***Project Report on College Management System***

Submitted in partial fulfillment of the requirements

for the award of the diploma in Computer Science



**GOVERNMENT POLYTECHNIC COLLEGE, AMRITSAR**

Submitted To: Submitted By:

S. Harpreet Singh Soch Viresh Goyal

100186206583

Raju

100186206567

Bharat Rana

100186206578

Anand Sehgal

100186206529

ACKNOWLEDGEMENT

**ABSTRACT**

**The project is running with the help of Socket Programming with the help of Java. MySQL server is the back-end tool used in this project. I like to think of a programmer as a bit like a plumber! A plumber will arrive at a job with a big bag of tools and spare parts. Having looked at it for a while, he will open his bag and produce various tools and parts, fit them all together and solve your problem. Programming is just like this. You are given a problem to solve. You have at**

**Your disposal a big bag of tricks, in this case a programming language. You look at the problem for a while and work out how to solve it and then fit the bits of the language together to solve the problem you have got. The art of programming is knowing which bits you need to take out of your bag of tricks to solve each part of the problem. There are many things you must consider when writing a program; not all of them are directly related to the problem in hand. I am going to start on the basis that you are writing your programs for a customer. He or she has problem and would like you to write a program to solve it. Coming up with a perfect solution to a problem the customer has not got is something which happens surprisingly often in the real world. Many software projects have failed because the problem that they solved was the wrong one. The developers of the system quite simply did not find out what was required, but instead created what they thought was required.**

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Particulars** | **Page No.** |
| **1** | Chapter 1 – Introduction   * Introduction to the project * System overview * Modules of the project | **6-13** |
| **2** | Chapter 2 – Objective of Project Work   * Objectives * Functional Requirements * Performance Requirements * Design Constraints * Activity Diagrams * E-R Diagrams * Class Diagrams * Flowcharts | **14-23** |
| **3** | Chapter 3 – Literature Review   * Introduction * Software used * Packages used * Different scenarios related to client-server | **24-31** |
| **4** | Chapter 4 – Present Work   * About Project * Hardware used * Development tools to be used * Snapshots related to the project | **32-56** |
| **5** | Chapter 5 – Conclusions and future scope   * Conclusion * Further Scope of Project in Future * Maintenance | **57-59** |

Chapter 1- Introduction

* **Introduction to the Project**
* **Name of Project**
* **System Overview**
* **EXISTING SYSTEM**
* **PROPOSED SYSTEM**
* **EXISTING SYSTEM:**

**Disadvantages:**

* **PROPOSED SYSTEM:**

**Advantages:**

* **Modules of the Project**

Chapter 2- Objective of Project Work

**Objectives:**

**Functional Requirements: -**

**Performance Requirements: -**

**Design Constraints: -**

1. **Standards Compliance: -** The basic standard for the proposed system is that the Server should never crash under any circumstances.
2. **Hardware Limitation: -** There is no hardware limitation to be specified.
3. **Reliability and Fault Tolerance: -** The proposed system should be highly reliable and must be fault tolerant. Probably there will not be any fault, and if any fault occurs then better to restart the Server.
4. **Security: -** The existing system is highly secured, and no further security is needed for proposed system.

**Activity Diagrams: -**

Activity Diagram shows the overall functionality of the system.

1. **Activity Diagram for Connecting to the Server.**

Server

Client 1

Client 2

IP Address Update the List

1. Activity Diagram for getting the information of a client (Identity of a client).

EXIT

The Information

Is displayed as

Tool tip.

Place the Hand Cursor on any client’s IP

1. Activity Diagram for transferring file.

Prompt that

Path Not Found.

Receives Acknowledgement

Enter Source & Destination Paths

Choose Multiple

Clients

Choose a Single

Client

If path exists If path doesn’t exist

EXIT

4. Activity Diagram for Disconnecting (Active/Passive mode) clients.

EXIT

Select the kill option to Disconnect

Choose Multiple

Clients

Choose a Single

Client

## Class Diagrams: - The things naturally fall into categories we refer to those categories as classes. A class is a category or group of things that have similar attributes and common behavior.

