

## **Chapter-1: Introduction to JAVA**

### **1.1 Introduction**

**Java** is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Macintosh and Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics.

Much of the syntax of Java is the same as C and C++. One major difference is that Java does not have pointers. However, the biggest difference is that you must write object oriented code in Java. Procedural pieces of code can only be embedded in objects. In the following we assume that the reader has some familiarity with a programming language. In particular, some familiarity with the syntax of C/C++ is useful.

In Java we distinguish between applications, which are programs that perform the same functions as those written in other programming languages, and applets, which are programs that can be embedded in a Web page and accessed over the Internet. Our initial focus will be on writing applications. When a program is compiled, a byte code is produced that can be read and executed by any platform that can run Java.

#### **1.1.1 Characteristics:**

- Platform independent
- Object oriented
- Robust
- Multithreaded
- High performance
- Dynamic

### **1.1.2 Java is too effective:**

Java has caused more excitement than any development on the Internet since Mosaic. Java was the first way to include inline sound and animation in a web page. Java also lets users interact with a web page. Instead of just reading it and perhaps filling out a form, users can now play games, calculate spreadsheets, chat in real-time, get continuously updated data and much, much more.

Here are just a few of the many things Java can do for a web page:

- Inline sounds that play in real-time whenever a user loads a page
- Music that plays in the background on a page
- Cartoon style animations
- Real-time video
- Multiplayer interactive games

## **1.2 What makes Java special?**

Java is a programming language for distributed applications. It doesn't just allow you to add new types of content to your pages like Netscape and Internet Explorer do. Rather it lets you add both the content and the code necessary to interact with that content.

e.g. let's say you want to use EPS files on your Web site. Previously you had to wait until at least one web browser implemented EPS support. Now with the help of java you can write your own code to view EPS files and send it to any client that requests your page at the same time they request the EPS file.

E.g. suppose you want people to be able to search your electronic card catalog. However the card catalog database exists on a mainframe system that doesn't speak HTTP.

Before Java you could hope that some browser implemented your proprietary card catalog protocol or you could try to program some intermediate cgi-bin on a UNIX box that can speak HTTP and talk to the card catalog, not an easy task. With Java when a client wants to talk to your card catalog you can send them the code they need to do so. You don't have to try to force things through an httpd server on port 80 that were never meant to go through it.

If that were all Java was, it would still be more interesting than a <marquee> or <frame> tag in some new browser beta. Java is platform independent. A Java program can run equally well on any architecture that has a Java enabled browser. With the release of Netscape Navigator 2.0 that includes Windows 95, Windows NT, the MacOS, Sun Solaris, Sun OS 4.1.3, SGI IRIX, OSF/1, HP-UX with more to come.

Java isn't just for web sites. Java is a programming language that lets you do almost anything you can do with a traditional programming language like Fortran or C++. However Java has learned from the mistakes of its predecessors. It is considerably cleaner and easier to use than those languages.

### **Advantages of Java:**

- Simple
- Object-Oriented
- Platform Independent
- Safe
- High Performance

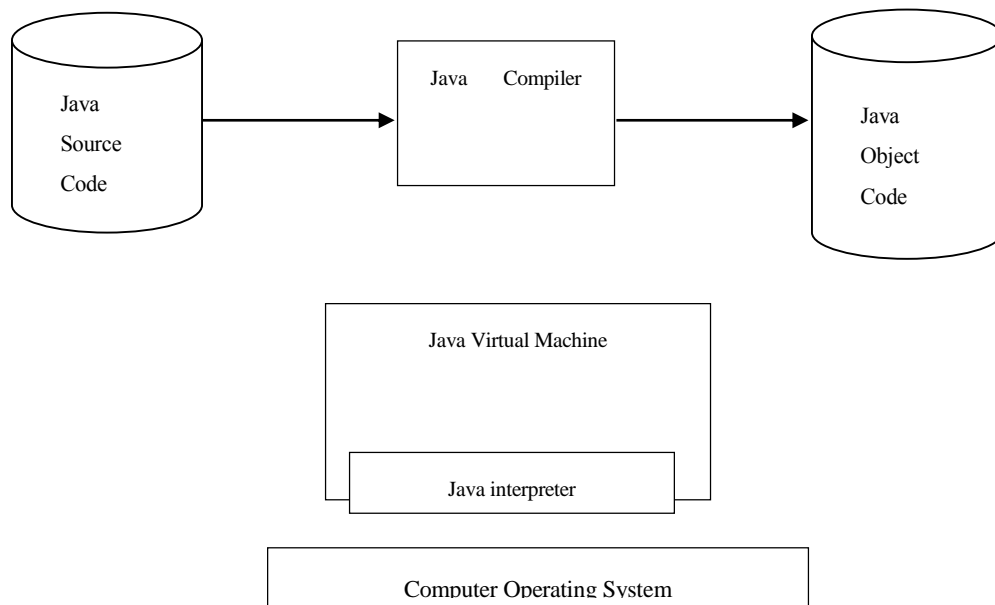
## 1.3 Java Platform

A java platform is the hardware or software environment in which a program runs.

The java platform has two components:

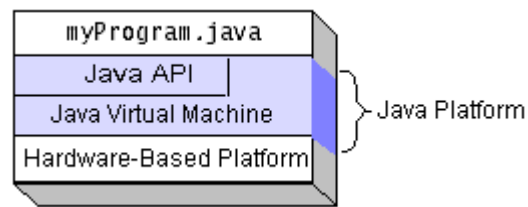
- Java Virtual Machine (JVM)
- Java Application Programming Interface (Java API)

JVM is a standardized hypothetical computer, which is emulated inside your computer by a program.



**Figure 1.3 Execution of program in Java**

The **Java API** is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI). The java API is grouped into libraries of related classes and interfaces these libraries are known as packages.



**Figure 1.3.2 Architecture of Java Platform**

## **1.4 Java Database Connectivity (JDBC)**

**JDBC** stands for "Java Database Connectivity". It is an API (Application Programming Interface) which consists of a set of Java classes, interfaces and exceptions and a specification to which both JDBC driver vendors and JDBC developers adhere when developing applications.

JDBC is a very popular data access standard. RDBMS (Relational Database Management Systems) or third-party vendors develop drivers which adhere to the JDBC specification. Other developers use these drivers to develop applications which access those databases.

The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.

JDBC helps you to write java applications that manage these three programming activities:

1. Connect to a data source, like a database.
2. Send queries and update statements to the database.
3. Retrieve and process the results received from the database in answer to your query.

## **Chapter-2: Introduction to MySQL**

**MySQL**, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL (R) software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

The MySQL software is Dual Licensed. Users can choose to use the MySQL software as an Open Source/Free Software product under the terms of the GNU General Public License or can purchase a standard commercial license.

### **➤ MySQL is a relational database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

### **➤ MySQL software is Open Source.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist.

1The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases much faster than

existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet

➤ **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

MySQL Server also provides an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

## **2.1 The Main Features of MySQL**

The following list describes some of the important characteristics of the MySQL Database Software.

### **1. Internals and Portability**

- Written in C and C++.
- Tested with a broad range of different compilers.
- For portability, uses CMake in MySQL 5.5 and up.
- Uses a very fast thread-based memory allocation system.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation in where no network is available.

