

# CHAPTER 1

# **CHAPTER-1**

## **INTRODUCTION**

### **1.1 Motivation**

The proposed system is based on the idea that if a centralized Mobile Number Portability Database is available for access by all the Telecom Service Providers, then it would greatly reduce the MNP conversion process duration to within the same day or even same time updation and approval of Mobile porting from one service provider to another.

The HLRs of each of the Mobile Service Providers may be in their own respective CRM databases for security reasons, but MNP numbers porting and approval should be maintained in a common MNP database for reducing the processing time from 2 or 3 days at present to within 1 Hour or even 30 minutes.

### **1.2 Problem Definition**

The actual problem is to create a web-based CRM and MNP application that addresses the problem of Mobile Number Porting from one service provider to the other without much delay in the process by providing a centralized database called MNP for all the service providers whereas, a separate CRM database for each service provider for maintaining customer details and connection details in a secure manner.

### **1.3 Objective of the project**

This Web Application is an CRM & MNP Application that runs with separate CRM database for each service provider and a common MNP database for all the service providers. The main purpose for this is to reduce the processing time of 3 to 4 days taken by service providers now to just within one hour or 30 min. for the convenience of customers.

## CHAPTER 2

## **CHAPTER 2**

### **PROFILE OF INDUSTRY**

#### **2.1 Profile of Industry**

**“Bharat Sanchar Nigam Ltd”**. was incorporated on 15th September 2000. It took over the business of providing of telecom services and network management from the erstwhile Central Government Departments of Telecom Services (DTS) and Telecom Operations (DTO), with effect from 1st October‘2000 on going concern basis. It is one of the largest & leading public sector units providing comprehensive range of telecom services in India.

BSNL has installed Quality Telecom Network in the country & now focusing on improving it, expanding the network, introducing new telecom services with ICT applications in villages & winning customer's confidence. Today, it has about 43.74 million line basic telephone capacity, 8.83 million WLL capacity, 72.60 million GSM capacity, 37,885 fixed exchanges, 68,162 GSM BTSs, 12,071 CDMA Towers, 197 Satellite Stations, 6, 86,644 RKm. of OFC, 50,430 RKm. of microwave network connecting 623 districts, 7330 cities/towns &BSNL is the only service provider, making focused efforts & planned initiatives to bridge the rural-urban digital divide in ICT sector.

BSNL has set up a world class multi-gigabit, multi-protocol convergent IP infrastructure that provides convergent services like voice, data & video through the same Backbone & Broadband Access Network. At present there are 8.09 million broadband customers.

##### **2.1.1 Vision:**

- Be the leading telecom service provider in India with global presence.
- Create a customer focused organization with excellence in customer care, sales and marketing.
- Leverage technology to provide affordable and innovative telecom. Services/ products across customer segments.

##### **2.1.2 Mission:**

- Be the leading telecom service provider in India with global presence.
  - Generating value for all stakeholders - employees, shareholders, vendors & business associates
  - Maximizing return on existing assets with sustained focus on profitability
  - Becoming the most trusted, preferred and admired telecom brand
  - To explore International markets for Global presence
- Creating a customer focused organization with excellence in customer care, sales& marketing .
  - Developing a marketing and sales culture that is responsive to customer needs mer care, sales& marketing
  - Excellence in customer service-”friendly, reliable, time bound, convenient and courteous service”

- Leveraging technology to provide affordable and innovative products/ services across customer segments

## 2.2 Profile of company:

### Organization documentation

**“Regional Telecom Training Center, Hyderabad”** is a reputed ISO 9001:2008 certified training center of Bharath Sanchar Nigam Limited , a Government of India Enterprise, well known for its Trainings in Latest Telecom Technologies and Information Technology. The Regional Telecom Training Center Hyderabad contributing it's share of Human Resource Development of BSNL, is situated in its own building at Gachibowli along old Bombay highway about 22 Kms from Secunderabad railway station and 15 Kms from Hyderabad railway station. It is in the vicinity of Hi-tec city a prime locality surrounded by Engineering staff college of India, Hyderabad Central University, International Institute of Information Technology, Indian school of Business and Sports village, with a number of prominent IT companies including CMC, WIPRO, Microsoft, and Infosys. Training center founded on 01-06-1973, Certified to ISO 9001:2008 The Administrative and academic block, the Hostel, the Staff quarters are spread over a sprawling area of about 42 acres. The administrative and academic block has a built up area of 1,27,000 sft covering administrative offices, ten Lecture halls, Switching, Transmission, Computer, Telegraph Labs, Library and other amenities like Canteen, Co-operative Stores, Recreation Club and Auditorium. There are Staff Quarters and Hostel, Garages for 4-wheelers & 2-wheelers and a Post Office building .

The following Trainings and Activities are conducted at RTTC,Hyderabad :

- 1). GSM – Radio frequency Planning
- 2) .Optical Fibre Communications Trainings
- 3) Training in Information Technology
- 4).Summer/In-Plant Trainings
- 5).Projects to Engineering Students in IT, GSM, Networking etc ..

## CHAPTER 3

## **CHAPTER 3**

### **LITERATURE SURVEY**

#### **3.1 Introduction**

We can get any information about the process online, because it is a web-based application. Convenient reports also could be generated because of the JSP, HTML and JQuery coding used in the project. The system is user friendly with JQuery Menu and a common theme for easy accessing of all web forms from within the menu etc.. Since The Database is MySQL, a powerful open-source database, with full normalization and. MySQL is a High-performance database with faster data retrievals.

#### **3.2 Existing system**

The existing system doesn't combine provisioning and MNP features for fast migration of a mobile number from one service provider to another. So also limited to a client server based application that requires some client software to be installed again at each Customer Service Centre.

#### **3.3 Disadvantages of existing system**

1. It is a redundant task to install client softwares at each and every Customer Service Center for running CRM software
2. It is a time taking task for Mobile Number portability at present as it is a different Process and needs additional software for porting.

#### **3.4 Proposed system**

The proposed system combines provisioning and MNP features and it is a web- based application that totally runs in the CRM server.

#### **Advantage:**

1. No need to install client software at each Customer Service Centre.
2. It also does mobile number porting at a very high speed.

### **3.5 Servlets/JSP**

#### **3.5.1 Introduction**

A Servlet is a generic server extension. a Java class that can be loaded Dynamically to expand the functionality of a server. Servlets are commonly used with web servers. Where they can take the place CGI scripts.

A servlet is similar to proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server, so it is safe and portable Servlets operate solely within the domain of the server. Unlike CGI and Fast CGI, which use multiple processes to handle separate program or separate Servlets are portable; both across operating systems and also across web servers. Java Servlets offer the best possible platform for web application development.

Servlets are used as replacement for CGI scripts on a web server, they can extend any sort of server such as a mail server that allows servlets extend its functionality perhaps by performing a virus scan on all attached documents or handling mail filtering tasks.

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming including inextensible scripting solutions platform-specific API's and incomplete interface.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the server-side what applets are to the client-side-object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects(without graphics or a GUI component).They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example an HTTP servlet can be used to generate dynamic HTML content when you use servlets to do dynamic content you get the following advantages:



- They're faster and cleaner than CGI scripts
- They use a standard API( the servlet API)
- They provide all the advantages of Java (run on a variety of servers without needing to be rewritten)

### **3.5.2 Attractiveness of servlets:**

They are many features of servlets that make them easy and attractive to use these include:

- Easily configure using the GUI-based Admin tool]
- Can be Loaded and Invoked from a local disk or remotely across the network.
- Can be linked together or chained, so that one servlet can call another servlet, or several servlets in sequence.
- Can be called dynamically from within HTML, pages using server-side include-tags.
- Are secure-even when downloading across the network, the servlet security model and servlet and box protect your system from unfriendly behavior.,

#### **3.5.2.1 Advantages of the servlet API**

One of the great advantages of the servlet API is protocol independent. It assumes nothing about:

- The protocol being used to transmit on the net
- How it is loaded
- The server environment it will be running in
- These quantities are important, because it allows the Servlet API to be embedded in many different kinds of servers. There are other advantages to the servlet API as well These include:
- It's extensible-you can inherit all your functionality from the base classes made available to you
- It's simple small, and easy to use.

### 3.6 Features of Servlets:

- Servlets are persistent. Servlet are loaded only by the web server and can maintain services between requests.
- Servlets are fast. Since servlets only need to be loaded once, they offer much better performance over their CGI counterparts.
- Servlets are platform independent.
- Servlets are extensible Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.
- Servlets are secure
- Servlets are used with a variety of client.

Servlets are classes and interfaces from to packages, `java. servlet` and `javax.servlet.http`. The `java. Servlet` package contains classes to support generic, protocol-independent servlets. The classes in the `javax.servelt.http` package to and HTTP specific functionality extend these classes

Every servlet must implement the `javax. servlet` interface. Most servlets implement it by extending one of two classes, `javax.servlet.GenericServlet` or `javax.servlet.http.HttpServlet`. A protocol-independent servlet should subclass `GenericServlet`. While an `Http` servlet should subclass `Http Servlet`, which is itself a subclass of `Generic-servlet` with added HTTP-specific functionality.

Unlike a java program, a servlet does not have a `main()` method, Instead the server in the process of handling requests invoke certain methods of a servlet. Each time the server dispatches a request to a servlet, it invokes the servlets `Service()` method,

A generic servlet should override its `service()` method to handle requests as appropriate for the servlet. The `service()` accepts two parameters a request object and a response object .The request object tells the servlet about the request, while the response object is used to return a response

In Contrast. An `Http` servlet usually does not override the `service()` method. Instead it overrides `doGet()` to handle GET requests and `do Post()` to handle Post requests. An `Http`

servlet can override either or both of these modules the `service()` method of `Http Servlet` handles the setup and dispatching to all the `do XXX()` methods. which is why it usually should not be overridden

The remainders in the `javax.servlet` and `javax.servlet.http.package` are largely support classes .The `ServletRequest` and `ServletResponse` classes in `javax.servlet` provide access to generic server requests and responses while `Http Servlet Request` and `Http Servlet Response` classes in `javax.servlet` provide access to generic server requests and responses while `Http Servlet Request` and `Http Servlet Response` in `javax.servlet.http` provide access a HTTP requests and responses . The `javax.servlet.http` provide contains an `HttpSession` class that provides built-in session tracking functionality and `Cookie` class that allows quickly setup and processing `Http Cookies`.

