with your original text.

The entire website are pre-designed websites all you need to do is adding your own personal content and start your own website.

There are various other template categories like Art, Finance, Nature, Night club, Communication, Computer, Music, Religious, Fashion, Holiday, Travel, Health, Medicine. This free Global Business bank which is built using XHTML and CSS.
These are user friendly that any user can download and customize it according to their wish.

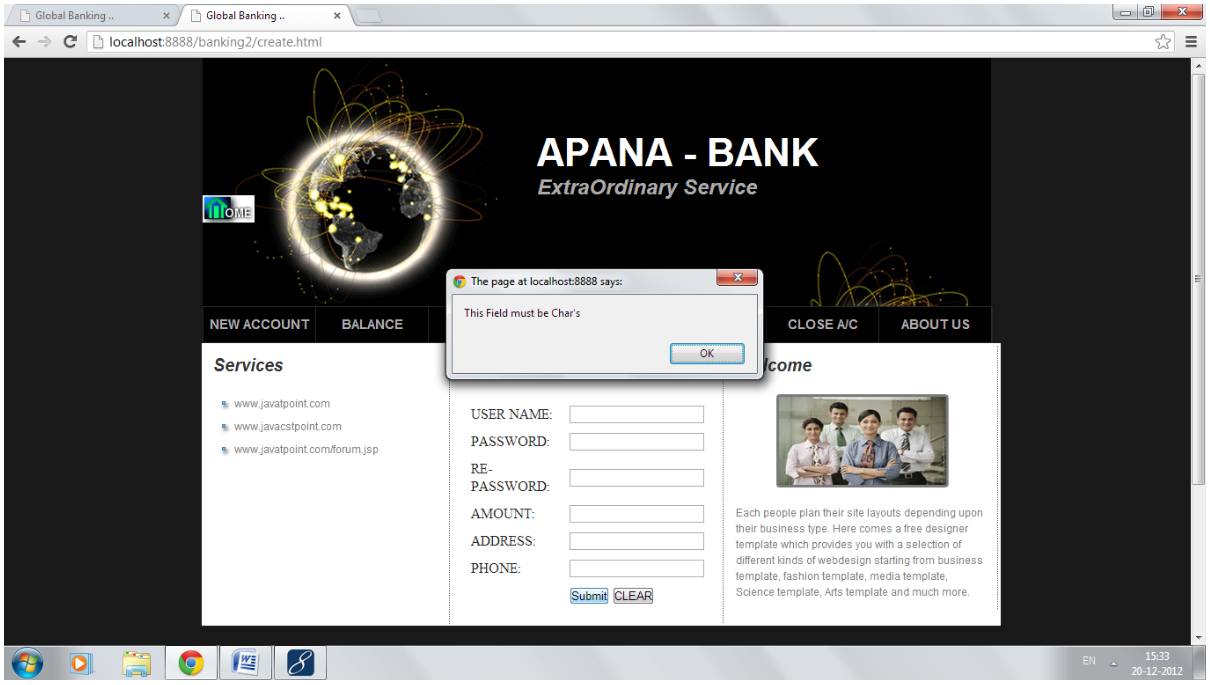
<u>Steps:</u>

1. Create new account by clicking on New Account link:



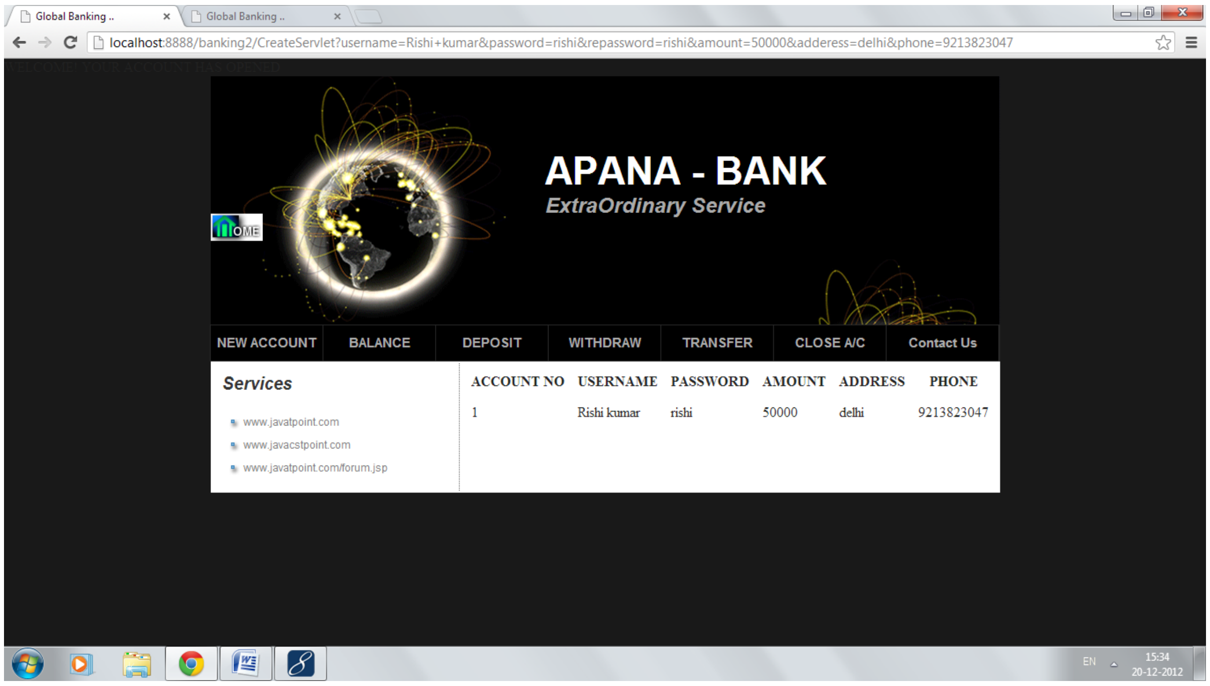
Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

Leaving any field blank:



This Field must be Char's

On submitting the form with correct information:



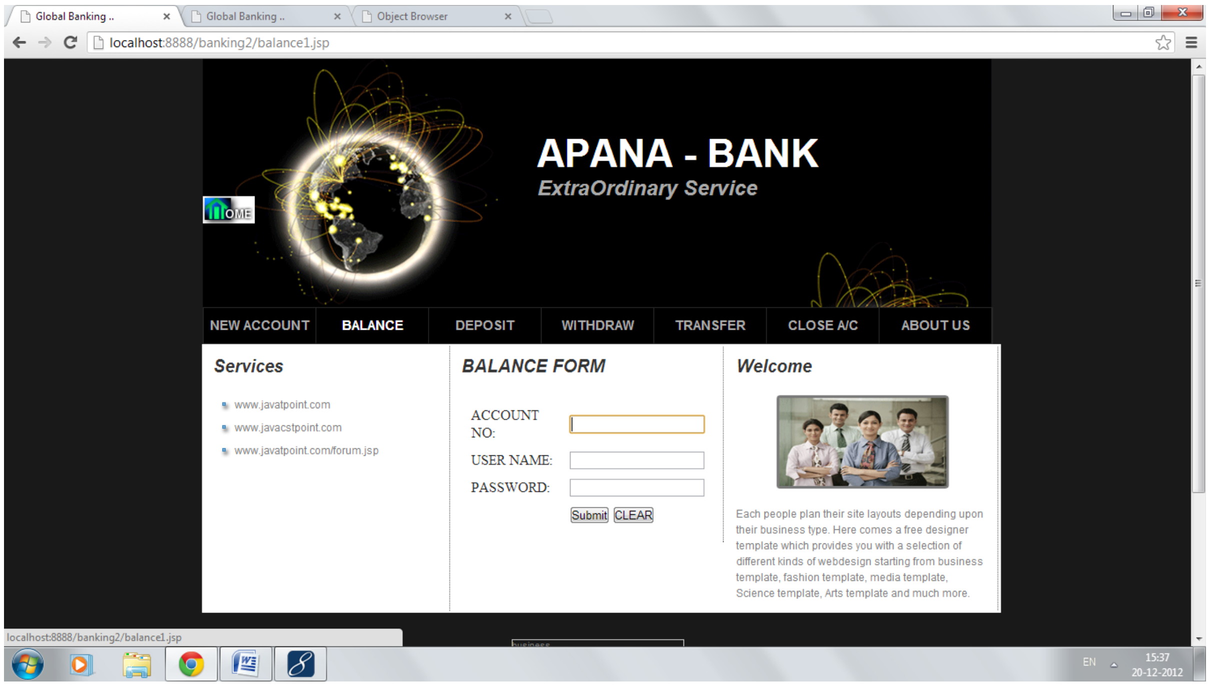
localhost:8888/banking2/CreateServlet?username=Rishi+kumar&password=rishi&repassword=rishi&amount=50000&adderess=delhi&phone=9213823047