## 2. Commands and Functions

- Full operator and function support in the SELECT list and WHERE clause of queries. For example:

```
mysql> SELECT CONCAT(first_name, ' ', last_name)
```

```
-> FROM citizen
```

```
-> WHERE income/dependents > 10000 AND age > 30;
```

- Full support for SQL GROUP BY and ORDER BY clause.
- Support for LEFT OUTER JOIN and RIGHT OUTER JOIN.
- Support for aliases on tables and columns as required by standard SQL.
- Support for DELETE, INSERT, REPLACE, and UPDATE to return number of rows that were changed, or to return the number of rows matched instead by setting a flag when connecting to the server.

## 3. Security

- A privilege and password system that is very flexible and secure, and allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.

## 4. Scalability and Limits

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table.



## 5. Connectivity

- Clients may connect to the MySQL server using TCP/IP sockets on any platform. On Windows systems in the NT family (NT, 2000, or XP), clients may connect using named pipes. On UNIX systems, clients may connect using UNIX domain socket files.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections. For example, you can use MS Access to connect to your MySQL server.
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or UNIX. Connector/J source is available.

## **Chapter-3: NetBeans IDE**

A free, open-source Integrated Development Environment for software developers.

One can get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java language, C/C++, and even dynamic languages such as PHP, JavaScript, Groovy, and Ruby.

NetBeans IDE is easy to install and use straight out of the box and runs on many platforms including Windows, Linux, Mac OS X and Solaris.

NetBeans IDE is an integrated development environment (IDE) for writing, compiling, testing, and debugging desktop applications and web applications for the Java platform.

NetBeans IDE includes a full-featured text editor with syntax highlighting and error checking, visual design tools, Ant support, version control system support, and many other features.

The NetBeans IDE 6.5 provides several new features and enhancements, such as rich PHP, JavaScript and Ajax editing features, improved support for using the Hibernate web framework and the Java Persistence API, and tighter GlassFish v3 and MySQL integration.

### **3.1 Netbeans – Developing Windows Application:**

- ✓ Netbeans can be used to create professional-looking GUIs with automatic spacing and alignment.

- **Professional Swing GUI Building**

Design Swing GUIs by dragging and positioning GUI components from a palette onto a canvas. The GUI builder automatically takes care of the correct spacing and alignment. Click into JLabels, JButtons, ButtonGroups, JTrees, JTextFields, ComboBoxes and edit their properties directly in place. You can use the GUI builder to prototype GUIs right in front of customers.

- **Intuitive and Customizable**

You can choose whether the GUI code should be generated with fully qualified or simple class names. The help bar displays context-sensitive hints about what can be done with the selected component and suggests shortcuts that can speed up your work in the future. The NetBeans IDE also comes with built-in support for GUI localization and accessibility. If you use JDK 6 Update 10, you can now preview your forms in the new Nimbus look and feel.

- **Standard and Custom GUI Components**

The extensible Component Palette comes with pre-installed Swing and AWT components and includes a visual menu designer. Use the Component Inspector to view a component's tree and properties.

## **Chapter-4: Introduction TO Project**

**CAM PRO – a Testing tool for IP Camera** will be a tool that will provide its users with a support for testing any IP (Internet Protocol) camera to check the format in which data is being sent by the IP Camera.

The data sent by IPCamera has two formats JPEG and MJPEG Using this tool users will be able to authenticate to any IP Camera, check the format of the data being sent by IP Camera and then process the data being sent to display the video to the user.

It will be developed using Java technologies.

### **4.1 Objective**

The **objective** of the system would be to:

1. Develop a tool to test IPCamera.
2. Develop a tool to display the data being sent from IPCamera to the user.
3. Maintain details of users who have privileges to use this tool

### **4.2 Business Functions**

1. **Login:** To access this application, users would be required to login through a login screen. After authentication user would be able to access the application according to his/her privileges.
2. **Users Maintenance:** That allows adding or updating the details of all users who are accessing this application.

3. **Testing tool for IPCamera:** This part of the application will provide a window interface which will be used to test the installed IPCamera for its configuration. It will also check if the live feed is being sent by the IPCamera or not.

### **4.3 Users**

The users of this system will be the users of the organization using this Testing Tool.

Following are the type of users:

- **Administrator:** Administrator will be the super user of the system having all the privileges. Administrator will be responsible to add, delete and assign privileges to the other users of the system.
- **Employee:** Employee is the member of staff of the organization using this application. Employee will be having limited privileges as assigned by the administrator. They will be responsible for testing of IPCamera.

### **4.4 Technologies to be used**

- **Java**
  - Java is a platform independent, object-oriented, robust, secure and innovative programming language.
  - It consist of Two Parts
    - JVM (Java Virtual Machine), which is software component that is used to execute java programs.
    - Java API (Application Programming Interface) that consist of

inbuilt classes that are used in different programs.

- **Database Programming**

- JDBC is Java Database Connectivity that is used for communication between database server and a java application. JDBC is a specification and API that is in programming database applications in java.

- **Windows Programming**

- Swings: Swings in java is a rich set of components for building GUIs and adding interactivity to java applications. Swing includes all the components that you would expect from a modern GUI toolkit that is table controls, list controls, tree controls, buttons and labels. The basic architecture of swing is MVC. And are entirely made in java.

## **4.5 Software Requirements**

- JDK 1.6
  - NetBeans 6.5.1
- JDBC Driver for MySQL Database Server
  - mysql-connector-java-5.1.7-bin.jar
- Database Server
  - MySQL Database Server 5.0
- Operating system
  - Windows Vista / XP sp3/ Linux Fedora 11

## **4.6 Hardware Requirements**

1. Intel P4 processor with minimum 2.0Ghz Speed
2. RAM: Minimum 512MB
3. Hard Disk: Minimum 20GB

## **4.7 Modules**

Following modules are to be developed:

- Login
- Admin
  - User Maintenance
- IPCamera Testing Tool.

## **Chapter-5: Literature Survey**

### **5.1 How to configure IP Camera**

#### **STEP 1: Configuring IP Camera**

There's several steps into making this work, so it's best to start at the camera. By default, most IP cameras are assigned a temporary IP address by your router. This is composed of 4 numbers from zero to 255 that are separated by periods that is how you find your camera. Then you use a program the vendor provided on a CD that finds the IP address assigned to the camera. You click on that and it brings up the in-camera software that lets you view and configure the camera. The problem with this is that it's likely that this address will change next time turn off and then back on the router or camera. The first step is to set a fixed address on the camera so that it doesn't change and you can always find it.

To do this, you will have to go into the configuration screen for your camera and set the option to have a fixed IP address and not use DHCP. Many times, unchecking DHCP will present you with the necessary fields for an IP address, subnet mask, gateway and DNS. This may be overwhelming. The easiest place to get this information is to go on a PC that is on the network and open a DOS window (on Windows, press the START button and then type "command" or on a Mac, run Terminal). In the DOS window, enter the command "ipconfig", on a MAC enter the command "ifconfig". You will see the IPv4 address, the subnet mask and the default gateway. Use this subnet mask (typically 255.255.255.0) and default gateway (the IP address of the router). For an IP address, use the first 3 sets of numbers and chose a higher number up to 253 for the last, for example, if it's 192.168.1.12, you can use an IP address like 192.168.1.100. Just make sure that you do not use the same number twice, it must be unique and not interfere with the lower number that are used by DHCP.