### **3.6.1 Loading Servlets:**

Servlets can be loaded from their places. From a directory that is on the `CLASSPATH`. The `CLASSPATH` of the Java Web Server includes `service root/classes/`, which is where the system classes reside

From the `<SERVICE_ROOT/servlets/directory`. This is not in the server's class path. A class loader is used to create servlets form this directory. New servlets can be added-existing servlets can be recompiled and the server will notice these changes. From a remote location .For this a code base like `http://nine.eng/classes/foo/` is required in addition to the servlet's class name. Refer to the admin Gui docs on servlet section to see how to set this up.

#### **Loading Remote Servlets**

Remote servlets can be loaded by:

- Configuring the admin Tool to setup automatic loading of remote servlets.
- Selection up server side include tags in .html files
- Defining a filter chain Configuration

### 3.7 Invoking Servlets

A servlet invoker is a servlet that invokes the “server” method on a named servlet. If the servlet is not loaded in the server, then the invoker first loads the servlet (either from local disk or from the network) and then invokes the “service” method. Also like applets, local servlets in the server can be identified by just the class name. In other words, if a servlet name is not absolute, it is treated as local.

A Client can Invoke Servlets in the Following Ways:

- The client can ask for a document that is served by the servlet.
- The client(browser) can invoke the servlet directly using a URL, once it has been mapped using the SERVLET ALIASES Section of the admin GUI
- The servlet can be invoked through server side include tags.
- The servlet can be invoked by placing it in the servlets/directory
- The servlet can be invoked by using it in a filter chain

#### 3.7.1 The Servlet Life Cycle:-

The Servlet life cycle is one of the most exciting features of Servlets. This life cycle is a powerful hybrid of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The servlet life cycle allows servlet engines to address both the performance and resource problems of CGI and the security concepts of low level server API programming.

Servlet life cycle is highly flexible. Servers have significant leeway in how they choose to support servlets. The only hard and fast rule is that a servlet engine must conform to the following life cycle contract:

- Create and initialize the servlets
- Handle zero or more service from clients
- Destroy the servlet and then garbage Collects it.

It's perfectly legal for a servlet to be loaded, created and initialized in its own JVM, only to be destroyed and garbage collected without handling any client request or after handling just one request

The most common and most sensible life cycle implantations for HTTP servlets are Single java virtual machine and astate persistence.

### **3.7.1.1 Init and Destroy:-**

Just like Applets servlets can define `init()` and `destroy()` methods, A servlets `init` (`ServiceConfig`) method is called by the server immediately after the server constructs the servlet's instance. Depending on the server and its configuration, this can be at any of these times

- When the server states
- When the servlet is first requested, just before the `service()` method is invoked
- At the request of the server administrator

In any case, `nit()` is guaranteed to be called before the servlet handles its first request

The `init()` method is typically used to perform servlet initialization creating or loading objects that are used by the servlet in handling of its request. In order to providing a new servlet any information about itself and its environment, a server has to call a servlets `init()` method and pass an object that implement the `ServletConfig` interface.

This `ServletConfig` object supplies a servlet with information about its initialization parameters. These parameters are given to the servlets and are not associated with any single request. They can specify initial values, such as where a counter should begin counting, or default values, perhaps a template to use when not specified by the request,

The server calls a servlet's `destroy()` method when the servlet is about to be unloaded. In the `destroy()` method, a servlet should free any resources it has acquired that will not be garbage collected. The `destroy()` method also gives a servlet a chance to write out its unsaved. cached information or any persistent information that should be read during the next call to `init()`.

### **3.8 Session Tracking:**

HTTP is a stateless protocol, it provides no way for a server to recognize that a sequence of requests is all from the same client. This causes a problem for application such as shopping cart applications. Even in chat application server can't know exactly who's making a request of several clients.

The solution for this is for client to introduce itself as it makes each request, Each clients needs to provide a unique identifier that lets the server identify it, or it needs to give some information that the server can use to properly handle the request, There are several ways to send this introductory information with each request Such as:

## CHAPTER 4

## **CHAPTER 4**

### **ANALYSIS**

#### **4.1 Introduction**

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system.

#### **4.2 Requirement specification**

##### **4.2.1 User requirements**

Requirements Specification plays an important role to create quality software solution. Requirements are refined and analyzed to assess the clarity.

Requirements are represented in a manner that ultimately leads to successful software implementation. Each requirement must be consistent with the overall objective. The development of this project deals with the following requirements:

- i). Hardware Requirements
- ii). Software Requirements

##### **4.2.2 Hardware requirements**

The selection of hardware is very important in the existence and performance of any software. The size and capacity are the main requirements. The typical Web Server must have the following specifications for good performance:

- System : Intel Core or above.
- Hard Disk : Minimum 40 GB.
- RAM : Minimum 1 GB or Above.



### 4.2.3 Software requirements

The software requirements specification is produced at the end of the analysis task. Software requirement is a difficult task, only decided after testing whether it fits the requirements.

- Operating System : Windows 7 Professional/Ultimate windows 8
- Technology : Data Source.
- Coding languages : JAVA,HTML,J QUERYC,JAVA SCRIPT
- Back-End : MySQL SERVER 5.1.
- Software : Apache tomcat server, Navicat

## CHAPTER 5

## CHAPTER 5

### 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 requirements 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.

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 Uml diagrams

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modeling tools include:

- **Use Case Diagram:** displays the relationship among actors and use cases.
- **Class Diagram** models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.
- **Sequence Diagram** displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

- **Collaboration Diagram** displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.
- **State Diagram** displays the sequences of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.
- **Activity Diagram** displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

### 5.3 Physical Diagrams

- **Component Diagram** displays the high level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time.
- **Deployment Diagram** displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code units.

### 5.4 Goals of UML

The primary goals in the design of the UML are:

1. Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

### Use Case Diagram:

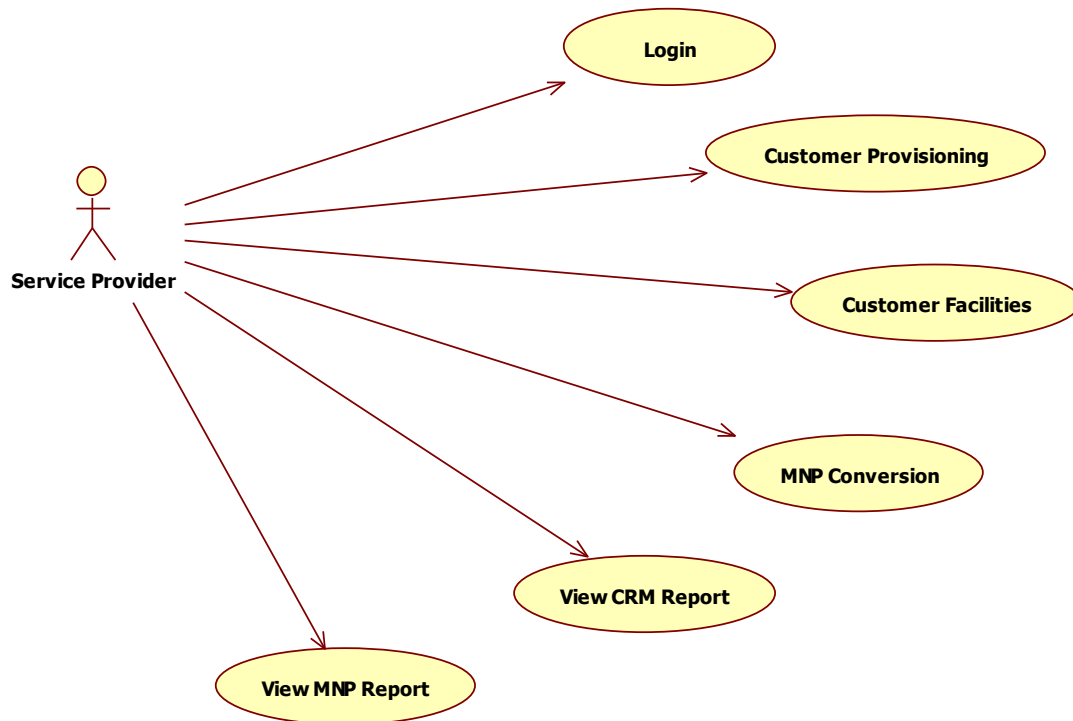


FIGURE 5.1 case diagram for CRM-MNP

Class Diagram:

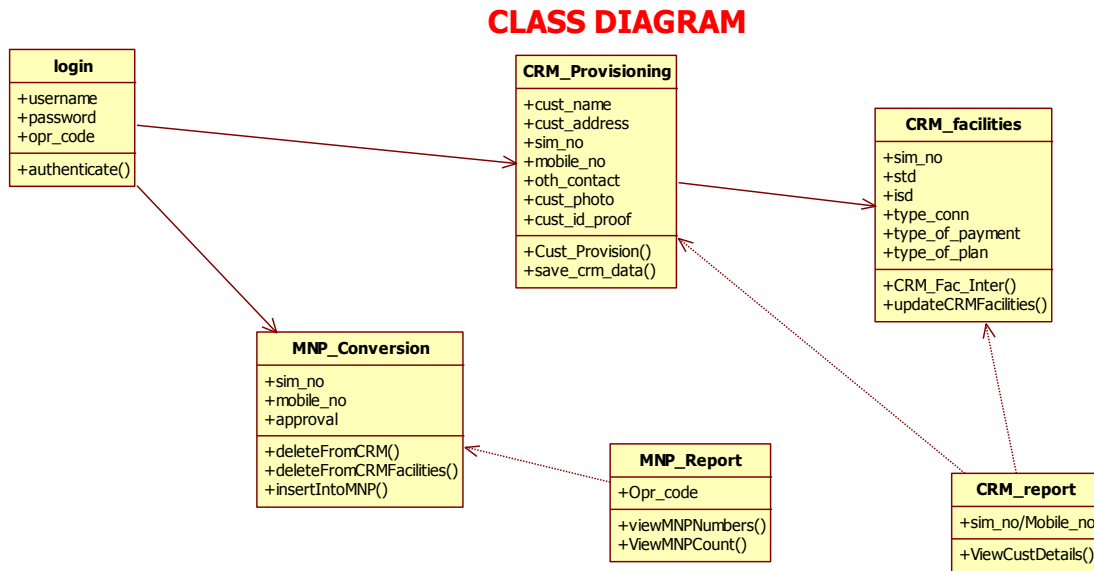


Figure 5.2 class diagram

## Activity Diagram:

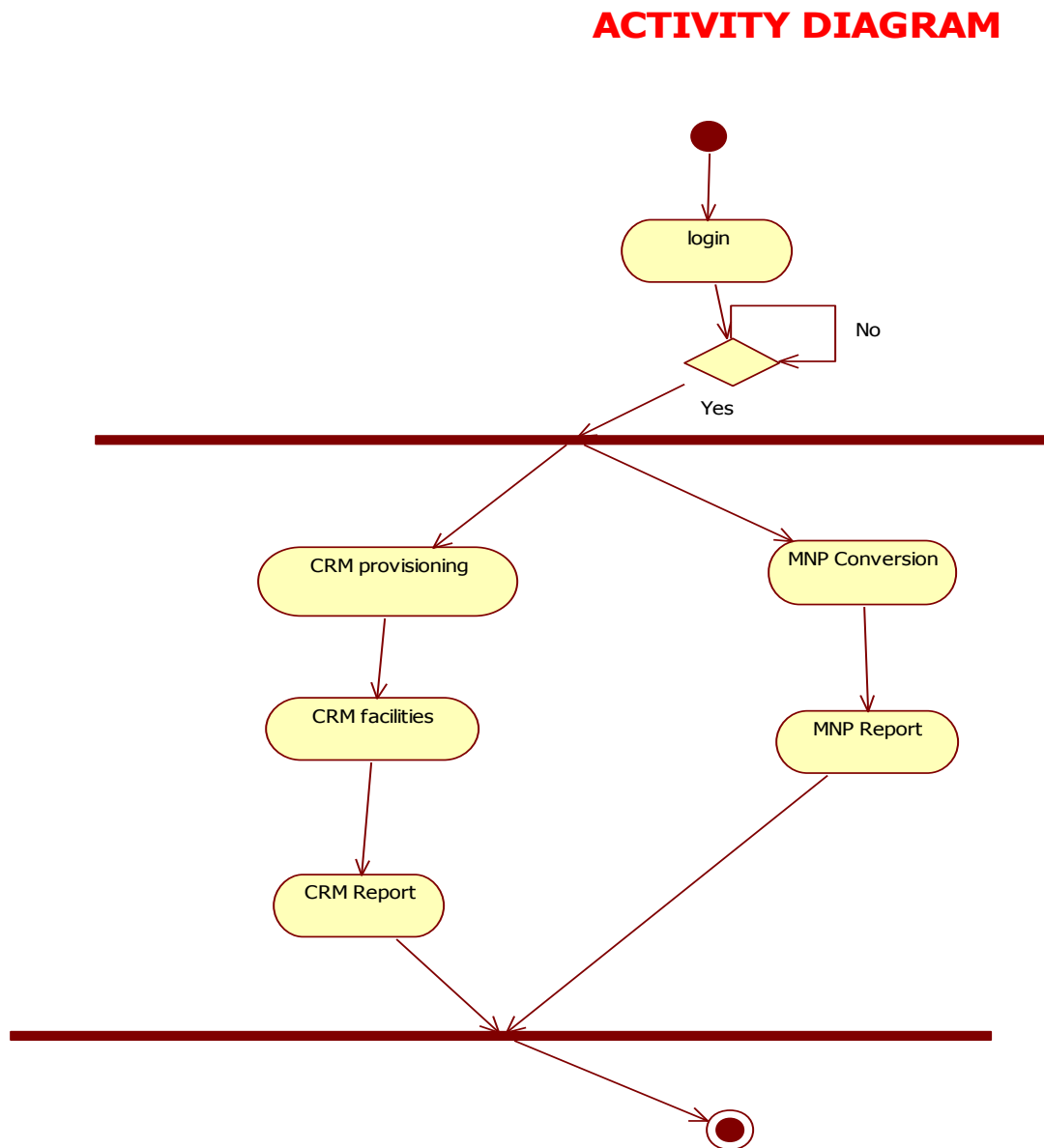
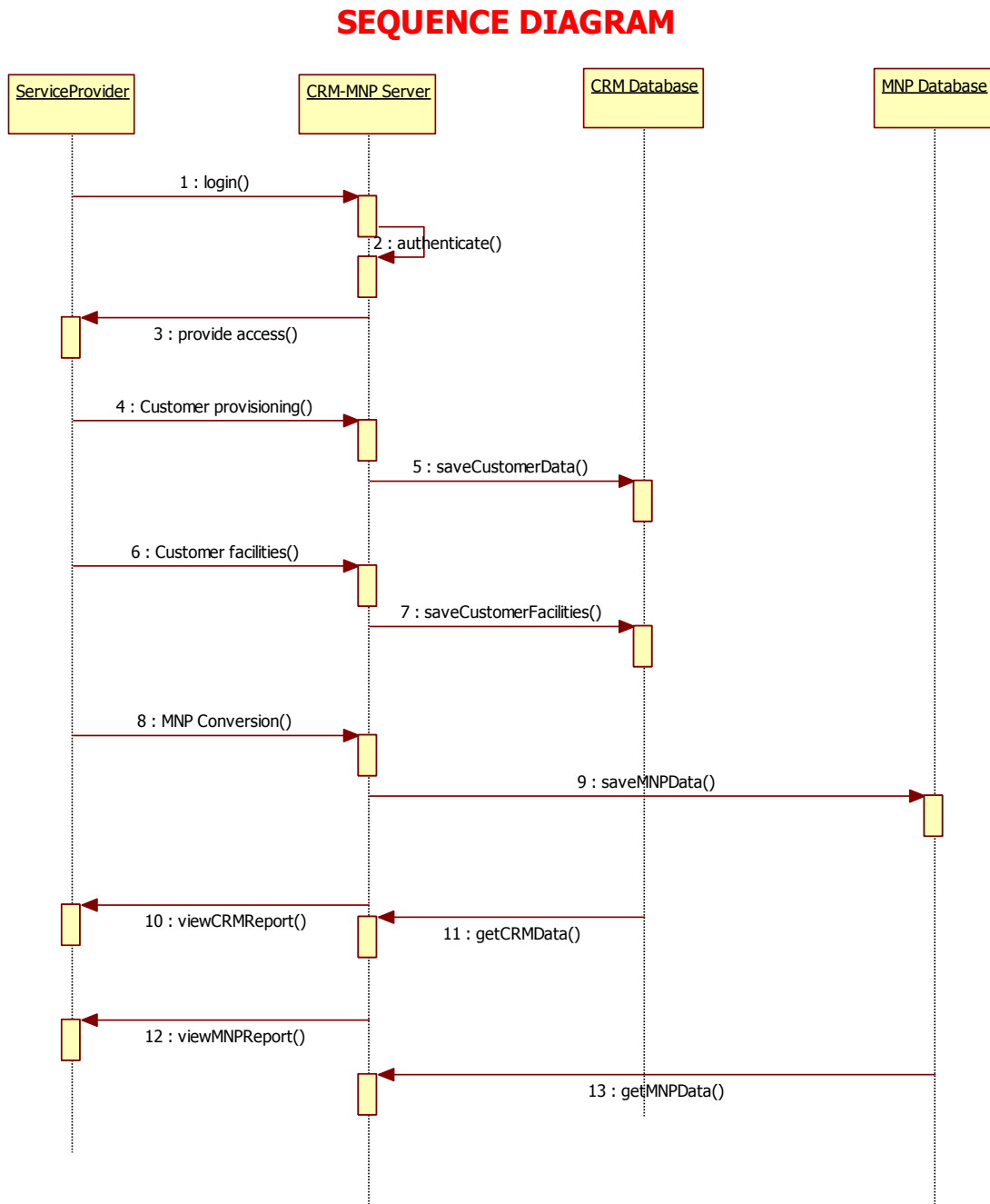