1. Class Diagram for a Client.

: Client

+ Socket Instance

+ Input Stream

+ Output Stream

+ Host Address

+ Connected Hosts

+ addLabel( )

2. Class Diagram for Server.

: Server

+ Socket Instance

+ Input Stream

+ Output Stream

+ Number of

Client

+ Connected Hosts

+ addClient( )

Chapter 3- Literature Review

* **Introduction**

This Package has been developed in **Java**, which is based on Object Oriented Methodology. There are several packages resided in Java and to develop this system, mainly networking, swing packages of Java are used. Various features of Java made it a first choice of the programmers. Java is a platform independent language, which can be run under any kind of environment.

The main features involve,

1. **Simple**: - Java is a language which is based on Object Oriented Methodology, so it is very easy to learn and can be used effectively.
2. **Robust**: - Java Programs are said to be robust because they will take care of memory management and will never crash under any circumstances.
3. **Secure**: - Even though Java is developed using Object Oriented Principles, it eliminated the Pointers Concept. So it is not possible to access memory directly, that’s why Java is said to be Secure and is applicable for Internet, for that Applet is designed which can be understandable by the browsers.
4. **Portable**: - Java Programs are Portable that those can be run under any kind of environment irrespective of the hardware used. This is known as platform independent.
5. **Compiled & Interpreted**: - Unlike remaining Programming languages Java code is both Compiled and Interpreted. The output after compilation is ‘Byte Code’ which is interpreted to produce output. This Byte Code is a new evolution, which makes Java a Platform Independent Language.

# Software Used: - Some of the features of Java that are used in this system include,

# Multithreaded programming that allows running 2 or more processes concurrently.

The Server program is implemented using this concept so that it will always be in listening state.

**Packages Used: -**

1. The default package to build any program in Java is **lang** that is used in developing this system
2. The API to build Graphical User Interface in Java is provided using the packages **awt** and **swing**. This system uses both the packages to provide GUI based forms.
3. To provide the communication over the network, the package net is used to develop the Server and Clients.
4. In Java JDBC (Java Database Connectivity) is used to connect to the database, a special package known as **sql** is developed for that.
5. The **util** package in Java is an enhancement of the existing classes. This package is also used in this system for declaring arrays (Vector), and to manipulate the strings.
6. The streams that are used to send and receive the data over network are also utilized in this system, which are in the package of **io**.
7. **Use case & Scenario for the need to connect to the Server.**

<<Include>>

Initiator(user) Initiator(user)

**Precondition: -** Here the Initiator is some personnel in the department, who wants to send a file or a message to some other client.

**Actor Initializing: -** A department personnel (One client)

Steps: - **1.** The User connects to the Server to send a file or a message.

**2.** The User selects another client to whom he wanted to send

a file or message.

**3.** Exit.

**Post condition: -** The user gets connected to the Server.

1. **Use case & Scenario to transferring file to single/multiple client(s).**

<<Include>>

Initiator(user) Initiator(user)

**Precondition: -** Here the Initiator is the user, who wants to send a file to a single client or multiple clients.

**Actor Initializing: -** A department personnel (One client)

Steps: - **1.** The user first connects to the Server.

**2.** The user can select a single client or multiple clients to transfer

a file simultaneously.

**3.** Exit

**Post condition: -** The user gets the acknowledgement that the file has been reached.

1. **Use case & Scenario for specifying the paths.**

<<Include>>

Initiator Initiator

(User) (User)

**Precondition: -** Here the Initiator is a client, who wants to send a file.

**Actor Initializing: -** A department personnel (One client).

Steps: - **1.** The User connects to the Server to send a file or a message.

**2.** The User selects another client to whom he wanted to send a file.

**3.** The User has to give the source path and the destination path

Correctly.

**4.** Exit.

**Post condition: -** The Specified file will be transferred to the specified location if exists, in the remote system, if not ‘Path not found’ message has to be displayed.