It will also give you the option of a port number. A port number is what is used to allow a single IP address to have multiple devices. The default is typically 80, but if you want to have more than one camera, you need to use a different number. To avoid port conflicts, it's better to use a number higher than 8000 but less than 60000. For example, if you have 3 cameras, you can use 10001, 10002 and 10003. From that point on, since port 80 is the default, you must specify the port number after the IP address separated by a colon when accessing the camera, for example if you used 10002, you will use <http://192.168.1.100:10002> to view the camera from your browser at home.

## **STEP 2: Port Forwarding**

Once you have a fixed IP address, the next step is to update your routers firewall with this information. The job of the router is to use a single connection to the internet and allow you to use it with multiple devices like cameras and computers. By default, the router has what is called a firewall and its job is to block anyone trying to get into your network including you. You have to define which devices can be accessed from the internet. You do this by going into the firewall configuration screen (sometimes called a Port Forward screen or Application & Gaming) and tell it which IP addresses can be accessed and what port number they are using. Some are as simple as putting in address 192.168.1.100 and port 10002 (start and end are typically the same). Some have separate screens where you define a service. A service is a port number assigned to a name, like Camera2 and port 10002, then in the other screen, you select Camera 2 and specify the IP address, 192.168.1.100. If it asks for protocol, it's TCP, but if you leave it at the default of both it doesn't matter.

### **STEP 3: Accessing IP Camera**

When you are home, you access the camera using what is called the LAN address. This can only either start with 192.168 or 10..., so you know you are using an internal address if it starts with this. To access the cameras remotely, you need its WAN address. This is the address that the router uses to connect to your cable or DSL service provider. Any router will have the ability to display this, but it's much easier to go any PC on your home network and go to <http://www.whatsmyip.org>. This will display your IP in large numbers on the top of the page. You should now test this, enter the IP and port and see if you can access the camera (that is <http://68.12.123.3:10002>). Some routers block the ability to loop back to your own network so it's possible that this won't work from home, so try it from a neighbor or friends network and see if you can connect. If it works you can use this to access your camera from anywhere in the world with one caveat, it may only last a few hours or at best a few days, so onto to the last step in the process.

### **STEP 4: Assigning Host Name**

Service providers have many, many customers and it would be very expensive to give each and every one of us a fixed IP address as you did with your camera in step 1. So they use DHCP and assign you temporary IP address and that can change. Some providers set this at once a day, some at once a week, it varies, but you can't rely on this address to be fixed for very long.

To get around this, there's a service called Dynamic DNS or DDNS. The way it works is a DDNS company assign you a host name, like [myhome.dyndns.com](http://myhome.dyndns.com) and then your router or camera has the ability to keep this up to date every time your service provider changes the WAN IP address. There are some large DDSN services that are free.

You create an account, pick a domain name from what is free (you can have your own domain name, but not for free), then pick a hostname which is the part of the name you can choose as long as someone else is not using it. Use only one hostname per location, so one for your entire home and maybe one for your vacation home or business.

The best place to maintain this is on your router. Check to see if it supports DDNS and most do. You'll have to specify the name you chose, your username and password and the DDNS provider you picked. If your router does not support this, many cameras do. ONLY DO THIS ON ONE DEVICE ON YOUR NETWORK.

Once this is set up, you can use this name and port address to visit each camera at home. For example, <http://myhomecams.dyndns.net:10001> for one camera and <http://myhomecams.dyndns.net:10002> for the other.

## **Chapter-6: Different Phase of Software Development**

### **6.1 Planning Phase**

#### **Problem Recognition**

A problem is well defined very rarely. It crops out with a vague feeling of some statements that leads to vague conclusions. So the first task is to get more crucial information by interviewing and meeting concerned people. It clarifies how the problem is felt, how often it occurs, how it affects the business and which departments are suffering with this. This phase consists of the following tasks.

#### **Problem Definition and Initial Investigation**

This was a preliminary investigation done with a view to have a “feel” of the working of the proposed system. This phase has been identified the end-user directly involved in the system who were the managers, assistant officer and database administrator, and the development department. By understanding the working of database, its flow and also after conducting meetings and interviews with the concerned persons of the department, a clear idea about the working was obtained. A flexible approach is adapted towards people who are interviewed. Short hand written notes are prepared based on the response of the employees. The interviews are preferably conducted at the work place of the person being interviewed. Detailed investigation is done in order to define the scope of the problem .The interview is concluded with a quick resume of the ground covered during the interview .The Questionnaire technique is combined with interviews to get the best result. Proper care has been taken in the design of such questionnaires so that the persons answering these questions do not feel hesitant. An explanatory note that serves to gain cooperation and avoid misunderstanding by setting out the purpose of the exercise clearly accomplishes each questionnaire.

***Observation technique*** is also used for fact finding. The work described at the time of interview is observed personally as it reduces the chances of misunderstanding and omissions. Some important things observed are like the flow of information through the system and important data transactions, the data being maintained and the frequency of their updating.

By the end of this phase, idea as to how the information enters the system, how it is stored, how it is processed, how information changes affects the working of the system and finally the output format required by the end-user was collected. All the information generated from this phase acted as an input to the next phase.

## **6.2 Analysis Phase**

There are a number of different approaches to system analysis. When a computer-based information system is developed, systems analysis would constitute the following steps:

- ✓ Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system.
- ✓ The development of a feasibility study, involving determining whether a project is economically, socially, technologically and organizationally feasible.

### **Detailed Study of the Existing System**

This phase provides the overall requirement for the system what is to be done. Input for this phase is the information collected through several data collecting schemes such as survey, cross-questioning-answering etc and the raw data obtained which is not properly ordered and not in the precise manner. So here this raw data is converted into useful information written in precise manner and thus output is a formal document. After collecting all the information and requirements, they were verified from the concerned persons by presenting a diagrammatic version of the proposed system. The points missing were added to the system specifications for the desired system. So this final document provides the system requirement specifications for the desired system. It helps in reducing the total development cost and also establishes the various points for validation and verification.

## **6.3 Design Phase**

After the analysis phase we have with us the details of the existing system and the requirements of the user for the new system. This phase diverts focus from the problem domain to the solution domain. It acts as a bridge between the requirement phase and its solution. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. Emphasis is on translating performance specifications into design specifications.

### ***6.3.1 The System Design***

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

### ***6.3.2 The External Design***

External design consists of conceiving, planning out and specifying the externally observable characteristics of the software product. These characteristics include user displays or user interface forms and the report formats, external data sources and the functional characteristics, performance requirements etc. External design begins during the analysis phase and continues into the design phase.