Figure 5.3: Activity Diagram



## Sequence Diagram:



**Figure 5.4: Sequence Diagram**

### Component Diagram :

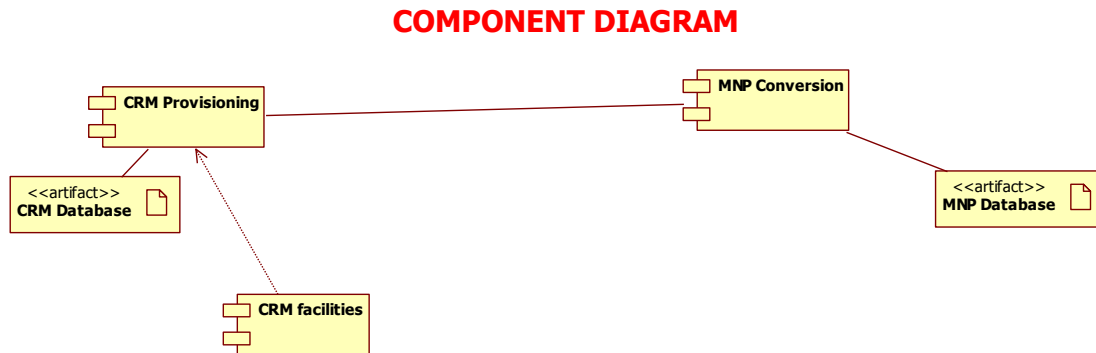


Figure 5.5: Component Diagram

### Deployment Diagram:

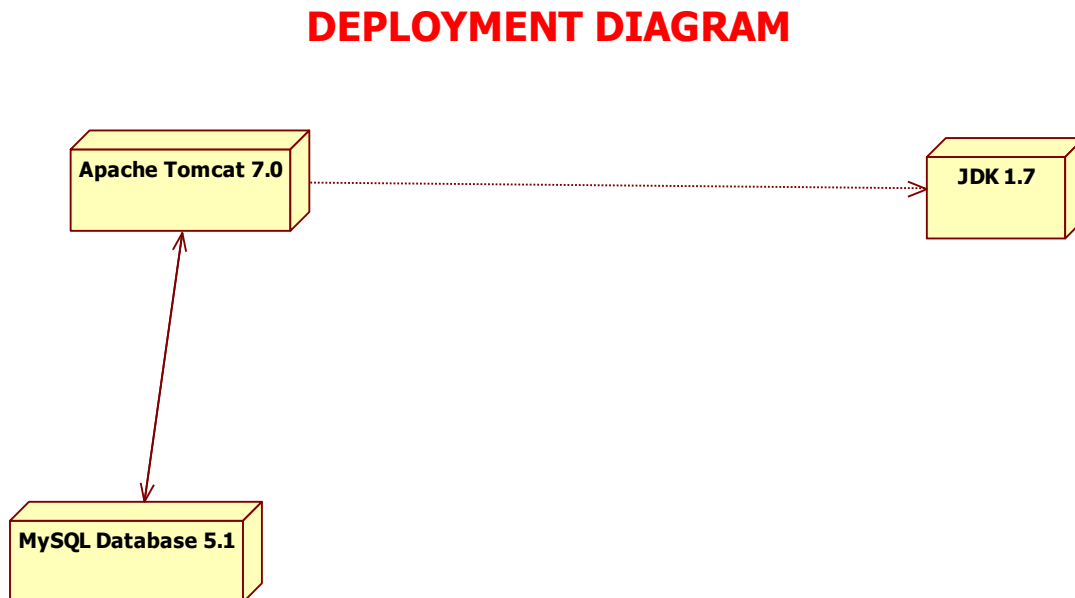


Figure 5.6:Deployment Diagram

## 5.5 Module design and organization

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 requirements 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 access for quality.

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 structural Design, Interface Design and Procedural Design.

### 5.5.1 Modules Description:

- 1. Login Module:** This is a module required for entering into CRM Software
- 2. CRM number provisioning Module:** This is for providing a new mobile number toThe customer and storing it in HLR followed by activation of the mobile
- 3. ADD Mobile Plans:** This is for creating new plans for the network.
- 4. Customer Facilities:** This is a module required for giving facilities and plans to mobile number.
- 5. Mobile Number Portability Module:** This is for porting a customer from one service Provider to another. This requires special authentication from MNP admin user.
- 6. CRM Reports:** CRM and Mobile Reports could be generated in this module
- 7. MNP Report:** MNP ported mobile number Report generated.

## CHAPTER 6

## CHAPTER 6

### IMPLEMENTATION AND RESULTS

#### 6.1 Introduction

Implementation literally means to put into effect or to carry out. The system implementation phase, the software deals with translation of the design specifications into source code. The ultimate goal of the implementation is to write the source code and the internal documentation so that it can be verified easily. The code and documentation should be written in a manner that eases debugging, testing and modification. System flow charts, sample run on packages, sample output etc., is part of the implementation. An effort was to made to satisfy the following goals in order:

- Clarity and simplicity of the code
- Minimization of Hard Coding
- Minimization of the amount of memory used
- Thorough phased implementation has been done so that we can use our proposed system correctly.

#### 6.2 Method of implementation

##### 6.2.1 Data base tables

**Description :** We are using NAVICAT Software to creating the Table

##### LOGIN TABLE:

```
CREATE TABLE `login` (`username` varchar(50) COLLATE utf8_unicode_ci NOT NULL
DEFAULT "",`password` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL, `opr_code`
char(2) COLLATE utf8_unicode_ci DEFAULT NULL,PRIMARY KEY (`username`))
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

User name	Password
Bsnl	b1234
Airtel	a1234
Idea	i1234
Uninor	u1234
Vodafone	v1234


Fields	Indexes	Foreign Keys	Triggers	Options	Comment	SQL Preview
Name	Type	Length	Decimals	Allow Null		
username	varchar	50	0	<input type="checkbox"/>	 1	
password	varchar	20	0	<input checked="" type="checkbox"/>		
opr_code	char	2	0	<input checked="" type="checkbox"/>		

Figure 6.1:login database

**Customer Details Table:**

```
CREATE TABLE `cust_details` (`mobile_no` bigint(12) NOT NULL DEFAULT '0',`opr_code` varchar(2) COLLATE utf8_unicode_ci NOT NULL DEFAULT "",`sim_no` bigint(20) NOT NULL DEFAULT '0',`cust_name` varchar(150) COLLATE utf8_unicode_ci DEFAULT NULL, `cust_photo_path` varchar(450) COLLATE utf8_unicode_ci DEFAULT NULL,`cust_id_proof` varchar(450) COLLATE utf8_unicode_ci DEFAULT NULL,`cust_address` varchar(450) COLLATE utf8_unicode_ci DEFAULT NULL,`oth_contact` bigint(12) DEFAULT NULL,`date_of_prov` date DEFAULT NULL,PRIMARY KEY(`mobile_no`,`opr_code`,`sim_no`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Name	Type	Length	Decimals	Allow Null	
mobile_no	bigint	12	0	<input type="checkbox"/>	1
opr_code	varchar	2	0	<input type="checkbox"/>	2
sim_no	bigint	20	0	<input type="checkbox"/>	3
cust_name	varchar	150	0	<input checked="" type="checkbox"/>	
cust_photo_path	varchar	450	0	<input checked="" type="checkbox"/>	
cust_id_proof	varchar	450	0	<input checked="" type="checkbox"/>	
cust_address	varchar	450	0	<input checked="" type="checkbox"/>	
oth_contact	bigint	12	0	<input checked="" type="checkbox"/>	
date_of_prov	date	0	0	<input checked="" type="checkbox"/>	

Figure 6.2: Customer Details

**Customer Facilities Table:**

```
CREATE TABLE `cust_facilities` (
  `sim_no` bigint(20) NOT NULL DEFAULT '0',
  `std` char(4) COLLATE utf8_unicode_ci DEFAULT NULL,
  `ISD` char(4) COLLATE utf8_unicode_ci DEFAULT NULL,
  `type_of_conn` varchar(5) COLLATE utf8_unicode_ci DEFAULT NULL,
  `type_of_payment` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  `plan_no` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  `type_plan` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`sim_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Name	Type	Length	Decimals	Allow Null	
sim_no	bigint	20	0	<input type="checkbox"/>	1
std	char	4	0	<input checked="" type="checkbox"/>	
ISD	char	4	0	<input checked="" type="checkbox"/>	
type_of_conn	varchar	5	0	<input checked="" type="checkbox"/>	
type_of_payment	varchar	20	0	<input checked="" type="checkbox"/>	
plan_no	varchar	20	0	<input checked="" type="checkbox"/>	
type_plan	varchar	20	0	<input checked="" type="checkbox"/>	

Figure 6.3: Customer Facilities

**Plans Table:**

```
CREATE TABLE `plans` (`plan_no` varchar(20) COLLATE utf8_unicode_ci NOT
NULL DEFAULT "", `plan_desc` varchar(150) COLLATE utf8_unicode_ci DEFAULT
NULL, `type_of_plan` varchar(25) COLLATE utf8_unicode_ci DEFAULT NULL,
`plan_type` varchar(25) COLLATE utf8_unicode_ci DEFAULT NULL, `type_of_conn`
varchar(5) COLLATE utf8_unicode_ci DEFAULT NULL, PRIMARY KEY (`plan_no`))
ENGINE=InnoDB DEFAULT CHARSET=utf8COLLATE=utf8_unicode_ci;
```

Fields	Indexes	Foreign Keys	Triggers	Options	Comment	SQL Preview
Name	Type	Length	Decimals	Allow Null		
plan_no	varchar	20	0	<input type="checkbox"/>		1
plan_desc	varchar	150	0	<input checked="" type="checkbox"/>		
type_of_plan	varchar	25	0	<input checked="" type="checkbox"/>		
plan_type	varchar	25	0	<input checked="" type="checkbox"/>		
type_of_conn	varchar	5	0	<input checked="" type="checkbox"/>		

**Figure 6.4: Plans Table****MNP DATA BASE:****MNP operators Table:**

```
CREATE TABLE `mnp_operators` (`opr_code` char(2) COLLATE utf8_unicode_ci NOT
NULL DEFAULT "", `opr_name` varchar(150) COLLATE utf8_unicode_ci DEFAULT
NULL, PRIMARY KEY (`opr_code`)) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci;
```

opr_code	opr_name
A	airtel
B	bsnl
I	idea
U	uninor
V	vodafone


**Figure 6.5: MNP operators and MNP data base**



**MNP Opr Mapping Table:**