1. **Use case & Scenario to disconnect a single/multiple client(s).**

<<Include>>

Initiator Initiator

(User) (User)

**Precondition: -** Here the Initiator is the user, who wants to disconnect a single or multiple clients.

**Actor Initializing: -** A department personnel (One client).

Steps: - **1.** The user first connects to the Server.

**2.** The user can select a single client or multiple clients to disconnect

at a time.

**3.** Exit.

**Post condition: -** The user’s presently connected clients list will be updated.

Chapter 4- Present Work

**ABOUT PROJECT**

Project can communicate various client to a server for data shearing . There are many clients and only one server because server side contain server socket and simple socket so, server accept connection pass to the simple socket then the sever socket is always free that are se to accept connection .

In client side only simple socket are used so server contain many client due to server socket but that clients are connection establish by server through login a new client .after that client contain his private key password. Login through that password are client are logically and physicaly connected to server communicate for data shearing between them the server shear the folder dfs to client .the client download and upload the data for client to server and server to client this process is called upload and download.

They still expected something advance to come up with all those features which were definitely the cause of worry to them with respect to the security of their code and this thought gave birth to a revolution which discovered another PROGRAMMING LANGUAGE-JAVA with the features to ensure the security and the portability of their programs. Developing your applications using the JAVA programming language results in software that is portable across multiple machine architectures, operating systems, and graphical user interfaces, secure, and high performance.

Project of client server are developed by using Java language is a highly robust, distributed, high performance, object-oriented, multi-threaded language with all of the usual features. As such, it builds upon years of C++ development, taking the good and dispensing with the bad. login through that password are client are logically and physically connected to server communicate for data shearing between them the server shear the folder dfs to client .the client download and upload the data for client to server and server to client this process is called upload and download .

**Hardware Used: -**

**Server side: -** The Server program must be as fast as possible to keep track of all the connected clients, and the memory must be as much as sufficient. The Systems which were kept as Oracle Server and NMS Server are having high speed processor and a large capacity of RAM (Random Access Memory). The Server program has to be placed in the NMS Server that is very fast and can handle any number clients.

Some of the configuration is listed below,

Console: - Mouse, 15” Monitor, 104keys Keyboard

Processor: - Intel Pentium IV processor, 1.9 GHz speed.

Memory: - 256MB RAM, 40GB Hard Disk.

Cache: - 56MB.

Ports: - Serial Ports, Parallel Ports (LPT-1, LPT-2)

N/w Components: - Network Adapter, RJ-45 Connector, HUB.

**Client side: -** The Client program doesn’t need high-speed processor, RAM like Server. The minimum configuration is needed to install the Client program.

Some of the configuration is listed below,

Console: - Mouse, 15” Monitor, 104keys Keyboard

Processor: - Intel Pentium III/IV processor, 1.1 GHz speed.

Memory: - 64/128 MB RAM, 40GB Hard Disk.

Cache: - 56MB.

Ports: - Serial Ports, Parallel Ports (LPT-1, LPT-2)

N/w Components: - Network Adapter, RJ-45 Connector, UTP

Cable, HUB.