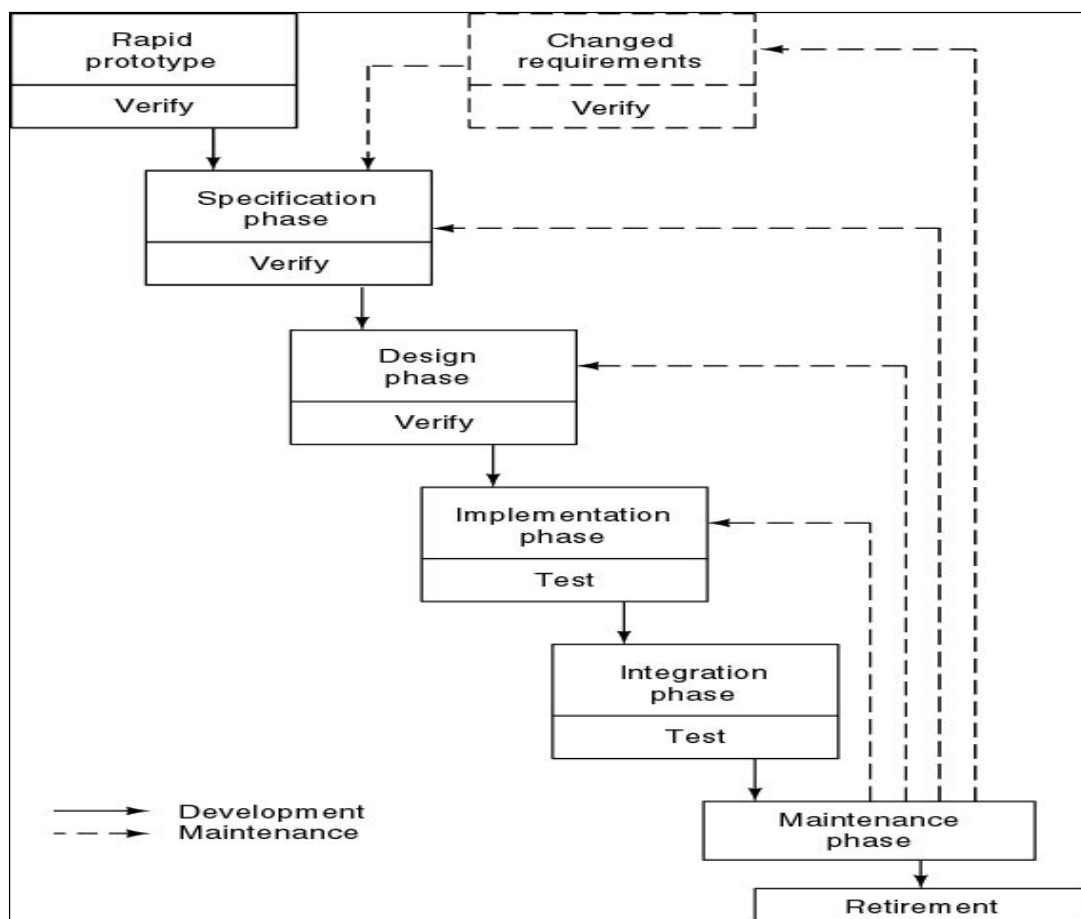
### ***6.3.3 Physical design***

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed as output.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

### **Design Methodology: Rapid Application Development (RAD)**

Rapid Application Development (RAD) is a methodology in which a systems designer produces prototypes for an end-user. The end-user reviews the prototype, and offers feedback on its suitability. This process is repeated until the end-user is satisfied with the final system.





## **Chapter-7: FRONT END**

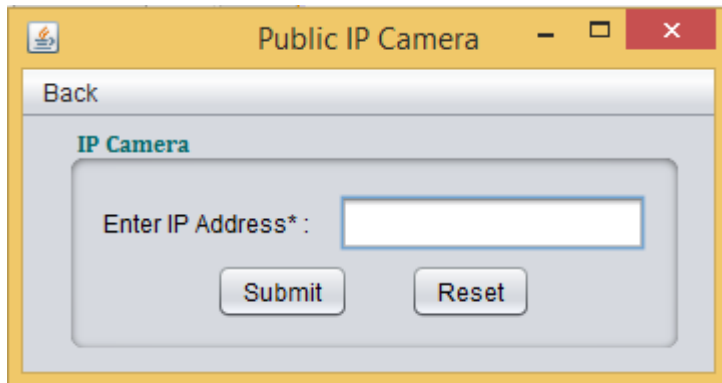
### **Screenshots:**

#### **Main**



**Figure 7.1 Main Display of Application**

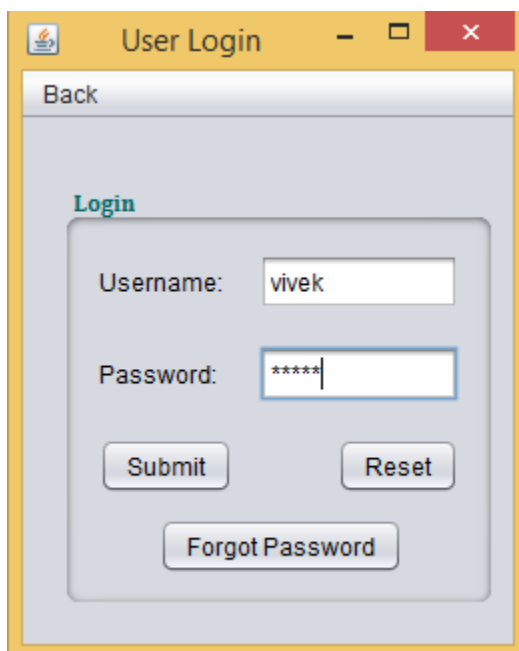
## Public IP Camera



The screenshot shows a web browser window titled "Public IP Camera". Inside the window, there is a "Back" button at the top left. Below it, the heading "IP Camera" is displayed. The main form area contains the text "Enter IP Address\*:" followed by a text input field. Below the input field are two buttons: "Submit" and "Reset".