```
CREATE TABLE `mobile_opr_mapping` (`opr_code` char(1) COLLATE
utf8_unicode_ci DEFAULT NULL,`mobile_no` bigint(15) NOT NULL DEFAULT '0',
PRIMARY KEY (`mobile_no`)) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci;
```

opr_code	mobile_no
B	9908153591
I	9912297779
A	9998989898

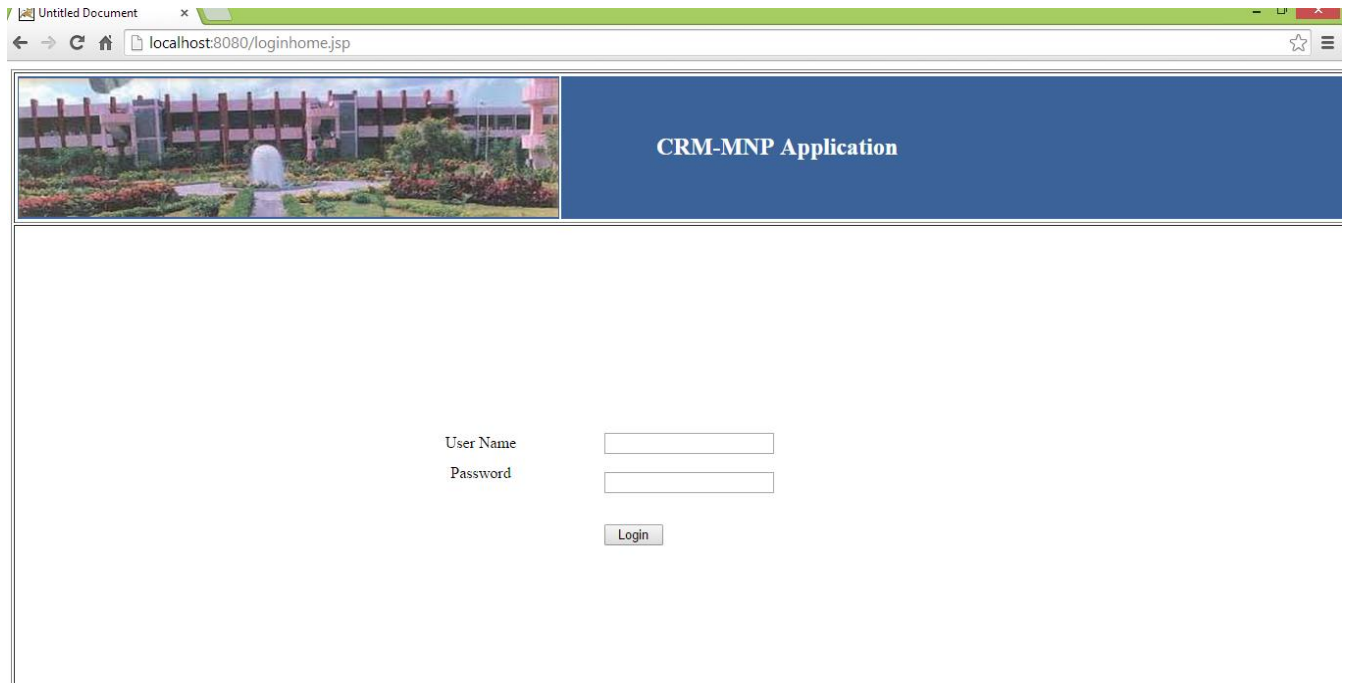
Name	Type	Length	Decimals	Allow Null	
opr_code	char	1	0	<input checked="" type="checkbox"/>	
mobile_no	bigint	15	0	<input type="checkbox"/>	 1

**Figure 6.6:MNP Opr Mapping and MNP OPR data base****6.3 RESULT ANALYSIS**

The code and documentation should be written in a manner that eases debugging, testing and modification. System flow charts, sample run on packages, sample output etc., is part of the implementation. An effort was made to satisfy the following goals in order:

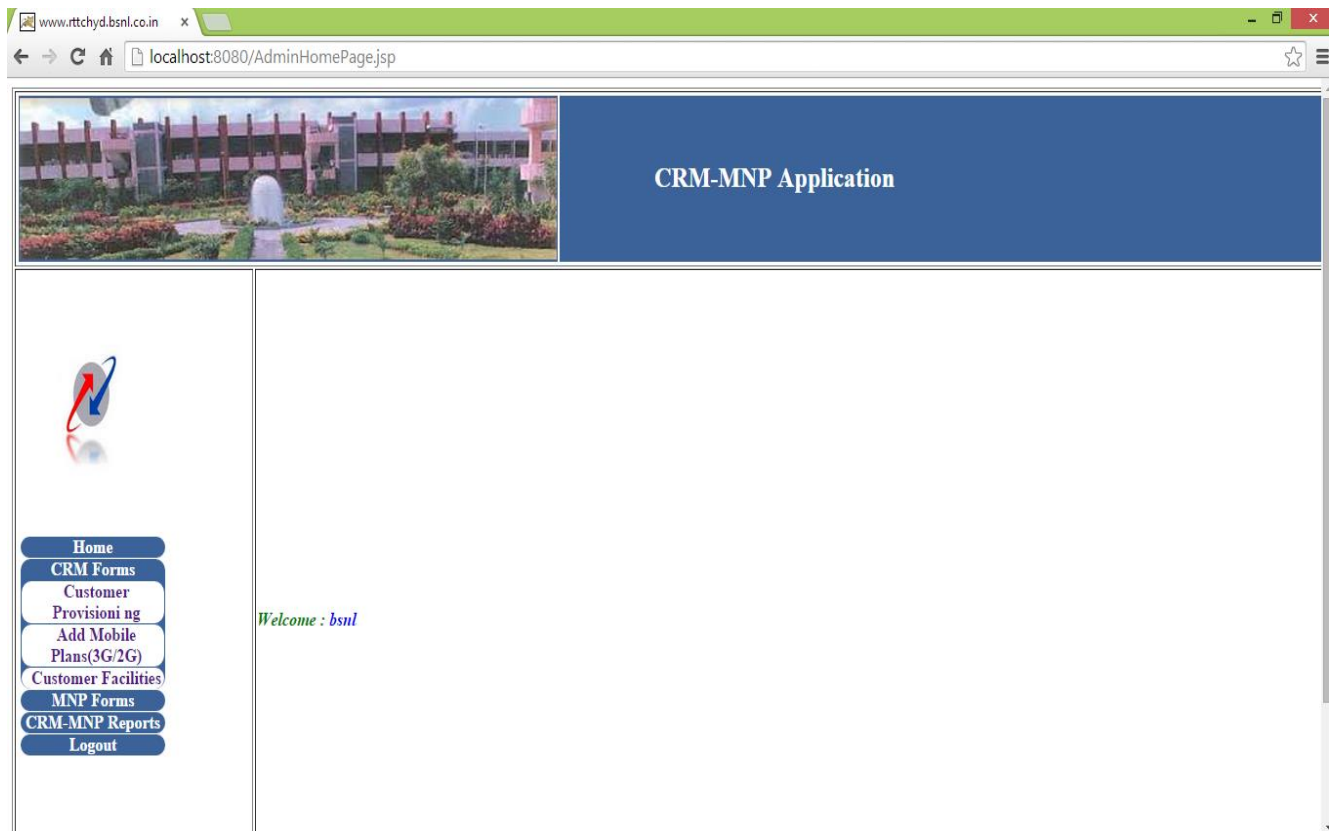
### 6.3.1 Output Screens

#### LOGIN PAGE:



**Figure 6.7:login page**

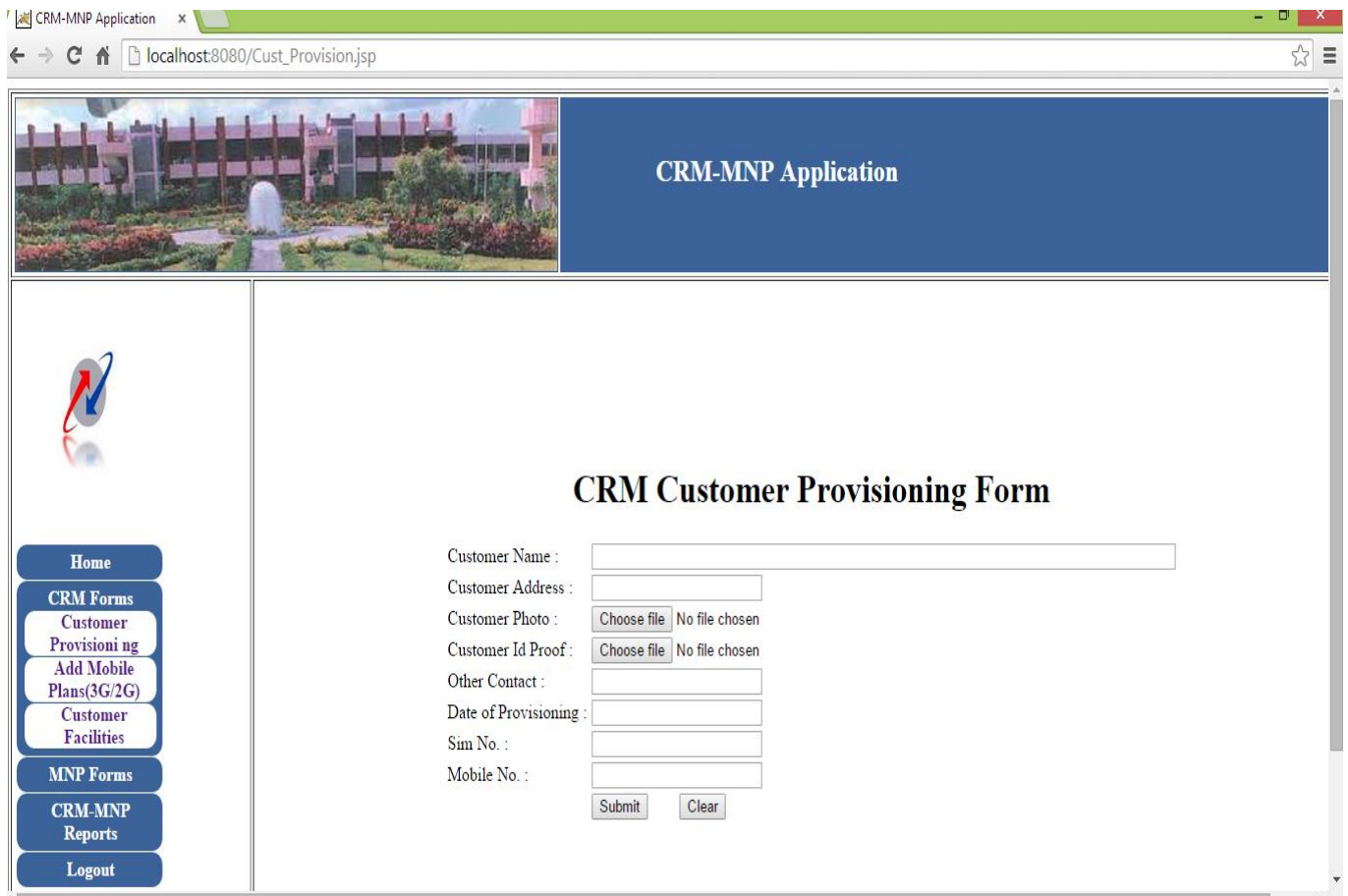
## HOME PAGE:



**Figure 6.8:Home page**

## CRM FORM:

### 1. Customer provisioning:

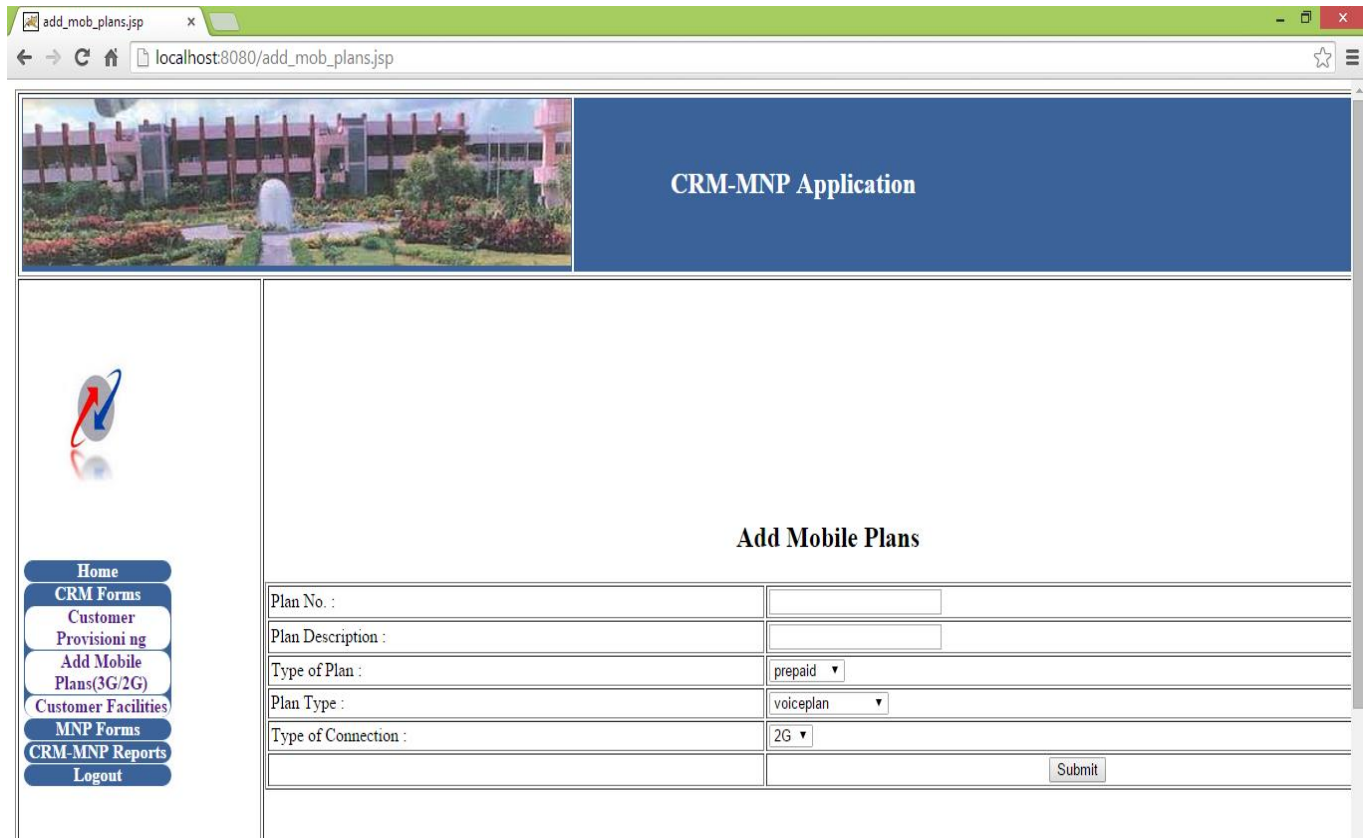


The screenshot shows a web browser window with the title "CRM-MNP Application" and the URL "localhost:8080/Cust\_Provision.jsp". The page features a header with a landscape image and the text "CRM-MNP Application". A left sidebar contains a navigation menu with the following items: Home, CRM Forms, Customer Provisioning, Add Mobile Plans(3G/2G), Customer Facilities, MNP Forms, CRM-MNP Reports, and Logout. The main content area is titled "CRM Customer Provisioning Form" and contains the following fields and controls:

- Customer Name :
- Customer Address :
- Customer Photo :  No file chosen
- Customer Id Proof :  No file chosen
- Other Contact :
- Date of Provisioning :
- Sim No. :
- Mobile No. :
- 

**Figure 6.9: CRM Customer provisioning form**

## 2. Add mobile Plans(2G/3G):



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/add\_mob\_plans.jsp'. The page features a header with a landscape image and the text 'CRM-MNP Application'. A left sidebar contains a navigation menu with buttons for Home, CRM Forms, Customer Provisioning, Add Mobile Plans(3G/2G), Customer Facilities, MNP Forms, CRM-MNP Reports, and Logout. The main content area is titled 'Add Mobile Plans' and contains a form with the following fields:

Plan No. :	<input type="text"/>
Plan Description :	<input type="text"/>
Type of Plan :	prepaid ▼
Plan Type :	voiceplan ▼
Type of Connection :	2G ▼
<input type="button" value="Submit"/>	

**Figure 6.10:add mobile plans**

### 3.Customer Facilites:

The screenshot displays the 'Customer Facilities' page of the CRM-MNP Application. The browser window shows the URL 'localhost:8080/Cust\_Facjsp'. The page layout includes a header with a landscape image and the title 'CRM-MNP Application'. A left sidebar contains navigation buttons: Home, CRM Forms, Customer Provisioning, Add Mobile Plans(3G/2G), Customer Facilities, MNP Forms, CRM-MNP Reports, and Logout. The main content area is titled 'Customer Facilities' and contains a form with the following fields:

SIM No.:	<input type="text" value="Select SIM No."/>
STD:	<input type="text"/>
ISD:	<input type="text"/>
<input checked="" type="radio"/> 3G <input type="radio"/> 2G	<input type="text"/>
<input checked="" type="radio"/> Post-Paid <input type="radio"/> Pre-Paid	<input type="text"/>
<input checked="" type="radio"/> Voice Plan <input type="radio"/> Data Plan <input type="radio"/> Voice & Data Plan	<input type="text"/>
Next	<input type="text"/>

**Figure 6.11:customer facilities**

MNP FORM:

Mobile Number porting:

The screenshot displays a web browser window with the address bar showing 'localhost:8080/mnp\_approve.jsp'. The application header features a banner image of a building and the title 'CRM-MNP Application'. A left sidebar contains a navigation menu with the following items: Home, CRM Forms, Customer Provisioning, Add Mobile Plans(3G/2G), Customer Facilities, MNP Forms, CRM-MNP Reports, and Logout. The main content area is titled 'Mobile Number Porting' and contains a form with two rows of input fields. The first row is labeled 'Mobile No:' and the second row is labeled 'Change Network'. The second row also includes a dropdown menu labeled '-----Select Operator-----'. A 'Submit' button is located below the form fields.

Mobile Number Porting	
Mobile No:	<input type="text"/>
Change Network	<input type="text"/> -----Select Operator----- ▼

Submit

**Figure 6.12:mobile number porting**

CRM&MNP REPORT:

MNP REPORT:

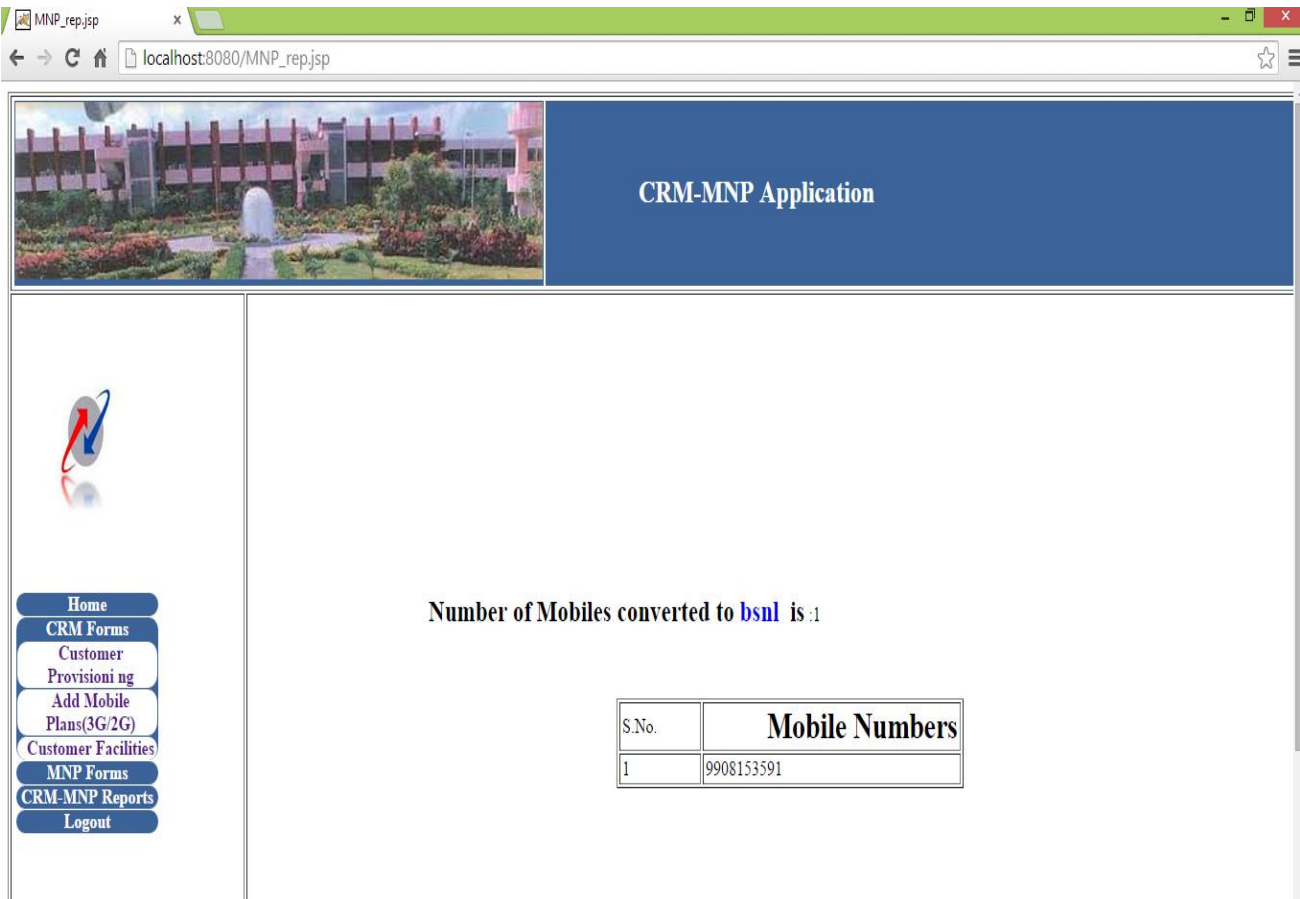
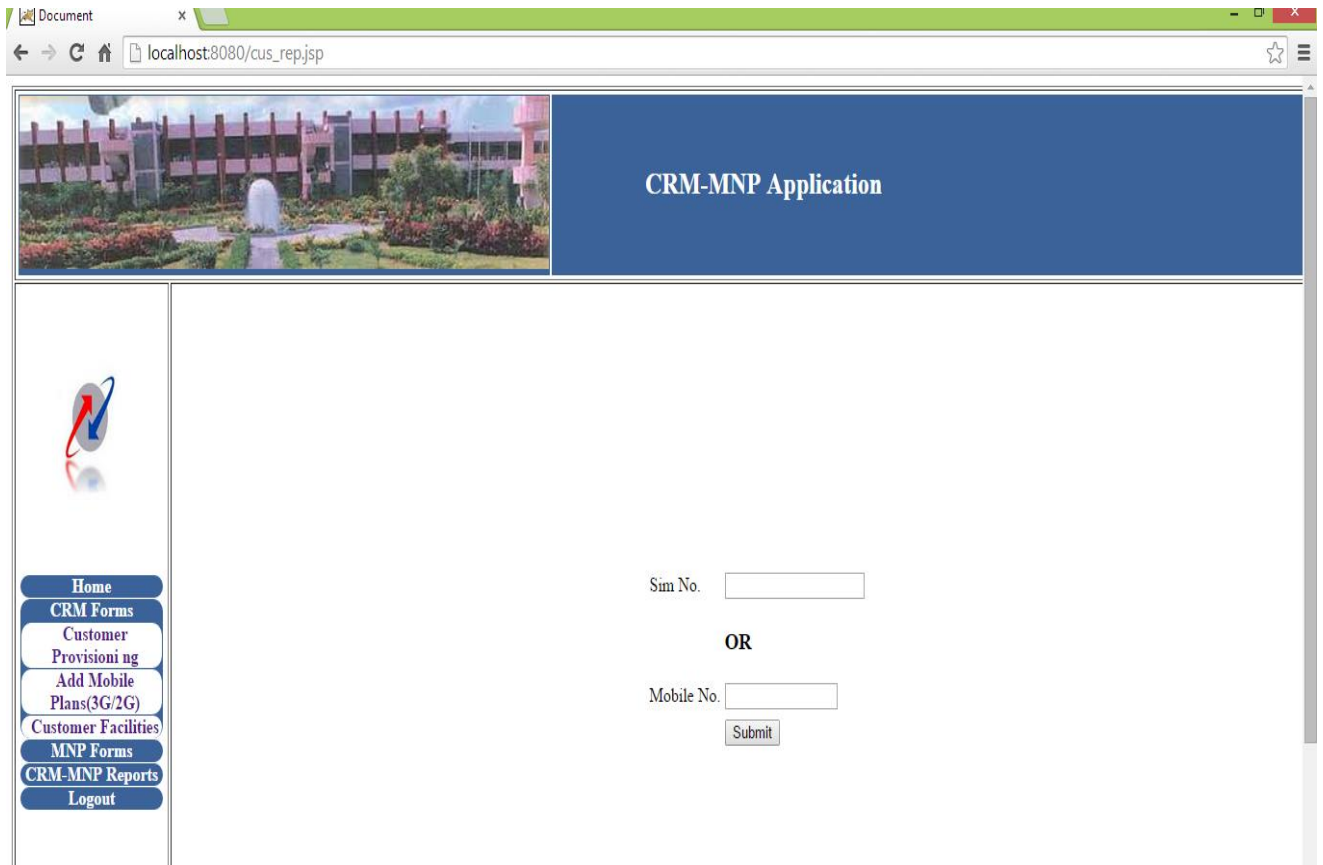


Figure 6.13:MNP report




## CRM REPORT:



Document x

localhost:8080/cus\_rep.jsp

CRM-MNP Application



Home  
CRM Forms  
Customer Provisioning  
Add Mobile Plans(3G/2G)  
Customer Facilities  
MNP Forms  
CRM-MNP Reports  
Logout

Sim No.

OR

Mobile No.

**Figure 6.14 CRM report**

### **6.3.2 Result Analysis**

In the system implementation phase, the software deals with the translation of the design specifications into the source code. The ultimate of the implementation is to write the source code and the internal documentation so that it can be verified easily. The code and the documentation should be written in a manner that eases debugging, testing and modification. System flow charts, sample run packages, sample output etc. is part of the implementation.

# CHAPTER 7

## **CHAPTER 7**

### **TESTING AND VALIDATION**

#### **7.1 Introduction**

Testing is one of the most important phases in the software development activity. In the software development life cycle (SDLC), the main aim of testing process is the quality; the developed software is tested against attaining the required functionality and performance.

During the testing process the software is worked with some particular test cases and the output of the test cases are analyzed for deciding whether the software is working according to the expectations or not.