* **Development Tools To Be Used**
* **Front-End Tool**
* **Java:**
* Java was conceived by James gosling, Patric naughton, Chris warth, Ed frank, and Mike sheriden at Sun Microsystem, inc in 1991. This language was initially called “oak” but was renamed “Java” in 1995, between the initial implementation of oak in the fall of 1992 and public announcement of java in the spring of 1995, many more people contributed to the design and evolution of the language. Bill Joy, Arthur van hoff, jonathan payne, frank yellin, and tim lindholm were key contribution to the maturing of the original prototype.
* Somewhere surprisingly, the original impetus for java was not the INTERNET ! Prototype. The primary motivation was to need for a platform-independent (that is architecture –natural) language that could be used to create software to be embedded in various consumer electronic devices such as microwave ovens and remote control as you can probably guess many different types of CPU are used as controllers the trouble with C and C++ (and most other language) is that they are designed to be compiled for a specific target. Although it is possible to compile a C++ program for just about any type of CPU, to do so requires a full compiler targeted for that CPU.
* Java is a programming language and environment invented by James Gosling in 1994. Gosling was the first designer of the Java programming language and implemented its original compiler and virtual machine.
* Java is the first and foremost programming Language. Creation of Java was driven by both elements in nearly equal measures which are:
* 1.        To adapt to changing environments and uses.
* 2.        To implement refinements and improvements in the art of programming.
* "Java: A simple, object-oriented, distributed, interpreted, robust, secure, portable, high-performance, multithreaded, and dynamic language" –Java Soft
* A Java Program: From Birth to Execution
* 1.        Coding: Human-readable Java code is produced by the programmer
* 2.        Building: A Java Development Tool "build’s the Java program into byte-code, which is saved as a ". class" file.
* 3.        Loading: Via the web or command line, the class file is sent to the Java Virtual Machine (VM) with an attached digital signature. The Java VM is simply an interpreter.
* 4.        Byte code Verification: The Java VM verifies the digital signature. When downloaded remotely, the Java VM isolates the Java program in a restricted part of memory. The Java program is not allowed to access local Hard drives and System resources.
* 5.        Internal Integrity: Verification checks are made to insure that the loaded Java program is well formed. Data types are verified along with other syntax structure
* 6.        Execution: Program execution begins
* Java Project
* In 1990, Sun Microsystems began a project called Green to develop software for consumer electronics. Gosling began writing software in C++ for embedding into such items as toasters, VCR's, and Personal Digital Assistants ( PDA's ). The embedded software makes more appliances more intelligent. Gosling's solution to the problem of C++ was a new language called Oak. Finally in 1995, Oak was renamed Java. Since then, Java is rising in popularity. Java Soft also sued Microsoft, for violating its Java license agreement. Microsoft wants to add Windows specific alterations to the Java language, which would blunt the "run anywhere" goal of Java.
* Java Soft, which presents compatibility problems with existing web browsers and Virtual Machines are currently expanding Java. Its syntax is similar to C and C++, but it omits many of the features that make C and C++ complex, confusing, and unsafe. The Java platform was initially developed to address the problems of building software for networked consumer devices. It was designed to support multiple host architectures and to allow secure delivery of software components. To meet these requirements, compiled code had to survive transport across networks, operate on any client, and assure the client that it was safe to run.
* The popularization of the World Wide Web made these attributes much more interesting. The Internet demonstrated how media-rich content could be made accessible in simple ways. Web browsers enabled millions of people to roam the Net and made Web surfing part of popular culture. At last there was a medium where what you saw and heard was essentially the same whether you were using a Mac, PC, or UNIX machine, and whether you were connected to a high-speed network or a slow modem.
* Web enthusiasts soon discovered that the content supported by the Web's HTML document format was too limited. HTML extensions, such as forms, only highlighted those limitations, while making it clear that no browser could include all the features users wanted. Extensibility was the answer.
* Sun's Hot Java browser showcases the interesting properties of the Java programming language and platform by making it possible to embed programs inside HTML pages.
* These programs are transparently downloaded into the Hot Java browser along with the HTML pages in which they appear. Before being accepted by the browser, the programs are carefully checked to make sure they are safe. Like HTML pages, compiled programs are network- and host-independent. The programs behave the same way regardless of where they come from or what kind of machine they are being loaded into and run on.
* Visitors to Web pages incorporating dynamic content can be assured that their machines cannot be damaged by that content. Programmers can write a program once, and it will run on any machine supplying a Java or Java 2 run time environment.
* The Java language is very secure and platform independent when compared to alternative languages. Java's secret is the tightly integrated language model.
* Java features
* Simple