WELCOME! YOUR ACCOUNT HAS OPENED

ACCOUNT NO	USERNAME	PASSWORD	AMOUNT	ADDRESS	PHONE
1	Rishi kumar	rishi	50000	delhi	9213823047

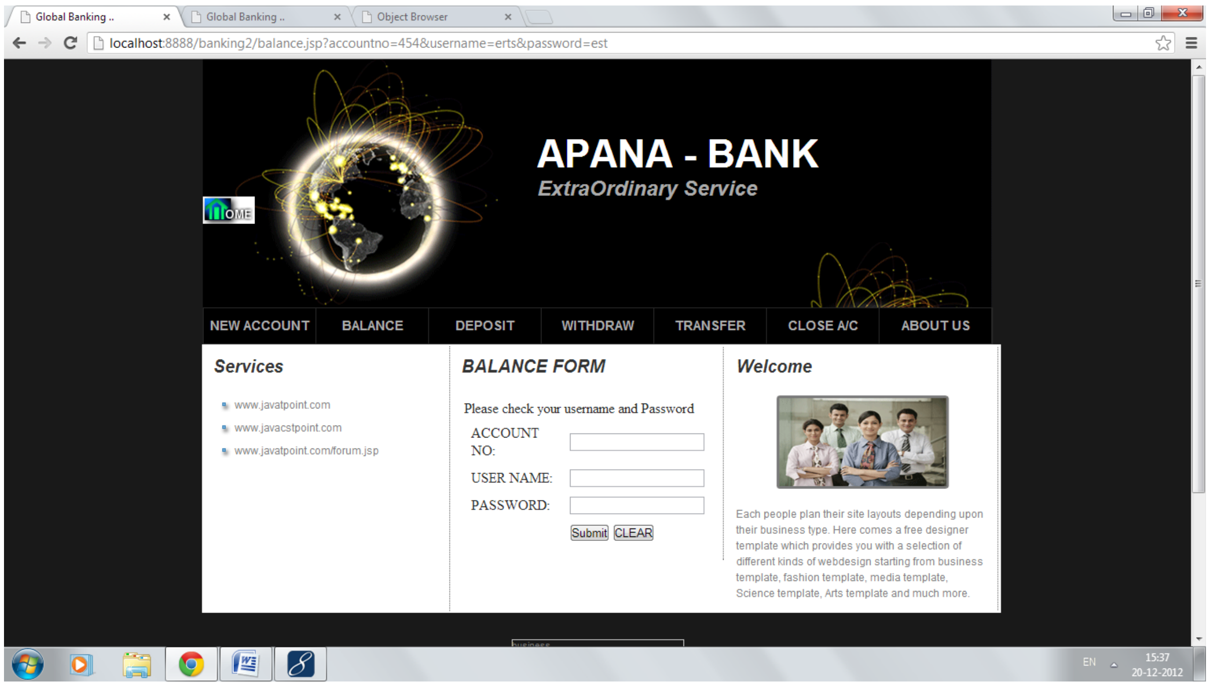
If you have already an account you can check balance, deposit, with-draw, money transfer and can close your account by clicking on respective link.

2.Account holder can check balance:



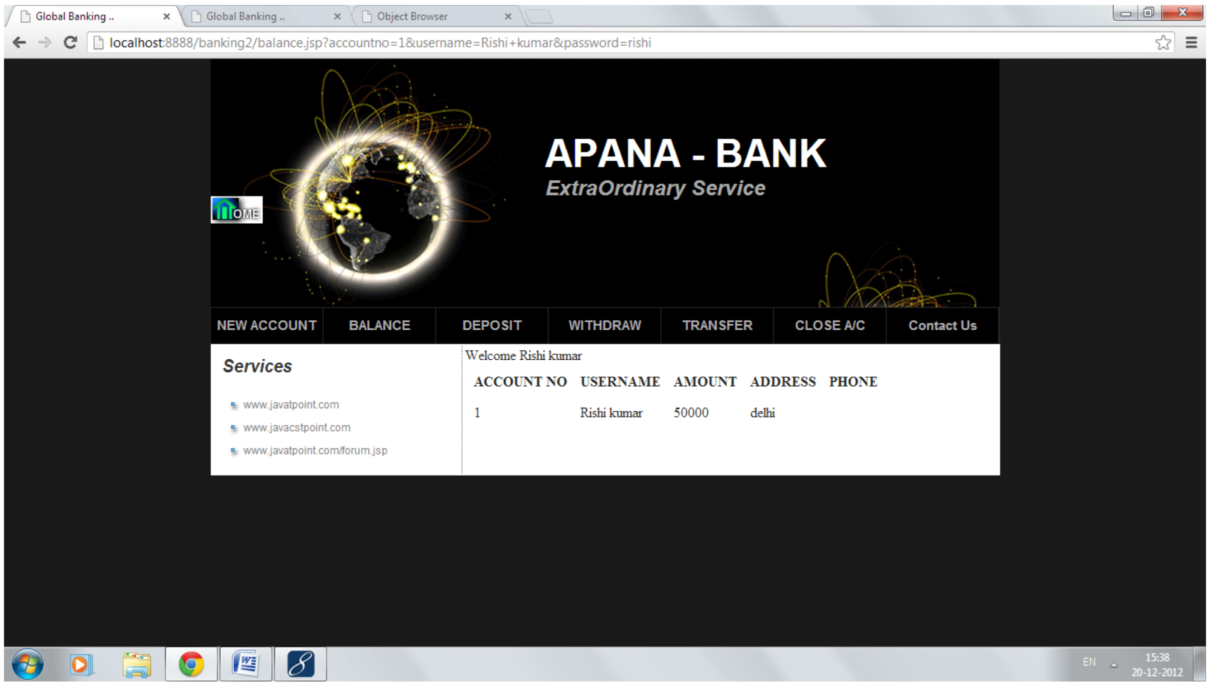
Global Banking ..
Global Banking ..
Object Browser
localhost:8888/banking2/balance1.jsp
APANA - BANK
ExtraOrdinary Service
NEW ACCOUNT
BALANCE
DEPOSIT
WITHDRAW
TRANSFER
CLOSE A/C
ABOUT US
Services
www.javatpoint.com
www.javacstpoint.com
www.javatpoint.com/forum.jsp
BALANCE FORM
ACCOUNT NO:
USER NAME:
PASSWORD:
Submit
CLEAR
Welcome
Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.
localhost:8888/banking2/balance1.jsp
15:37
20-12-2012

On filling Incorrect Detail:



Global Banking ..
Global Banking ..
Object Browser
localhost:8888/banking2/balance.jsp?accountno=454&username=erts&password=est
APANA - BANK
ExtraOrdinary Service
NEW ACCOUNT
BALANCE
DEPOSIT
WITHDRAW
TRANSFER
CLOSE A/C
ABOUT US
Services
www.javatpoint.com
www.javacstpoint.com
www.javatpoint.com/forum.jsp
BALANCE FORM
Please check your username and Password
ACCOUNT NO:
USER NAME:
PASSWORD:
Submit
CLEAR
Welcome
Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.
15:37
20-12-2012