The success of the testing process in determining the errors is mostly dependent on the test criteria. For testing any software we need to have a description of the system and method of determining whether the observed behavior confirmed to the expected behavior.

#### **7.2 Design of test cases and scenarios**

##### **7.2.1 Levels of testing**

Since the errors in the software can cause trouble at any stage, we have to carry out the testing process at different levels during the development. The basic levels of testing are Unit, Integration, System and Acceptance testing.

The Unit Testing is carried out on coding. Here different modules are tested against the specifications produced during design of the modules. In Case of integration testing, different tested modules are combined into sub systems and tested. In case of System testing, the full software is tested and in the next level of testing, the system is tested with user requirement document prepared during SRS.

## **7.3 Validation**

There are two basic approaches for testing. They are:

### **7.3.1 Functional testing:**

In Functional Testing, test cases are decided solely on the basis of requirements of the program or module and the internals of the program or modules are not considered for selection of test cases. This is also called Black-Box testing.

### **7.3.2 Structural testing:**

In Structural testing, test cases are generated on actual code of the program or module to be tested. This is called White Box testing. This project is tested and found out that results are correct. Finally this project produces the correct results.

**7.4 Testing cases for crm-mnp application:**

S.NO	TEST CASE	Expecting Behavior	Exhibiting Behavior	Result
1	Test accessing the CRM –MNP Web page	it should display the CRM-MNP application in a browser	Tested the path it display the output without any error.	passed
2	Testing of the login web page	Login the page with username and password it can go to the next page when it is wrong it displays the same web page	Tested the login page when operator entered username and password correctly it will shows home page	passed
3	Testing operators	when operators login to CRM application it shows welcome like(welcome:bsnl)	Tested the operator login	passed
4	Testing customer provision forms	When operator click the customer provision form it shows what are the details we need for filling the form	It is entered the customer details successfully	passed
5	Testing the mobile plan forms	It is creating the new plans in the networks	it created the plans	Passed
6	Testing the customer facilities	When the operator clicks customer facilities it required sim number of customer for operator to fill that form and what are the requirements customer needs	When giving the customer requirements that is allowed to the customer number	passed
7	Testing the MNP approval forms	When operator clicks into MNP APPROVAL FORM it porting the customer number	After testing mobile number is portable to the other network	passed

8	Testing the MNP report	it shows how many numbers are portable to other networks	It shows the portable of customer numbers	passed
9	Testing the CRM report	When operator entered CRM reports it shows customer details	It shows customer details	passed
10	Testing the logout	When operator logout the account it go for login page	After testing when operator logout the page it goes to login page	passed

## CHAPTER 8



## CHAPTER 8

### **Conclusion:**

The analysis of the project, which has Five modules is an implementation of a more efficient mobile number portability solution basing on an idea of creating a centralized database for all service providers and providing access to this database from within their CRM Web Portal, thereby reducing the time taken for porting a mobile number from one service provider to another within no time or in real-time within a few hours or even minutes.

Convenient reports also could be generated based on this data for both CRM and MNP Mobile Numbers.

### **Future scope:**

This project could be extended further to support IN Services and NGN services. The HLRs of each of the Mobile Service Providers may be in their own respective CRM databases for security reasons, but MNP numbers porting and approval should be maintained in a common MNP database for reducing the processing time from 2 or 3 days at present to within 1 Hour or even 30 minutes.

# APPENDIX

## **APPENDIX A: JAVA CONCEPT**

### **Introduction to java**

#### **About Java:**

Initially the language was called as “oak” but it was renamed as “java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent
- Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is to Internet Programming where c was to System Programming.

#### **Importance of Java to the Internet**

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Applications and applets. An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

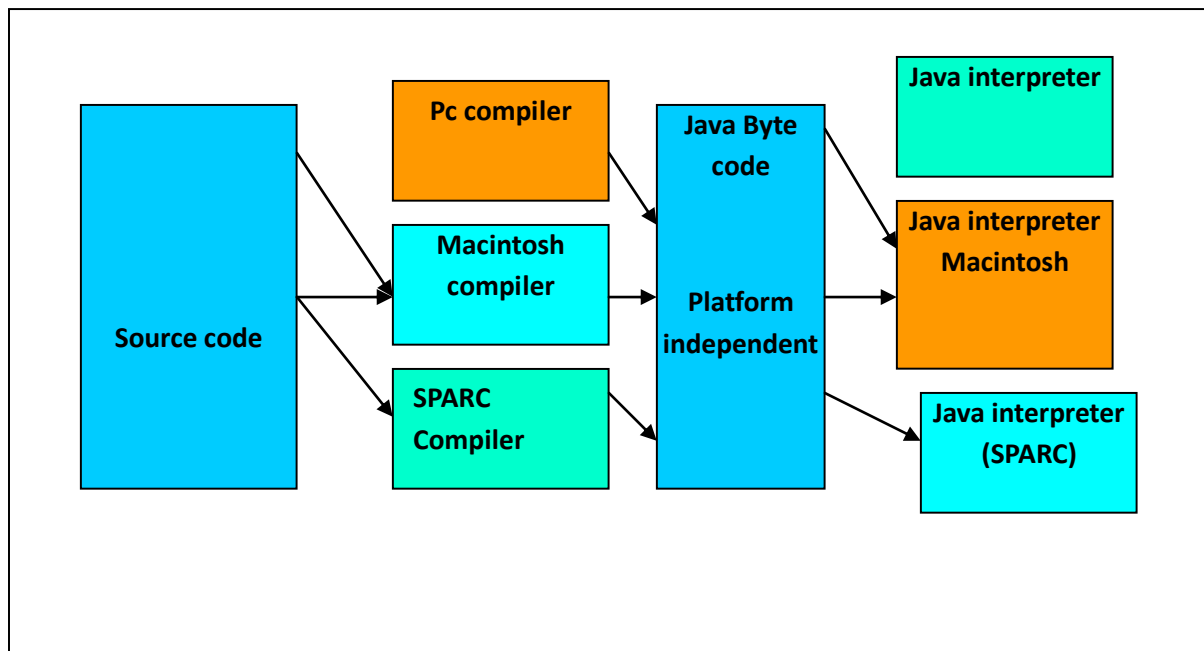
## Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

### Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

### Compiling and interpreting java source code.



### **Compilation of code**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

### **Simple:**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++ . Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

### **Object oriented**

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

### **Robust**

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

## **APPENDIX B:USER AUTHORIZATION**

### **User Authorization:**

**One way to perform session tracking is to leverage the information that comes with**

User authorization. When a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. After the client logs in, the username is available to a servlet through `getRemoteUser ()`