**Figure 7.2 Public IP Camera**

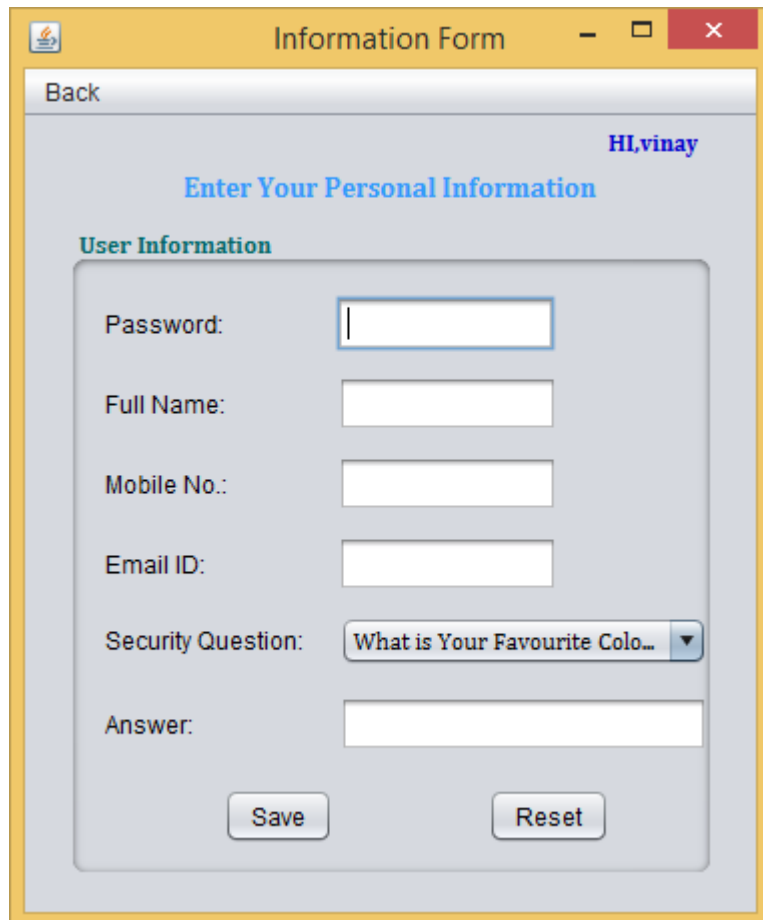
## Login screen



The screenshot shows a web browser window titled "User Login". Inside the window, there is a "Back" button at the top left. Below it, the heading "Login" is displayed. The main form area contains two labeled input fields: "Username:" with the text "vivek" and "Password:" with masked characters "\*\*\*\*\*". Below these fields are three buttons: "Submit", "Reset", and "Forgot Password".

**Figure 7.3 User Login Screen**

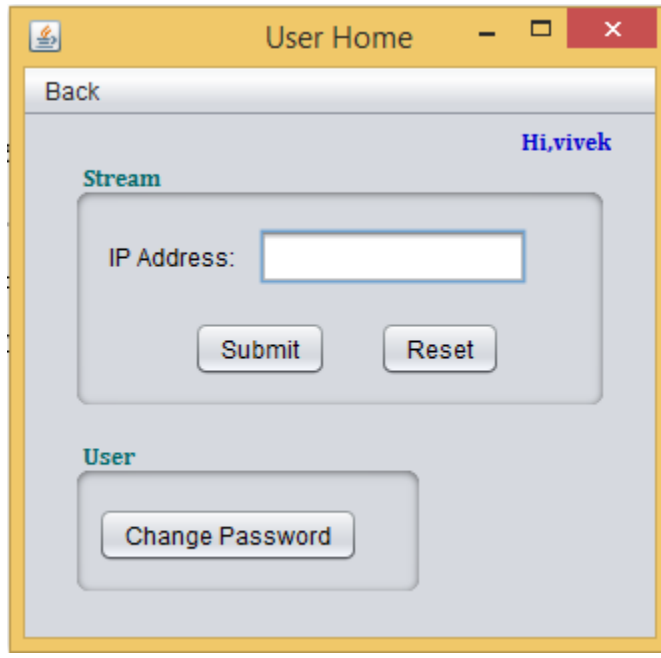
## Information Form of Added User



The screenshot shows a web application window titled "Information Form". At the top left is a "Back" button. At the top right, the text "HI,vinay" is displayed. Below this is the heading "Enter Your Personal Information". Underneath is a section titled "User Information" which contains several input fields: "Password:" with a text box, "Full Name:" with a text box, "Mobile No.:" with a text box, "Email ID:" with a text box, "Security Question:" with a dropdown menu showing "What is Your Favourite Colo...", and "Answer:" with a text box. At the bottom of the form are two buttons: "Save" and "Reset".

**Figure 7.4 Information Form of Added User**

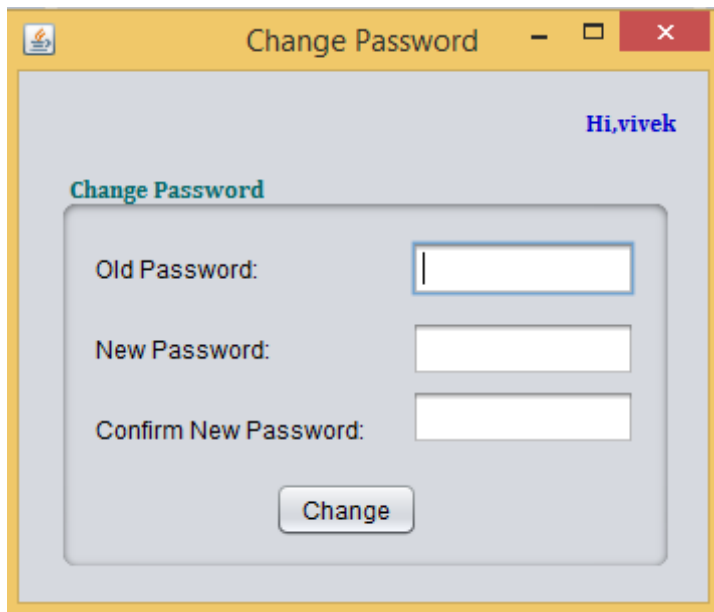
## User Login Home



The image shows a web application window titled "User Home". At the top left is a "Back" button. In the top right corner, it says "Hi,vivek". Below this, there is a section titled "Stream" which contains an "IP Address:" label followed by a text input field. Underneath the input field are two buttons: "Submit" and "Reset". Below the "Stream" section is another section titled "User" which contains a single button labeled "Change Password".

**Figure 7.5 User Login Home**

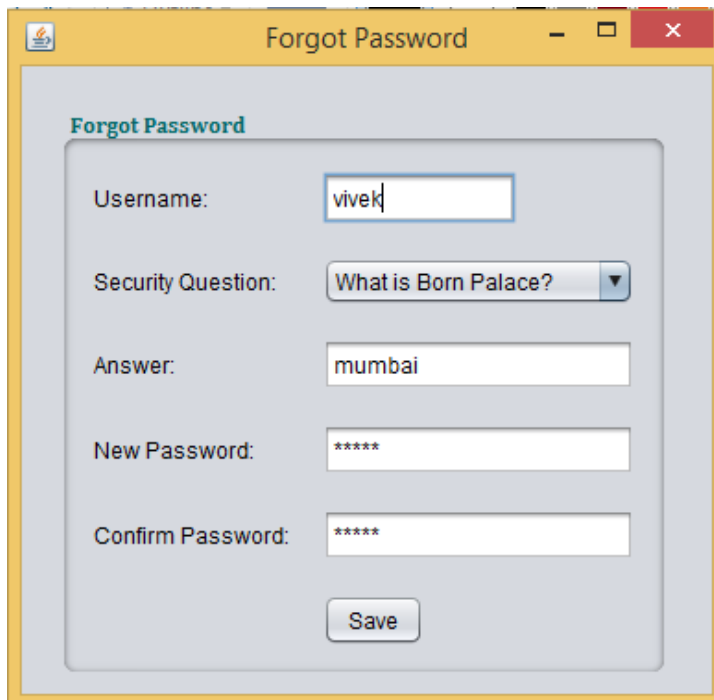
## Change Password User Login



The image shows a web application window titled "Change Password". In the top right corner, it says "Hi,vivek". Below this, there is a section titled "Change Password" which contains three labels with corresponding text input fields: "Old Password:", "New Password:", and "Confirm New Password:". At the bottom of this section is a button labeled "Change".