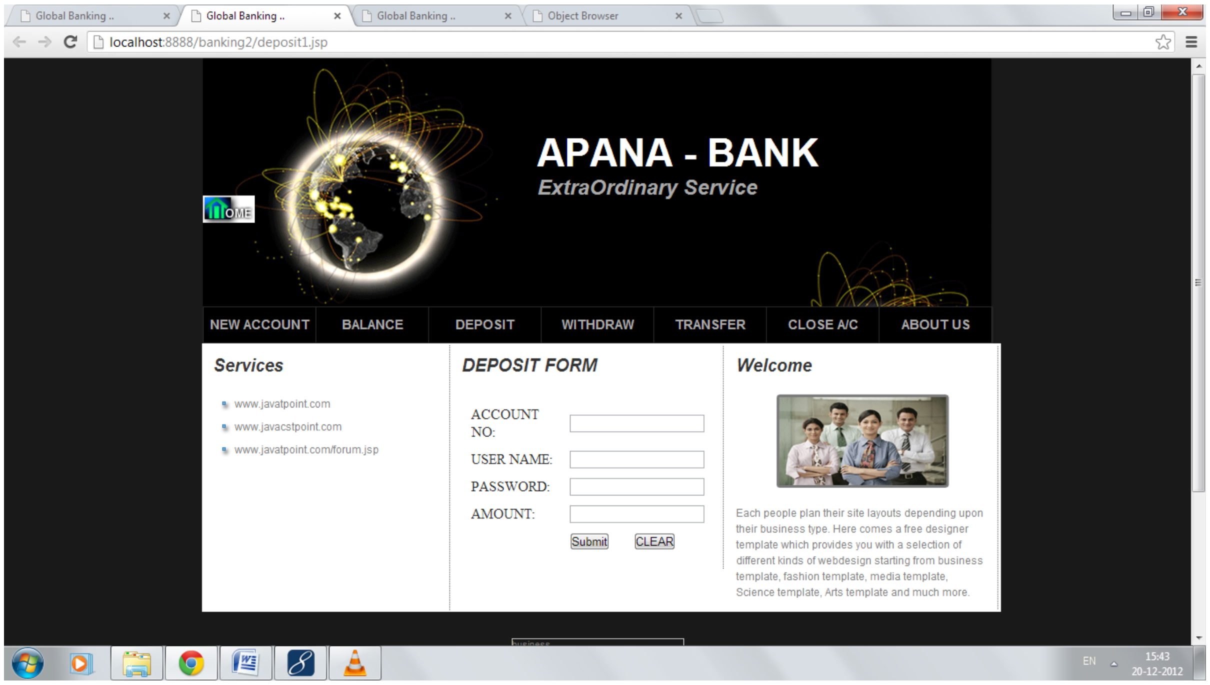
On filling Correct Detail:


localhost:8888/banking2/balance.jsp?accountno=1&username=Rishi+kumar&password=rishi

Welcome Rishi kumar

ACCOUNT NO	USERNAME	AMOUNT	ADDRESS	PHONE
1	Rishi kumar	50000	delhi	

3.Can deposit money:

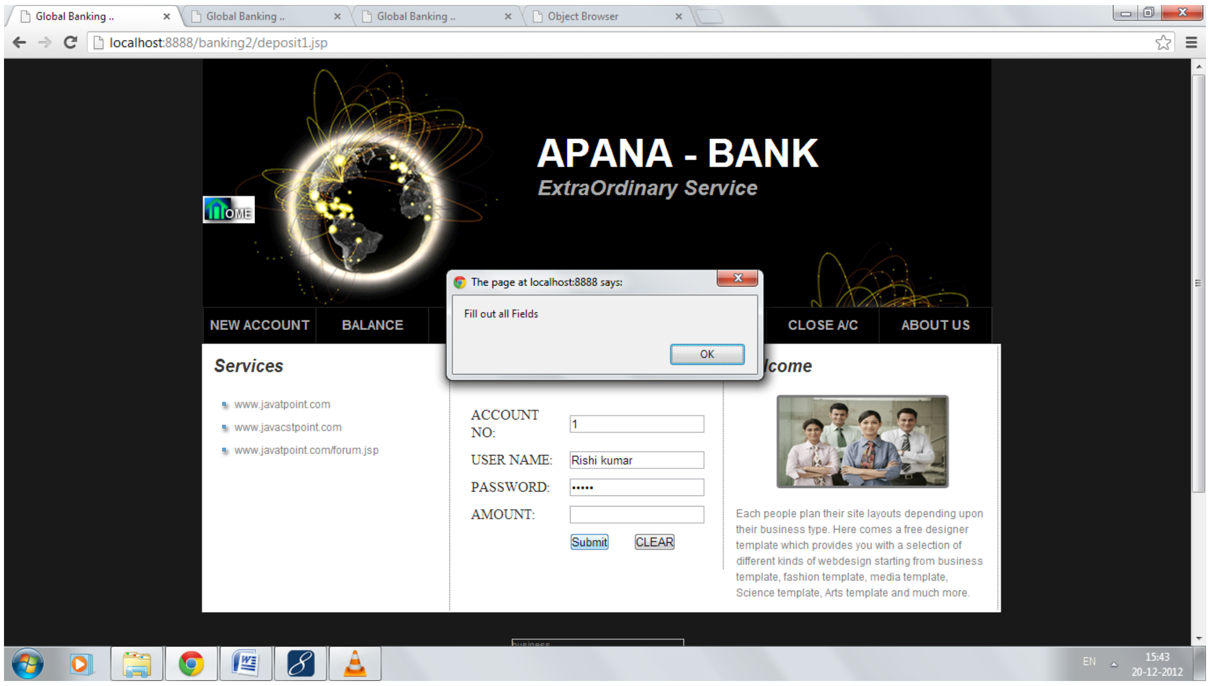

localhost:8888/banking2/deposit1.jsp

DEPOSIT FORM

ACCOUNT NO:	
USER NAME:	
PASSWORD:	
AMOUNT:	

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling Incorrect Detail:

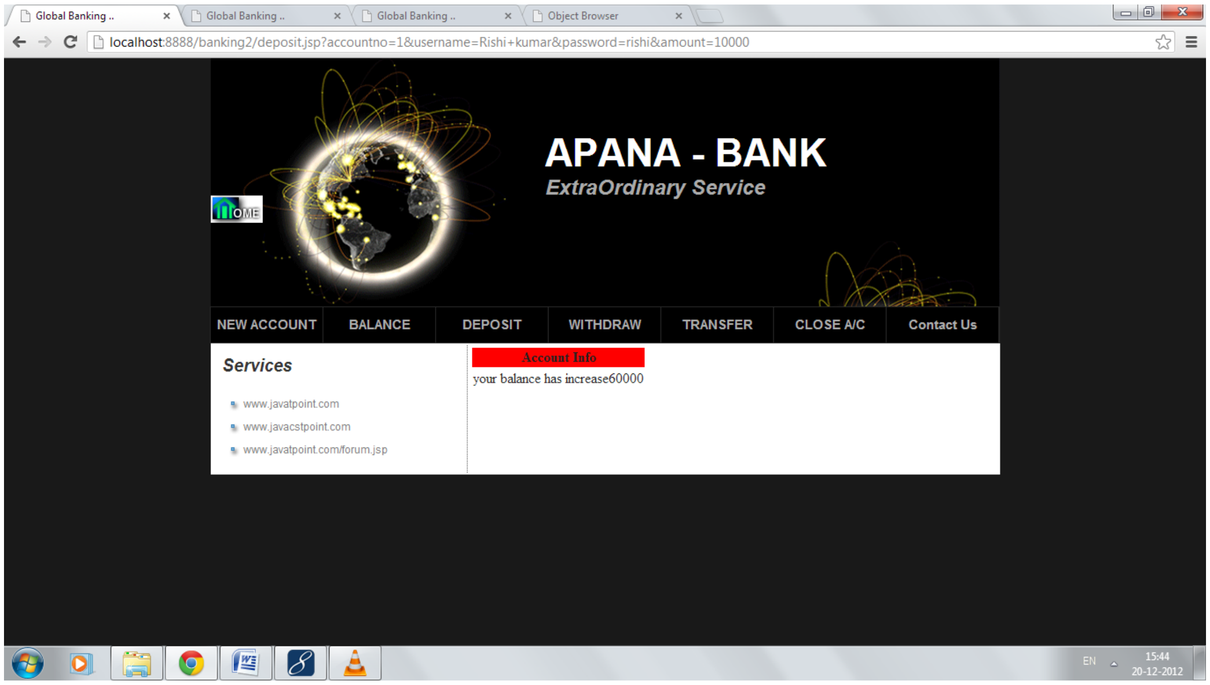

localhost:8888/banking2/deposit1.jsp

The page at localhost:8888 says:
Fill out all Fields

ACCOUNT NO: 1
USER NAME: Rishi kumar
PASSWORD: •••••
AMOUNT:

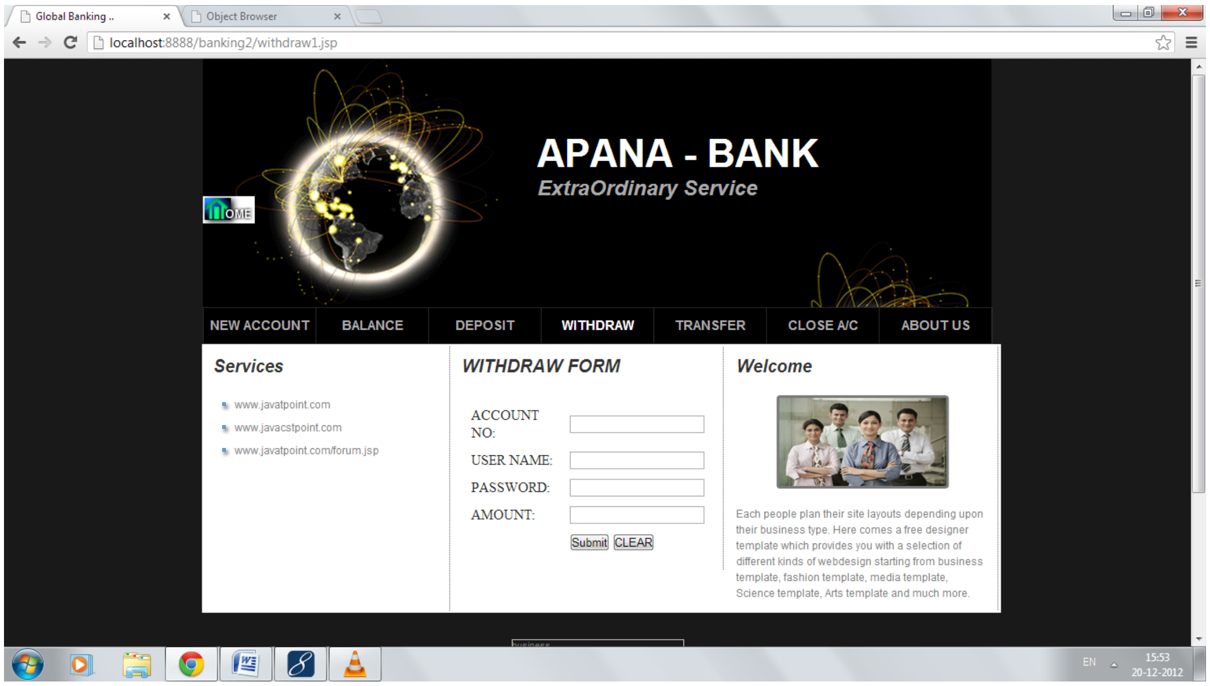
Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling correct Detail:


localhost:8888/banking2/deposit.jsp?accountno=1&username=Rishi+kumar&password=rishi&amount=10000

Account Info
your balance has increase60000

4.Can withdraw money:



localhost:8888/banking2/withdraw1.jsp

APANA - BANK

ExtraOrdinary Service

Services

- www.javatpoint.com
- www.javacstpoint.com
- www.javatpoint.com/forum.jsp

WITHDRAW FORM

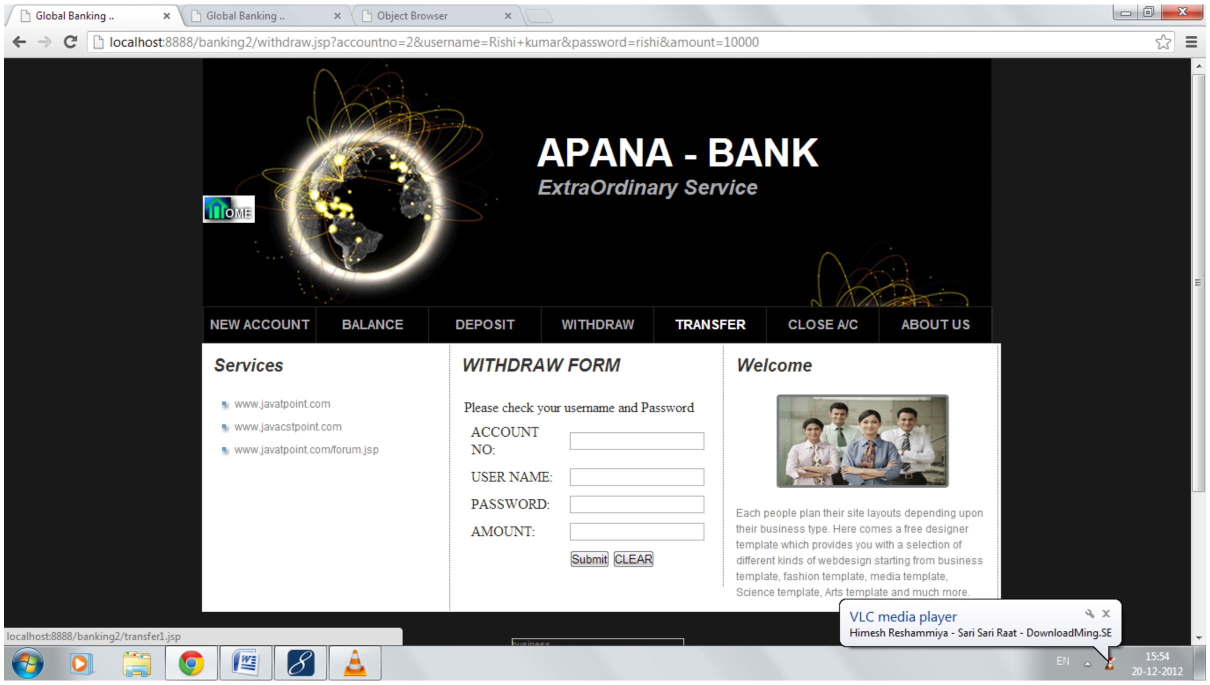
ACCOUNT NO:
USER NAME:
PASSWORD:
AMOUNT:

Welcome

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

EN 15:53 20-12-2012

On filling Incorrect Detail:



localhost:8888/banking2/withdraw.jsp?accountno=2&username=Rishi+kumar&password=rishi&amount=10000

APANA - BANK

ExtraOrdinary Service

Services

- www.javatpoint.com
- www.javacstpoint.com
- www.javatpoint.com/forum.jsp

WITHDRAW FORM

Please check your username and Password

ACCOUNT NO:
USER NAME:
PASSWORD:
AMOUNT:

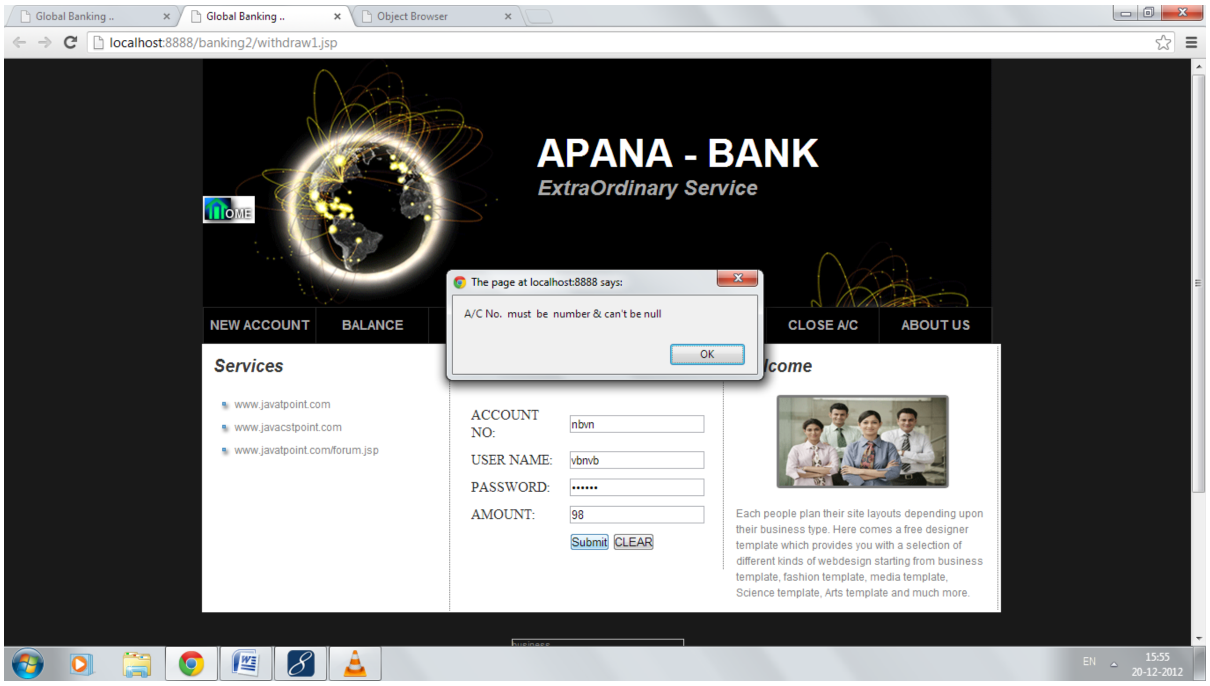
Welcome

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

localhost:8888/banking2/transfer1.jsp

VLC media player
Himesh Reshammiya - Sari Sari Raat - DownloadMing.SE

EN 15:54 20-12-2012



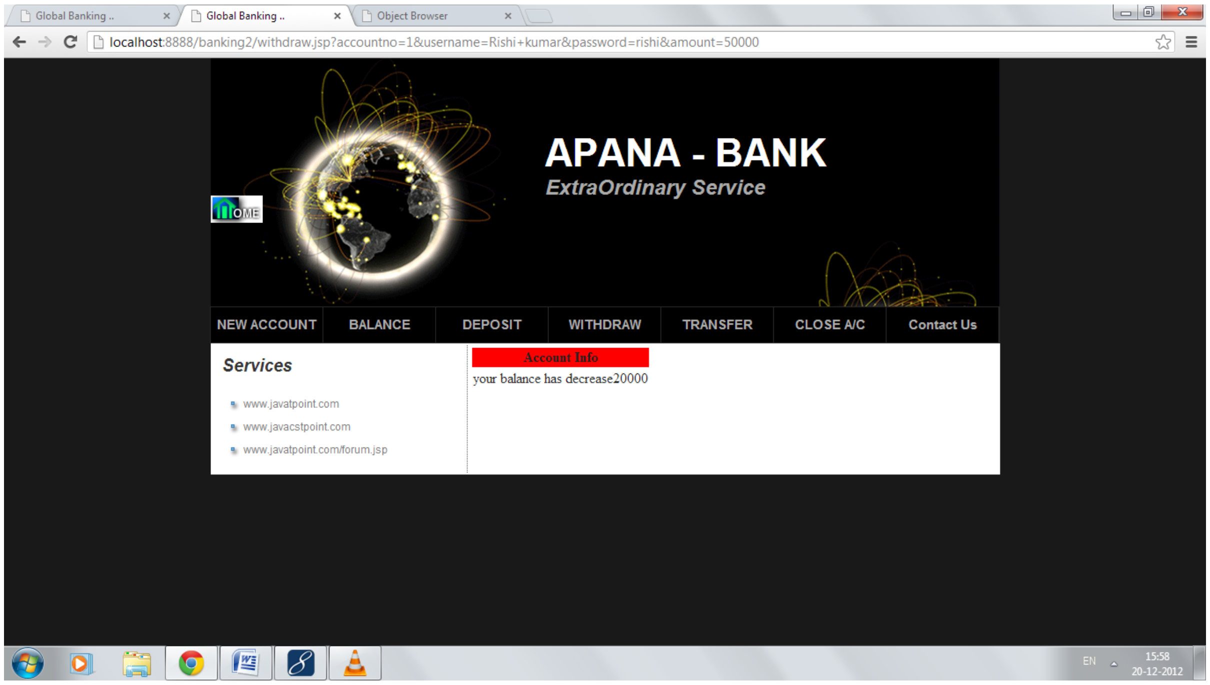
The page at localhost:8888 says:

A/C No. must be number & can't be null

ACCOUNT NO:	nbvn
USER NAME:	vbnvb
PASSWORD:	••••••
AMOUNT:	98

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling Correct Detail:

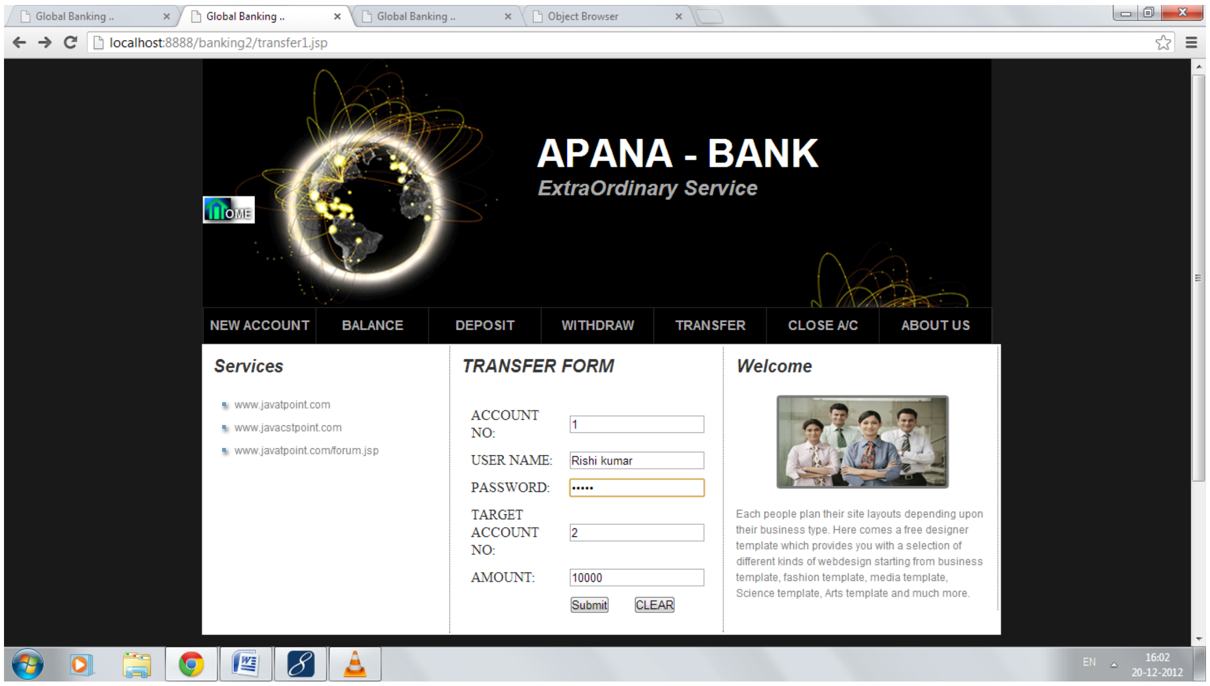


localhost:8888/banking2/withdraw.jsp?accountno=1&username=Rishi+kumar&password=rishi&amount=50000

Account Info

your balance has decrease20000

5.Can transfer Money:

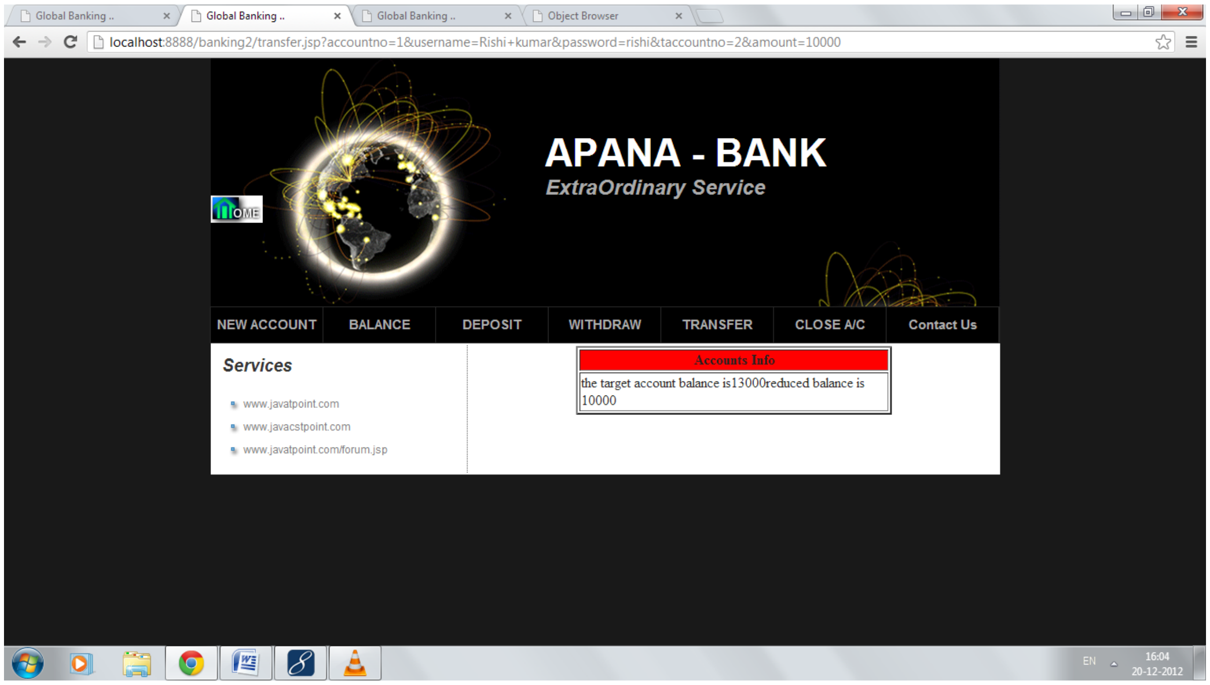


TRANSFER FORM

ACCOUNT NO:	1
USER NAME:	Rishi kumar
PASSWORD:	•••••
TARGET ACCOUNT NO:	2
AMOUNT:	10000

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling Correct Detail:

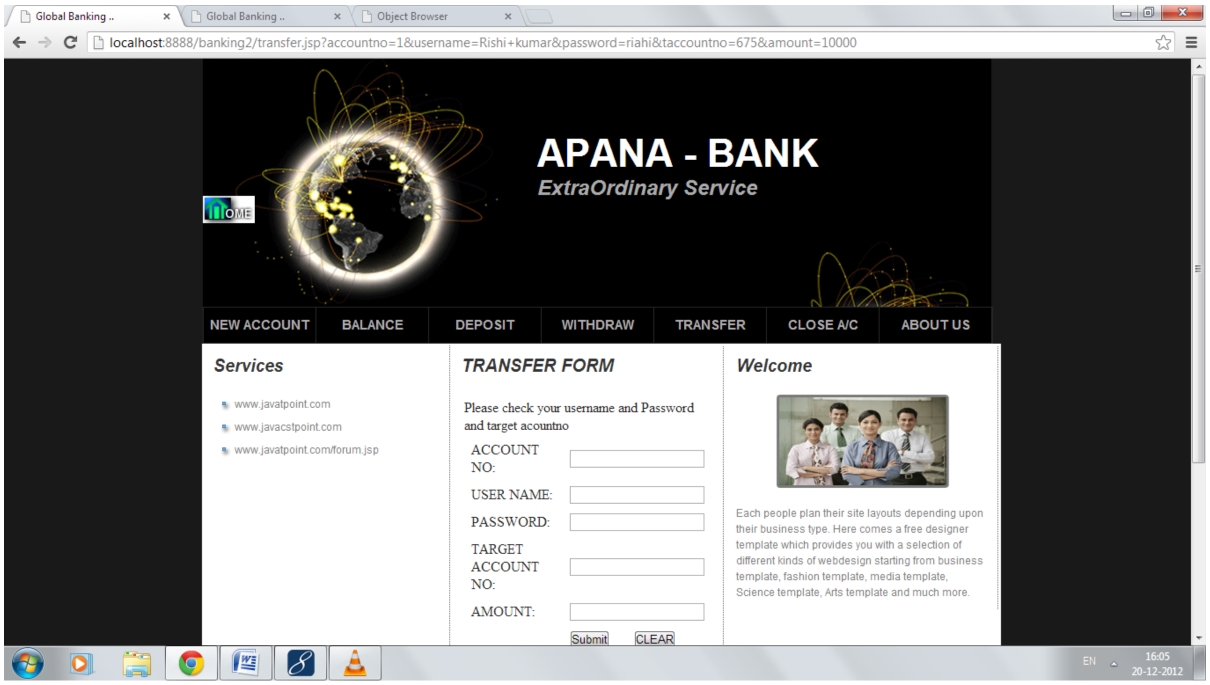


localhost:8888/banking2/transfer.jsp?accountno=1&username=Rishi+kumar&password=rishi&taccountno=2&amount=10000

Accounts Info

the target account balance is13000reduced balance is 10000

On filling Incorrect Detail:


localhost:8888/banking2/transfer.jsp?accountno=1&username=Rishi+kumar&password=riahi&taccountno=675&amount=10000

APANA - BANK
ExtraOrdinary Service

NEW ACCOUNT | BALANCE | DEPOSIT | WITHDRAW | TRANSFER | CLOSE A/C | ABOUT US

Services
- www.javatpoint.com
- www.javacstpoint.com
- www.javatpoint.com/forum.jsp

TRANSFER FORM

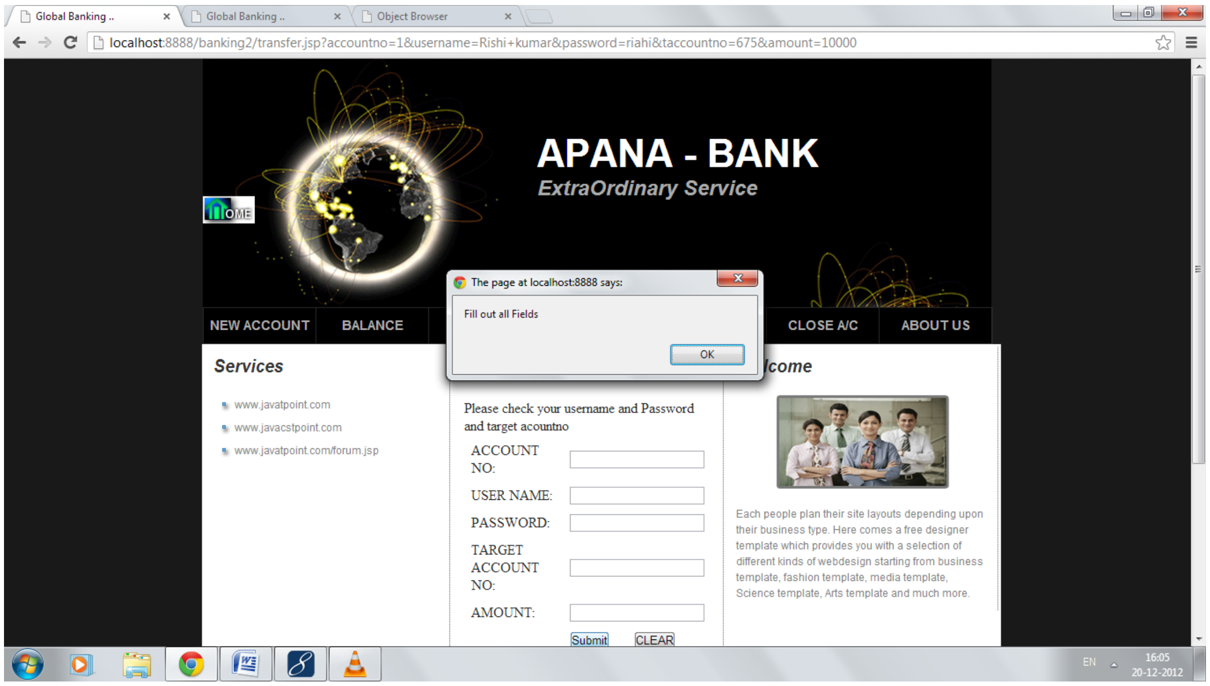
Please check your username and Password and target acountno