We can use the username to track the session. Once a user has logged in, the browser remembers her username and resends the name and password as the user views new pages on the site. A servlet can identify the user through her username and they're by Track her session.

The biggest advantage of using user authorization to perform session tracking is that it's easy to implement. Simply tell the protect a set of pages, and use `getRemoteUser()` to identify each client. Another advantage is that the technique works even when the user accesses your site from or exists her browser before coming back.

The biggest disadvantage of user authorization is that it requires each user to register for an account and then log in in each time the starts visiting your site. Most users will tolerate registering and logging in as a necessary evil when they are accessing sensitive information, but its all overkill for simple session tracking. Other problem with user authorization is that a user cannot simultaneously maintain more than one session at the same site.

### **Hidden Form Feilds:**

One way to support anonymous session tracking is to use hidden from feilds. As the name implies, these are feilds added to an HTML, form that are not displayed in the client's browser, they are sent back to the server when the form that contains them is submitted.

In a sense, hidden form feilds define constant variables for a form. To a servlet receiving a submitted form, there is no difference between a hidden fields and a visible filed.

As more and more information is associated with a client's session. It can become burdensome to pass it all using hidden form fields. In these situations it's possible to pass on just a unique session ID that identifies as particular clients session.

That session ID can be associated with complete information about its session that is stored on the server.

The advantage of hidden form fields is their ubiquity and support for anonymity. Hidden fields are supported in all the popular browsers, they demand on special server requirements, and they can be used with clients that haven't registered or logged in.

The major disadvantage with this technique, however is that works only for a sequence of dynamically generated forms, the technique breaks down immediately with static documents, emailed documents book marked documents and browser shutdowns.

### **URL Rewriting:**

URL rewriting is another way to support anonymous session tracking, With URL rewriting every local URL the user might click on is dynamically modified. Or rewritten, to include extra, information. The extra information can be in the deform of extra path information, added parameters, or some custom, server-specific.URL change. Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session.

Each rewriting technique has its own advantage and disadvantage

Using extra path information works on all servers, and it works as a target for forms that use both the Get and Post methods. It does not work well if the servlet has to use the extra path information as true path information

The advantages and disadvantages of URL.rewriting closely match those of hidden form fields,The major difference is that URL rewriting works for all dynamically created documents, such as the Help servlet, not just forms. With the right server support, custom URL rewriting can even work for static documents.

### **Persistent Cookies:**

A fourth technique to perform session tracking involves persistent cookies. A cookie is a bit of information. sent by a web server to a browser that can later be read back form that

browser. When a browser receives a cookie, it saves the cookie and there after sends the cookie back to the server each time it accesses a page on that server.

Persistent cookies offer an elegant, efficient easy way to implement session tracking. Cookies provide as automatic an introduction for each request as we could hope for. For each request, a cookie can automatically provide a client's session ID or perhaps a list of clients performance. The ability to customize cookies gives them extra power and versatility.

The biggest problem with cookies is that browsers don't always accept cookies sometimes this is because the browser doesn't support cookies. The browser doesn't support cookies. More often its because the user has specifically configured the browser to refuse cookies.

### **The power of serves:**

The power of servlets is nothing but the advantages of servlets over other approaches, which include portability, power, efficiency, endurance, safety elegance, integration, extensibility and flexibility.

### **Portability:**

As servlets are written in java and conform to a well defined and widely accepted API.they are highly portable across operating systems and across server implementation

We can develop a servlet on a Windows NT machine running the java web server and later deploy it effortlessly on a high-end UNIX server running apache. With servlets we can really "write once, serve everywhere"

Servlet portability is not the stumbling block it so often is with applets, for two reasons First,Servlet portability is not mandatory i.e. servlets has to work only on server machines that we are using for development and deployment

Second, servlets avoid the most error-prone and inconstancy implemented portion of the java languages.



**Power:**

Servlets can harness the full power of the core java. API's: such as Networking and Url access, multithreading, image manipulation, data compression, data base connectivity, internationalization, remote method invocation(RMI) CORBA connectivity, and object serialization, among others,