* Java was designed to be easy for the professional programmer. It is easy to learn and can be used effectively. If you are an experienced C++ programmer, moving to Java will require very little effort.
* Secure
* There is a concept of  applets in Java which can be downloaded without fear or virus or malicious content, because the Java programs are confined to Java execution environment and are not allowed to access other parts of the CPU.
* Portable
* The Java programs called Applets run in the JVM (Java virtual machine) environment that is in every browser therefore the programs can run anywhere.
* Object Oriented
* Java Classes follow the Oops concept of encapsulation, inheritance, and polymorphism.
* Robust
* Garbage collection and Exception handling make Java a robust language. In garbage collection the user doesn’t have to bother about the memory allocation as, when the object is no longer in use it is automatically deleted to release memory space.
* Multithreaded
* A single threaded application has one thread of execution running at all times and such programs can do only one task a time.
* A multi-threaded application can have several threads of execution running independently and simultaneously. These threads may communicate and cooperate and will appear to be a single program to the user.
* Interpreted
* The Java code is compiled into the byte code, which is the class file. The byte code is then interpreted to the machine language by the JVM environment.
* Distributed
* Java handles the TCP/IP protocols, which makes it easier to use in Internet.
* Some Other Features Of Java Programming
* Encapsulation
* ENCAPSULATION is the mechanisms that binds together code and the data it manipulates, and keeps both safe outside interference and misuse. It is a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside  the wrapper.
* Encapsulation is the capability to represent, denote and handle information at a higher level that is inherent to a computer or base language. Variables and methods are formerly known as instance variables and instance methods to distinguish from class variables and class methods.
* Inheritance
* Inheritance is the process by which one object acquires the properties of another object.
* Classes inherit state and behavior from their superclass. A class is a blueprint or prototype that defines the variables and methods common to all objects of a certain kind.
* Object oriented systems allow classes to be defined in terms of other classes. For example, mountain bikes, racing bikes and tandems are all subclasses of  the bicycle class. Similarly, the bicycle class is the superclass of mountain bikes, racing bikes and tandems.
* Each subclass inherits state (in the form of variable declarations ) from the superclass. Mountain bikes, racing bikes and tandems share some states : Cadence, speed, and the like. Also each subclass inherits methods from the superclass.
* Benefits of Inheritance
* Subclasses provide specialized behaviors on the basis of common elements provided by the superclass. Through the use of inheritance programmers can reuse this code in the superclass many times.
* Abstraction
* Abstraction, is this process of categorising data. consider that a person is looking for a frame in an optician's shop. To be able to choose a frame from amongst the various types of frames available, he has to first identify the attributes he is looking far. Once he has identified the attributes, he has with him a category or class of frames. Similarly, to model any real life objects in OOPS an "object" has to be instantiated from a specific "class". This basic process of forming a class is known as "abstraction".
* Java and World Wide Web
* Java was basically designed for the web browsing. Java had some excellent features which other languages did not have. The internet helped catapult java to the forefront of programming, and java, in turn, has had a profound effect on the internet. The reason for this quite simple:java expands the universe of objects that can move about freely in cyberspace. Ina network, two very broad categories of objects are transmitted between the server and your personal computer: passive information and dynamic, active programs
* For example: when you read your e-mail, you are viewing passive data. Even when you download a program ‘s code is still only passive data
* Java/Java Applets: Java is probably the most famous of the programming languages of the Web. Java is an object-oriented programming language similar to C++. Developed by Sun Microsystems, the aim of Java is to create programs that will be platform independent. The Java motto is, "Write once, run anywhere." A perfect Java program should work equally well on a PC, Macintosh, Unix, and so on, without any additional programming. This goal has yet to be realized. Java can be used to write applications for both Web and non-Web use.
* Web-based Java applications are usually in the form of Java applets. These are small Java programs called from an HTML page that can be downloaded from a Web server and run on a Java-compatible Web browser. A few examples include live newsfeeds, moving images with sound, calculators, charts and spreadsheets, and interactive visual displays. Java applets can tend to load slowly, but programming improvements should lead to a shortened loading time.
* JavaScript/JScript: JavaScript is a programming language created by Netscape Communications. Small programs written in this language are embedded within an HTML page, or called externally from the page, to enhance the page's the functionality. Examples of JavaScript include moving tickers, drop-down menus, real-time calendars and clocks, and mouse-over interactions. JScript is a similar language developed by Microsoft and works with the company's Internet Explorer browser.
* **Back-End Tool**
* **MySQL Server:**