**Figure 7.6 Change Password Display**

## Forgot Password User Login

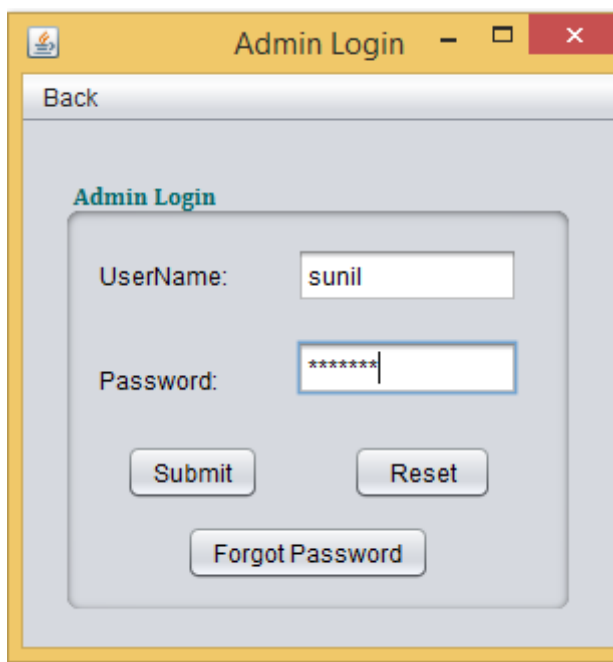


The screenshot shows a window titled "Forgot Password" with a yellow border. Inside, there is a form titled "Forgot Password" with the following fields and controls:

- Username:** A text input field containing the text "vivek".
- Security Question:** A dropdown menu showing "What is Born Palace?".
- Answer:** A text input field containing the text "mumbai".
- New Password:** A text input field containing six asterisks "\*\*\*\*\*".
- Confirm Password:** A text input field containing six asterisks "\*\*\*\*\*".
- Save:** A button located at the bottom of the form.

**Figure 7.7 Forgot Password Display**

## Admin Login

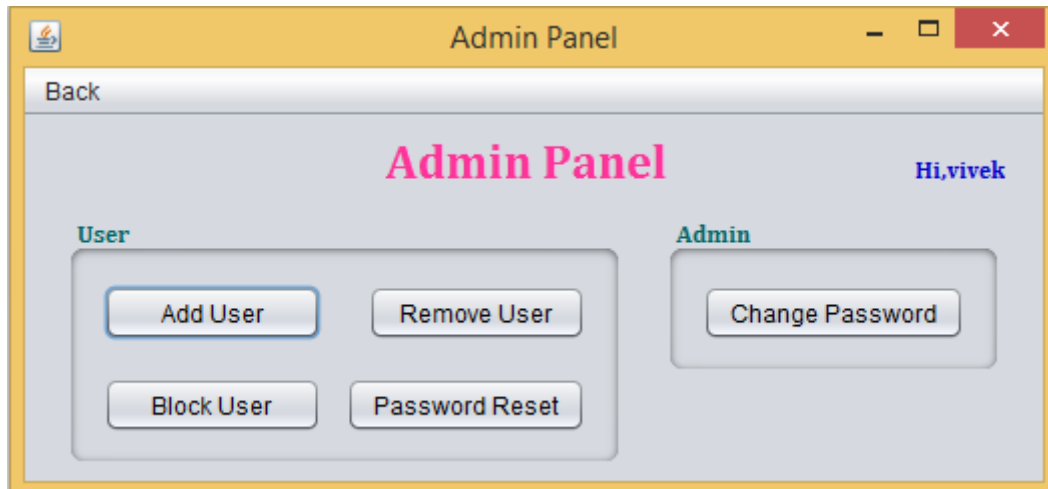


The screenshot shows a window titled "Admin Login" with a yellow border. Inside, there is a form titled "Admin Login" with the following fields and controls:

- Back:** A button at the top left of the form.
- UserName:** A text input field containing the text "sunil".
- Password:** A text input field containing seven asterisks "\*\*\*\*\*".
- Submit:** A button located below the password field.
- Reset:** A button located to the right of the Submit button.
- Forgot Password:** A button located at the bottom of the form.