**Efficiency and Endurance:**

Servlet invocation is highly efficient, Once a servlet is loaded it generally remains in the server's memory as a single object instance, There after the server invokes the servlet to handle a request using a simple, light weighted method invocation .Unlike the CGI, there's no process to spawn or interpreter to invoke, so the servlet can begin handling the request almost immediately, Multiple, concurrent requests are handled the request almost immediately. Multiple, concurrent requests are handled by separate threads, so servlets are highly scalable.

Servlets in general are enduring objects. Because a servlets stays in the server's memory as a single object instance. It automatically maintains its state and can hold onto external resources, such as database connections.

**Safety:**

Servlets support safe programming practices on a number of levels.

As they are written in java, servlets inherit the strong type safety of the java language. In addition the servlet API is implemented to be type safe. Java's automatic garbage collection and lack of pointers mean that servlets are generally safe from memory management problems like dangling pointers invalid pointer references and memory leaks.

Servlets can handle errors safely, due to java's exception – handling mechanism. If a servlet divides by zero or performs some illegal operations, it throws an exception that can be safely caught and handled by the server.

A server can further protect itself from servlets through the use of java security manager.A server can execute its servlets under the watch of a strict security manager.

**Elegance:**

The elegance of the servlet code is striking .Servlet code is clean, object oriented modular and amazingly simple one reason for this simplicity is the served API itself. Which includes methods and classes to handle many of the routine chores of servlet development. Even advanced to operations like cookie handling and session tracking tracking are abstracted int convenient classes.

**Integration:**

Servlets are tightly integrated with the server. This integration allows a servlet to cooperate with the server in two ways. for e.g.: a servlet can use the server to translate file paths, perform logging, check authorization, perform MIME type mapping and in some cases even add users to the server's user database.

**Extensibility and Flexibility:**

The servlet API is designed to be easily extensible. As it stands today the API includes classes that are optimized for HTTP servlets. But later it can be extended and optimized for another type of servlets. It is also possible that its support for HTTP servlets could be further enhanced.

Servlets are also quite flexible, Sun also introduced java server pages. Which offer a way to write snippets of servlet code directly with in a static HTML page using syntax similar to Microsoft's Active server pages (ASP)

## **APPENDIX C:JDBC**

### **What is JDBC?**

Any relational database. One can write a single program using the JDBC API, and the JDBC is a Java Api for executing SQL, Statements(As a point of interest JDBC is trademarked name and is not an acronym; nevertheless, Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API

Using JDBC, it is easy to send SQL statements to virtually program will be able to send SQL .statements to the appropriate database. The Combination of Java and JDBC lets a programmer writes it once and run it anywhere.

### **What Does JDBC Do?**

**Simply put, JDBC makes it possible to do three things**

- Establish a connection with a database
- Send SQL statements
- Process the results
- JDBC Driver Types
- The JDBC drivers that we are aware of this time fit into one of four categories
- JDBC-ODBC Bridge plus ODBC driver
- Native-API party-java driver
- JDBC-Net pure java driver
- Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC, data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than

development and very limited development. JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavors. There are four driver categories

### **Type 01-JDBC-ODBC Bridge Driver**

Type 01 drivers use a bridge technology to connect a java client to an ODBC database service. Sun's JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

### **Type 02-Native-API party-java Driver**

Type 02 drivers wrap a thin layer of java around database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for c/c++ programmers, Because type-02 drivers are implemented using native code. in some cases they have better performance than their all-java counter parts. They add an element of risk, however, because a defect in a driver's native code section can crash the entire server

### **Type 03-Net-Protocol All-Java Driver**

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all java, which makes them useful for applet deployment and safe for servlet deployment

### **Type-04-native-protocol All-java Driver**

Type 04 drivers are the most direct of the lot. Written entirely in java, Type 04 drivers understand database-specific networking. protocols and can access the database directly without any additional software

## **JDBC-ODBC Bridge**

If possible use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge(that is, the Bridge native library, the ODBC driver manager library, library, the ODBC driver library, and the database client library)

### **WHAT IS The JDBC-ODBE Bridge ?**

The JDBC-ODBC Bridge is a Jdbc driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge is implemented as the sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is joint development of Intervolve and Java Soft

# **BIBILOGRAPHY**

## BIBILOGRAPHY

1. JAVA Server Pages ,\_SAMS, Tec media.
2. JavaScript **Ivan Bayross**
3. HTML Complete **Sybex Authors**
4. Dynamic HTML **Joseph Schmuller**
5. System Analysis & Design **Ilias M Awad**
6. Software Engineering A Practitioner's Approach: **Roger S Pressman**
7. <http://www.mysql.com/>
8. <http://specials.msn.co.in/ilit/>
9. <http://www.w3schools.com/>
10. <http://www.rttchyd.co.in>.