ACCOUNT NO:	
USER NAME:	
PASSWORD:	
TARGET ACCOUNT NO:	
AMOUNT:	

Submit CLEAR

Welcome

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

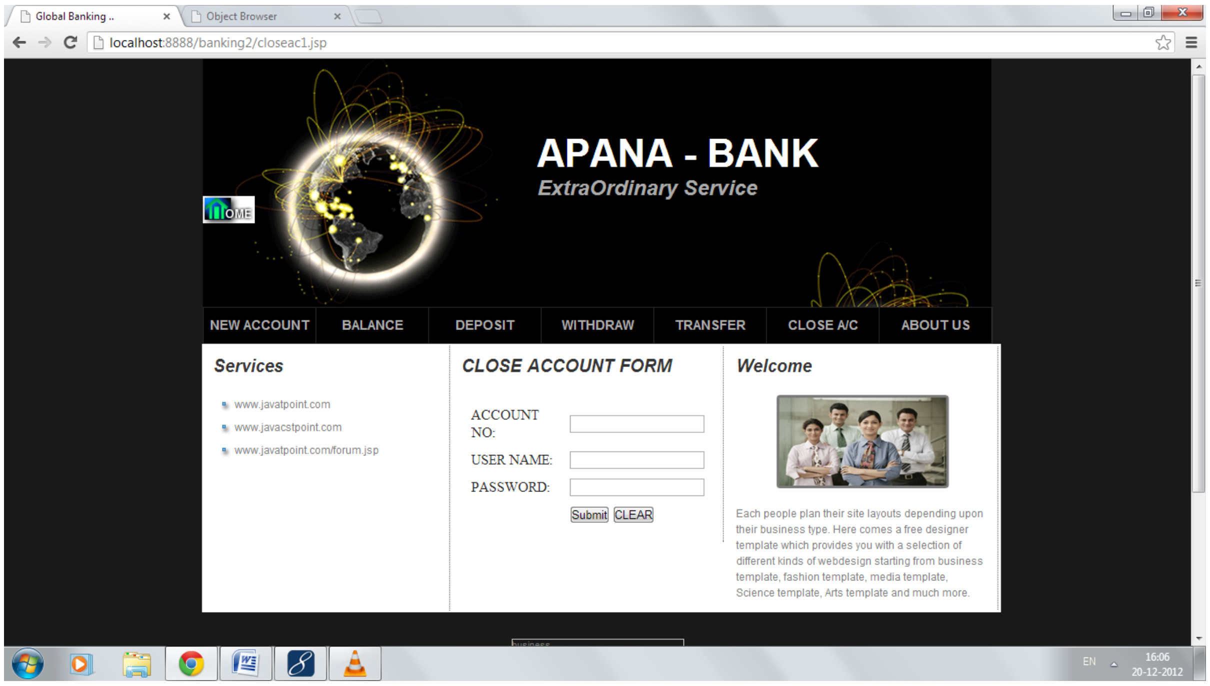

localhost:8888/banking2/transfer.jsp?accountno=1&username=Rishi+kumar&password=riahi&taccountno=675&amount=10000

The page at localhost:8888 says:

Fill out all Fields

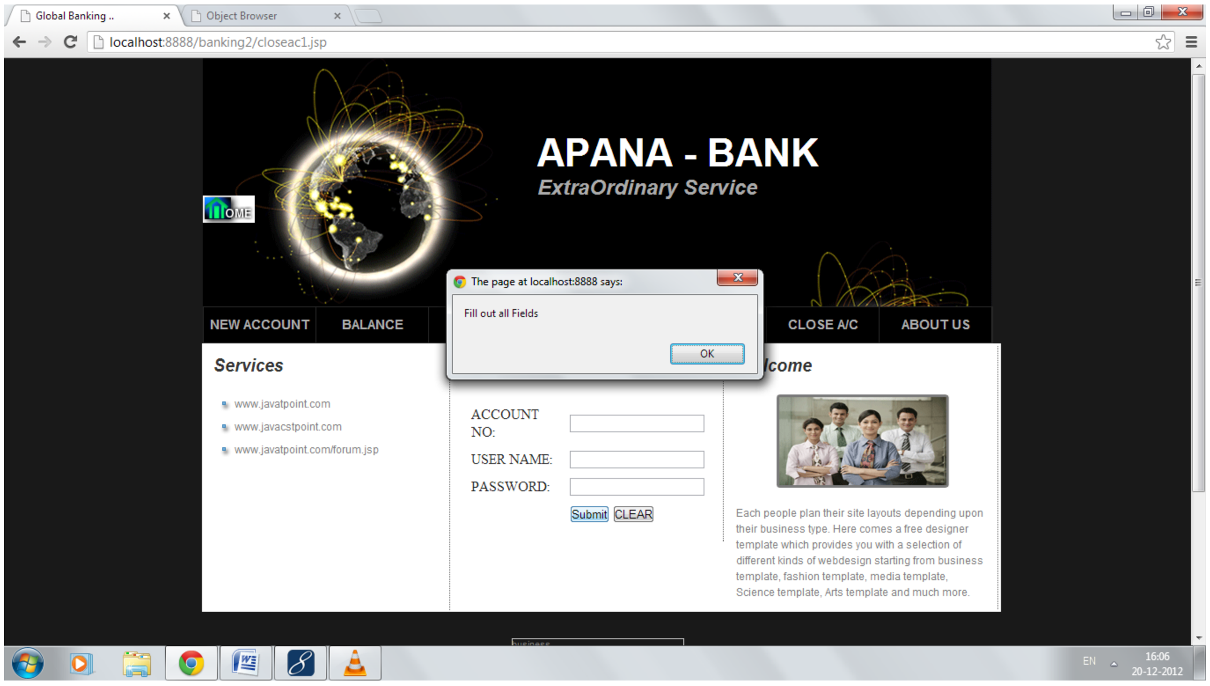
OK

6.Can close Account:



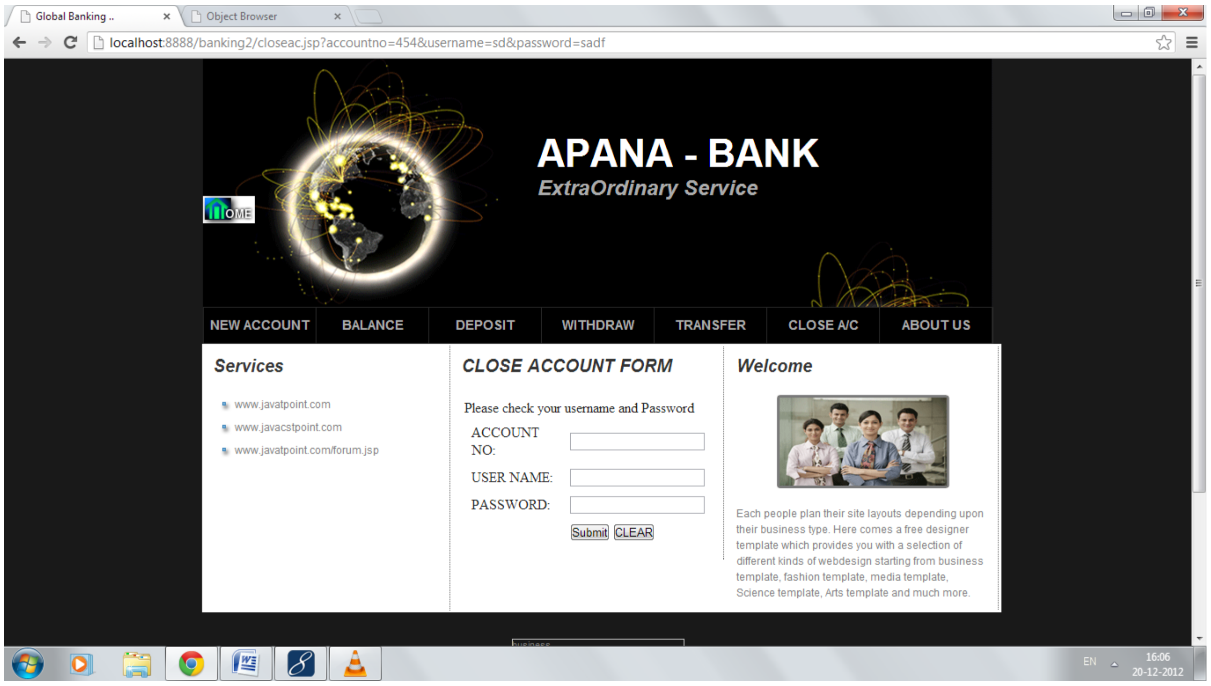
Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling incorrect Detail:



Fill out all Fields

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

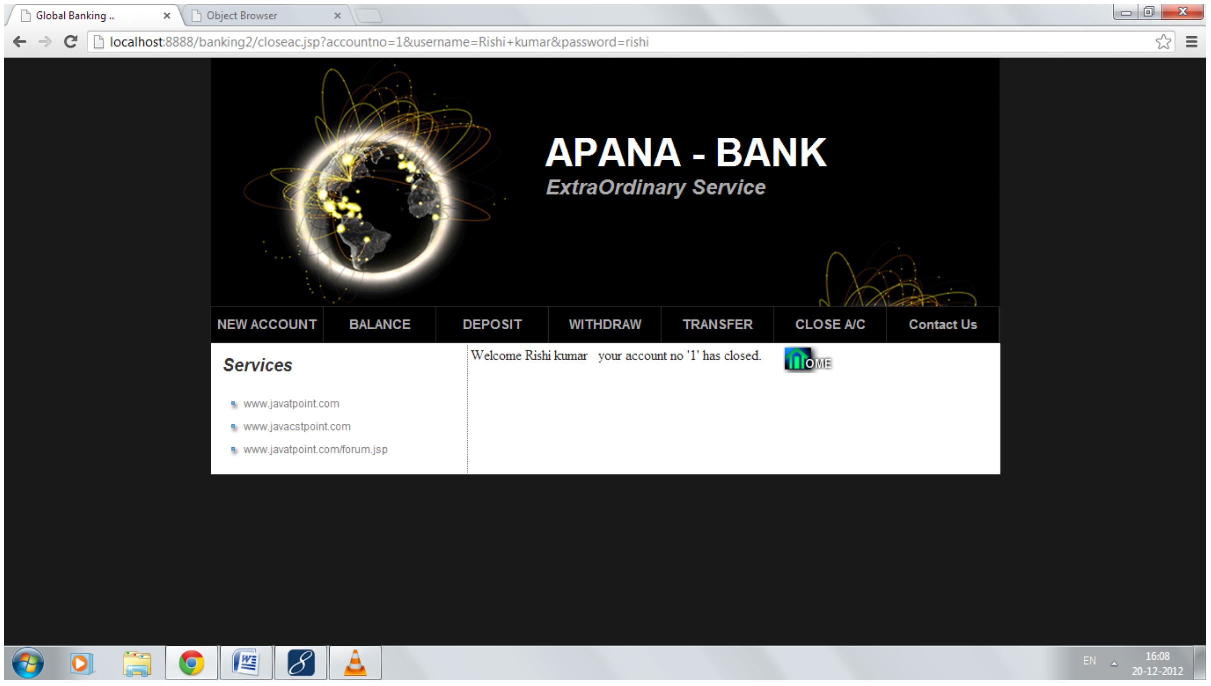


Please check your username and Password

ACCOUNT NO:	
USER NAME:	
PASSWORD:	

Each people plan their site layouts depending upon their business type. Here comes a free designer template which provides you with a selection of different kinds of webdesign starting from business template, fashion template, media template, Science template, Arts template and much more.

On filling Correct Detail:



Welcome Rishi kumar your account no '1' has closed.

CHAPTER- 8

SYSTEM TESTING AND IMPLIMENTATION

8.1. INTRODUCTION

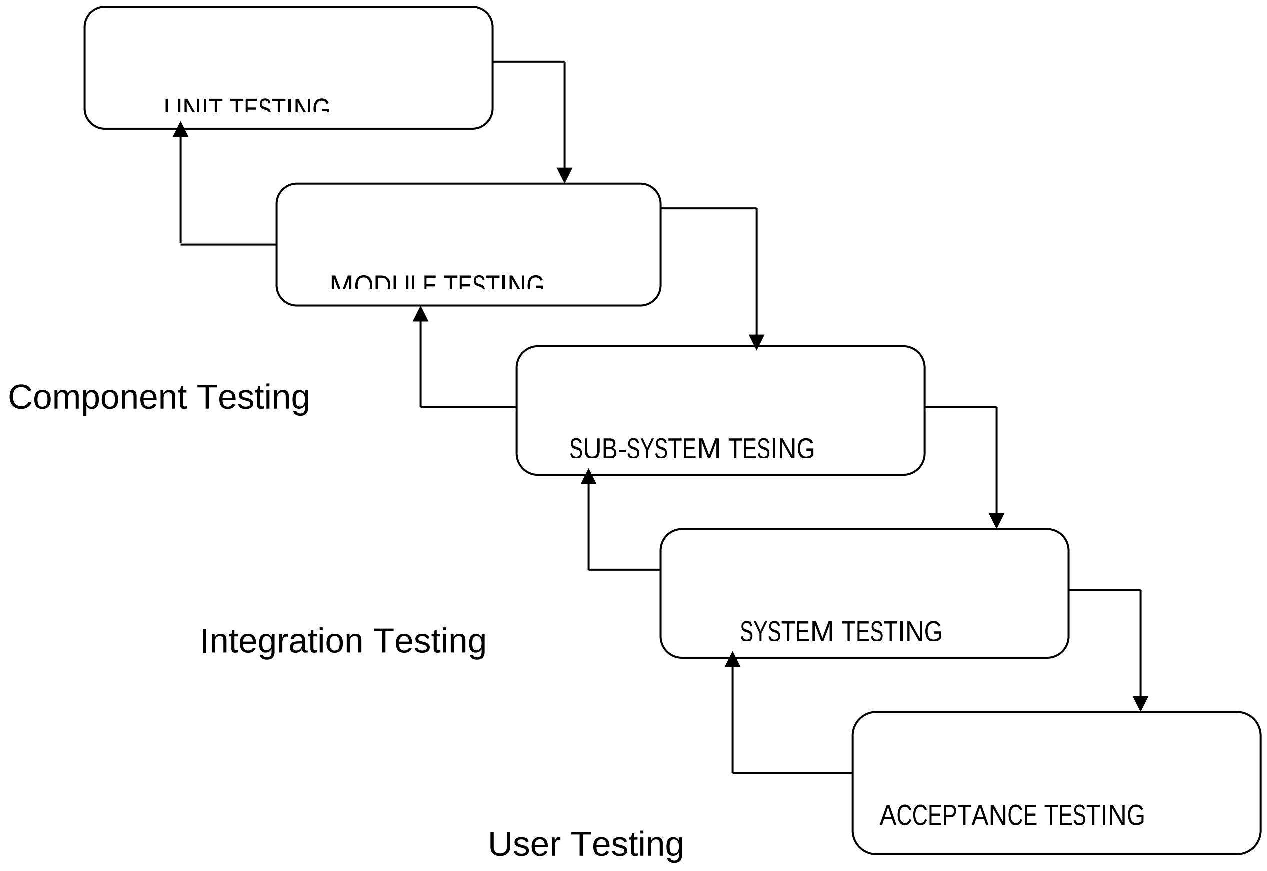
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

8.2. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



8.3. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

2. BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$V(G)=E-N+2$ or

$V(G)=P+1$ or

$V(G)=\text{Number Of Regions}$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

CHAPTER- 9

SYSTEM SECURITY

9.1. Introduction

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

9.2. SECURITY IN SOFTWARE

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employees two types of checks and controls:

CLIENT SIDE VALIDATION

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

- VBScript in used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
- Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

SERVER SIDE VALIDATION

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the

organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.

- Using server side validation, constraints on several restricted operations are imposed.