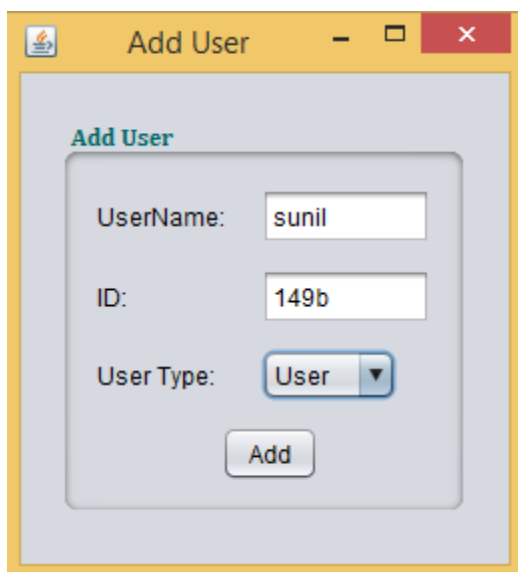
**Figure 7.8 Admin Login**

## Admin Panel



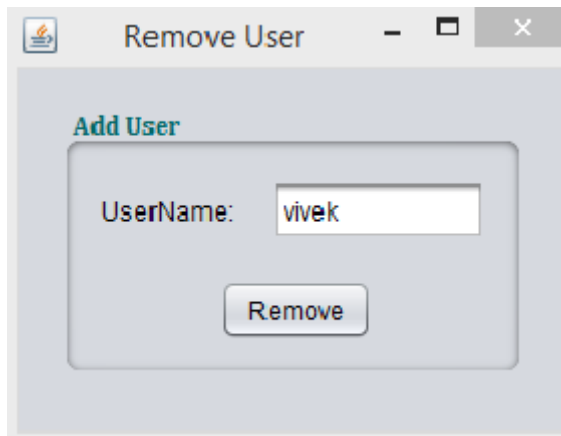
**Figure 7.9 Admin Panel**

## Add User (Admin Panel)



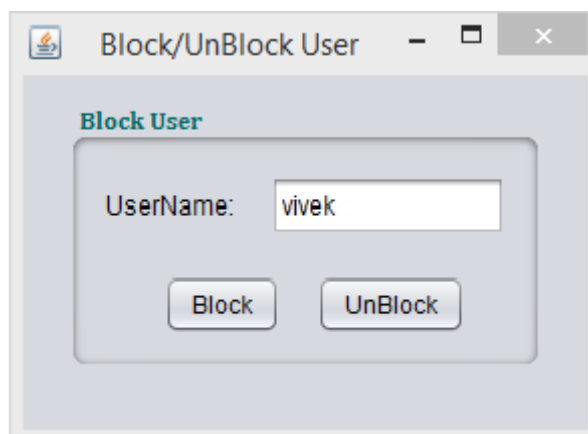
**Figure 7.10 Add User**

## [Remove User \(Admin Panel\)](#)



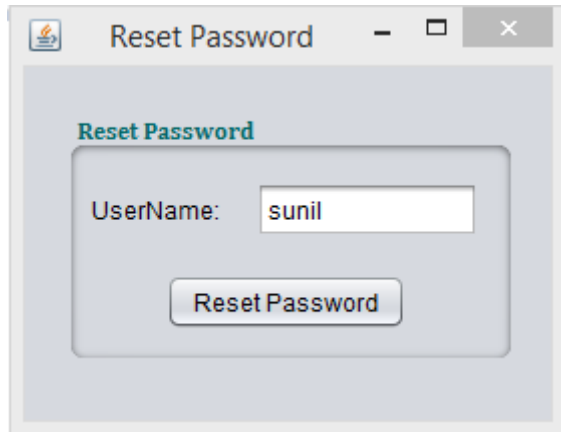
**Figure 7.11 Remove User**

## [Block/Unblock User \(Admin Panel\)](#)



**Figure 7.12 Block/Unblock User**

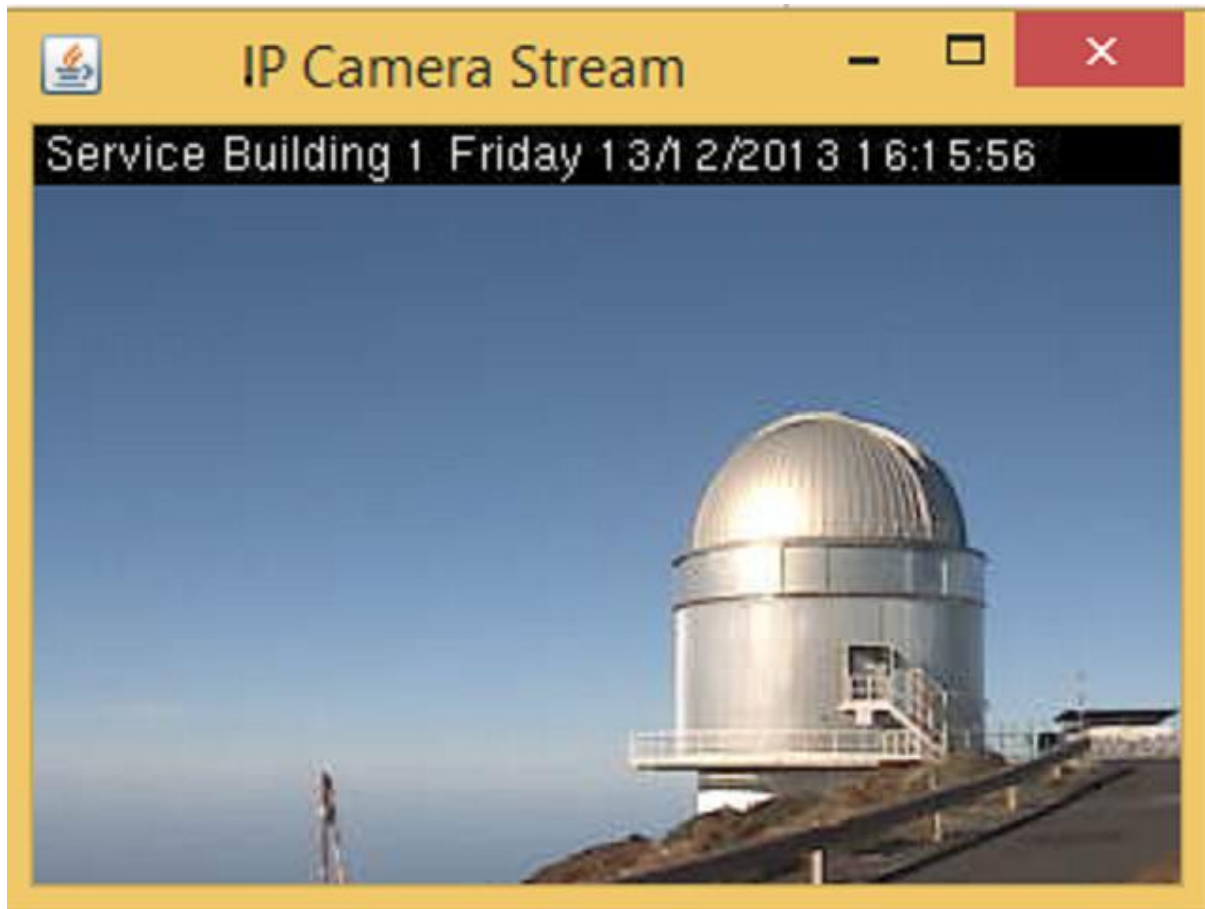
## Reset Password (Admin Panel)



**Figure 7.13 Reset Password**



## IP Camera Stream



**Figure 7.14 IP Camera Stream**

## **Chapter-8: Back End**

### **Program 1: Database Connection**

#### **Connection SQL.java**

```
package cam.pro;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Connection_SQL
{
    public static Connection connectId()
    {
        Connection con;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/cam_pro","root","");
            return con;
        }
        catch(ClassNotFoundException | SQLException e) {
            return null;
        }
    }
}
```

## **Program 2: IP Camera Stream**

### **Camera1.java**

```
package cam.pro;

import java.net.*;
import com.sun.image.codec.jpeg.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.swing.*;

public class Camera1 extends JPanel implements Runnable
{
    public boolean useMJPEGStream = true;
    public String mjpgURL=null;
    DataInputStream dis;
    private BufferedImage image=null;
    int counter=0;

    public Dimension imageSize = null;
    public boolean connected = false;
    private boolean initCompleted = false;
    HttpURLConnection huc=null;
    JMenuBar jm;
    JMenu j;
    Component parent;
```

```
public Camera1(Component parent_ , String st)
{
    parent = parent_;
    mjpgURL=st;
}

public static Image getScaledInstanceAWT(BufferedImage source, double factor)
{
    int w = (int) (source.getWidth() * factor);
    int h = (int) (source.getHeight() * factor);
    return source.getScaledInstance(w, h, Image.SCALE_SMOOTH);
}

public static BufferedImage toBufferedImage(Image image)
{
    new ImageIcon(image); //load image
    int w = image.getWidth(null);
    int h = image.getHeight(null);
    BufferedImage bimage = new BufferedImage(w, h,
BufferedImage.TYPE_BYTE_INDEXED);
    Graphics2D g = bimage.createGraphics();
    g.drawImage(image, 0, 0, null);
    g.dispose();
    return bimage;
}

public void connect()
{
    try
```

```
{
    URL u = new URL(mjpgURL);
    huc = (URLConnection) u.openConnection();
    System.out.println(huc.getInputStream());
    InputStream is = huc.getInputStream();
    connected = true;
    BufferedInputStream bis = new BufferedInputStream(is);
    dis= new DataInputStream(bis);
    if (!initCompleted) initDisplay();
}
catch(IOException e)
{
    try
    {
        huc.disconnect();
        Thread.sleep(1);
    }
    catch(InterruptedException ie)
    {
    }
}
catch(Exception e){;}
}

public void initDisplay()
{
    if (useMJPEGStream)
        readMJPEGStream();
}
```

```
        else
        {
            readJPG();
            disconnect();
        }
        imageSize = new Dimension(image.getWidth(this), image.getHeight(this));
        setPreferredSize(imageSize);
        parent.setSize(imageSize);
        parent.validate();
        initCompleted = true;
    }

    public void disconnect(){
        try
        {
            if(connected)
            {
                dis.close();
                connected = false;
            }
        }
        catch(Exception e){}
    }

    public void paint(Graphics g)
    {
        if (image != null)
            g.drawImage(image, 0, 0, this);
    }
}
```

```
}

public void readStream()
{ //the basic method to continuously read the stream
    try
    {
        if (useMJPEGStream)
        {
            while(true)
            {
                readMJPEGStream();
                parent.repaint();
            }
        }
        else
        {
            while(true)
            {
                connect();
                readJPG();
                parent.repaint();
                disconnect();
            }
        }
    }
    catch(Exception e){}
}
```

```
public void readMJPEGStream()
{
    readLine(4,dis); //discard the first 3 lines
    readJPG();
    readLine(1,dis); //discard the last two lines
}

public void readJPG()
{ //read the embedded jpeg image
    try
    {
        JPEGImageDecoder decoder = JPEGCodec.createJPEGDecoder(dis);
        image = decoder.decodeAsBufferedImage();
        char[] c=mjpgURL.toCharArray();
        String u="";
        for(int i=7;i<=c.length;i++)
        {
            if(c[i]!='/')
                u=u+c[i];
            else
                break;
        }
        String st="c:\\stream\\"+u+"\\";
        File dir = new File(st);
        dir.mkdir();
        String s="a"+counter+".jpg";
        counter++;
        String filename=st+s;
```



```
        System.out.println(filename);
        BufferedImage smaller = toBufferedImage(getScaledInstanceAWT(image,
1.0/3.0)); //scaling image size
        try
        {
            OutputStream out = new FileOutputStream(filename);
            JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);
            encoder.encode(smaller);
            out.close();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
    catch (Exception e)
    {
        e.printStackTrace(); disconnect();
    }
}

public void readLine(int n, DataInputStream dis)
{ //used to strip out the header lines
    for (int i=0; i<n;i++)
        readLine(dis);
}

public void readLine(DataInputStream dis)
```

```
{
    try
    {
        boolean end = false;
        String lineEnd = "\n"; //assumes that the end of the line is marked with this
        byte[] lineEndBytes = lineEnd.getBytes();
        byte[] byteBuf = new byte[lineEndBytes.length];
        while(!end)
        {
            dis.read(byteBuf,0,lineEndBytes.length);
            String t = new String(byteBuf);
            System.out.print(t); //uncomment if you want to see what the lines actually
look like
            if(t.equals(lineEnd))
                end=true;
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

public void run()
{
    System.out.println("in Run.....");
    connect();
    readStream();
}
```

```
public static void main(String[] args) {  
    }  
    public static void startCamera(String st)  
    {  
        JFrame jframe = new JFrame();  
        jframe.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);  
        Camera1 axPanel = new Camera1(jframe,st);  
        new Thread(axPanel).start();  
        jframe.setTitle("IP Camera Stream");  
        jframe.setLocation(520, 250);  
        jframe.getContentPane().add(axPanel);  
        jframe.pack();  
        jframe.show();  
    }  
}
```

## **Chapter-9: System Implementation and Testing**

### **8.1 Implementation Issues**

Implementation phase of the software development is concerned with translating the design specifications into the source code. After the system has been designed, arrives the stage of putting it into actual usage known as the implementation of the system. This involves putting up of actual practical usage of the theoretically designed system. The primary goal of implementation is to write the source code and the internal documentation so that conformance of the code to its specifications can easily be verified and so the debugging, modifications and testing are eased. This goal can be achieved by making the source code as clear and as straightforward as possible. Simplicity, Elegance and Clarity are the hallmarks of good programs whereas complexity are indications of inadequate design and misdirected thinking. The system implementation is a fairly complex and expensive task requiring numerous inter-dependent activities. It involves the effort of a number of groups of people: user and the programmers and the computer operating staff etc. This needs a proper planning to carry out the task successfully. Thus it involves the following activities:

- Writing and testing of programs individually
- Testing the system as a whole using the live data
- Training and Education of the users and supervisory staff

Source code clarity is enhance buy using structured coding techniques, by efficient coding style, by appropriate supporting documents, by efficient internal comments and by features provided in the modern programming language.

The following are the structured coding techniques:

- 1) Single Entry, Single Exit
- 2) Data Encapsulation
- 3) Using recursion for appropriate problems

## **9.2 Testing**

The most important activity at the implementation stage is the system testing with the objective of validating the system against the designed criteria. During the development cycle, user was involved in all the phases that are analysis, design and coding. After each phase the user was asked whether he was satisfied with the output and the desired rectification was done at the moment. During coding, generally bottom up technique is used. Firstly the lower level modules are coded and then they are integrated together. Thus before implementation, it involves the testing of the system. The testing phase involves testing first of separate parts of the system and then finally of the system as a whole. Each independent module is tested first and then the complete system is tested. This is the most important phase of the system development. The user carries out this testing and test data is also prepared by the user to check for all possible combinations of correct data as well as the wrong data that is trapped by the system. So the testing phase consists of the following steps:

- **Unit testing:**

In the bottom of coding technique, each module is tested individually. Firstly the module is tested with some test data that covers all the possible paths and then the actual data was fed to check for results.

- **Integration testing:**

After all the modules are ready and duly tested, these have to be integrated into the application. This integrated application was again tested first with the test data and then with the actual data.

- **Parallel testing:**

The third in the series of tests before handing over the system to the user is the parallel processing of the old and the new system. At this stage, complete and thorough testing is done and supports out the event that goes wrong. This provides the better practical support to the persons using the system for the first time who may be uncertain or even nervous using it.

***The testing will be performed considering the following points:***

- 1) Clerical procedure for collection and disposal of results
- 2) Flow of data within the organization
- 3) Accuracy of report output
- 4) Software testing which involves testing of all the programs together. This involves the testing of system software utilities being used and specifically develops application software.
- 5) Incomplete data formats
- 6) Halts due to various reasons and the restart procedures.
- 7) Range of items and incorrect formats
- 8) Invalid combination of data records.
- 9) Access control mechanism used to prevent unauthorized access to the system.

## **Conclusion**

This project that I undertook was truly a very rewarding experience for me in more than one way. It has given a big thrust to my technical knowledge as prospective Software professional. It has also helped me enhance my skills on the personal front.

And I feel extremely satisfied by the fact that I have managed to develop the project of course with equal contribution from my team members. I think I have exploited the opportunity that came my way to the fullest extent by increasing my technical know-how and also gaining the valuable work experience apart from studying the other subjects in our curriculum.

I gained an insight into the working of the window-based systems. This was even more rewarding as it was a typical real world situation.

## **References**

### **❖ Books Referred:**

---

1. The Complete Reference Java 2  
Herbert Schildt  
Tata McGraw-Hill
3. Pure JFC Swing  
Dr Satyaraj Pantham  
PHI
4. Professional Java Programming  
Brett Spell  
SPD

### **❖ Websites: -**

---

[Http://www.sun.java.com](http://www.sun.java.com)  
<https://jdic.dev.java.net/>  
<http://stackoverflow.com/questions/tagged/java>  
[http://en.wikipedia.org/wiki/IP\\_camera](http://en.wikipedia.org/wiki/IP_camera)