MySQL, pronounced either "My S-Q-L" or "My Sequel," is an open source relational database management system. It is based on the structure query language ([SQL](http://www.techterms.com/definition/sql)), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a Web scripting language like [PHP](http://www.techterms.com/definition/php) to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

**Features Of MySQL Server:**

 A broad subset of [ANSI SQL 99](http://en.wikipedia.org/wiki/SQL:1999), as well as extensions

 Cross-platform support

 [Stored procedures](http://en.wikipedia.org/wiki/Stored_procedure" \o "Stored procedure)

 [Triggers](http://en.wikipedia.org/wiki/Database_trigger" \o "Database trigger)

 [Cursors](http://en.wikipedia.org/wiki/Cursor_%28databases%29" \o "Cursor (databases))

 Updatable [Views](http://en.wikipedia.org/wiki/View_%28database%29)

 [Information schema](http://en.wikipedia.org/wiki/Information_schema" \o "Information schema)

 Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)

 [X/Open XA](http://en.wikipedia.org/wiki/X/Open_XA" \o "X/Open XA) [distributed transaction processing](http://en.wikipedia.org/wiki/Distributed_transaction_processing) (DTP) support; [two phase commit](http://en.wikipedia.org/wiki/Two-phase-commit_protocol) as part of this, using Oracle's [InnoDB](http://en.wikipedia.org/wiki/InnoDB) engine

 Independent [storage engines](http://en.wikipedia.org/wiki/Storage_engine) ([MyISAM](http://en.wikipedia.org/wiki/MyISAM" \o "MyISAM) for read speed, InnoDB for transactions and [referential integrity](http://en.wikipedia.org/wiki/Referential_integrity), [MySQL Archive](http://en.wikipedia.org/wiki/MySQL_Archive) for storing historical data in little space)

 Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB

 [SSL](http://en.wikipedia.org/wiki/Secure_Sockets_Layer" \o "Secure Sockets Layer) support

 Query [caching](http://en.wikipedia.org/wiki/Cache_%28computing%29)

 Sub-[SELECTs](http://en.wikipedia.org/wiki/Select_%28SQL%29) (i.e. nested SELECTs)

 Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.

 Full-text [indexing](http://en.wikipedia.org/wiki/Index_%28database%29) and searching using MyISAM engine

 Embedded database library

 [Unicode](http://en.wikipedia.org/wiki/Unicode" \o "Unicode) support (however prior to 5.5.3 [UTF-8](http://en.wikipedia.org/wiki/UTF-8) and [UCS-2](http://en.wikipedia.org/wiki/UTF-16/UCS-2) encoded strings are limited to the [BMP](http://en.wikipedia.org/wiki/Basic_Multilingual_Plane), in 5.5.3 and later use utf8mb4 for full unicode support)

 [ACID](http://en.wikipedia.org/wiki/Atomicity,_consistency,_isolation,_durability" \o "Atomicity, consistency, isolation, durability) compliance when using transaction capable storage engines (InnoDB and Cluster)

 Partitioned tables with pruning of partitions in optimizer

 [Shared-nothing](http://en.wikipedia.org/wiki/Shared-nothing" \o "Shared-nothing) clustering through [MySQL Cluster](http://en.wikipedia.org/wiki/MySQL_Cluster)

 Hot backup (via mysqlhotcopy) under certain conditions.

 Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at [run time](http://en.wikipedia.org/wiki/Run_time_%28program_lifecycle_phase%29)):

* Native storage engines (MyISAM, [Falcon](http://en.wikipedia.org/wiki/Falcon_%28storage_engine%29), Merge, Memory (heap), [Federated](http://en.wikipedia.org/wiki/MySQL_Federated), Archive, [CSV](http://en.wikipedia.org/wiki/Comma-separated_values), Blackhole, Cluster, EXAMPLE, [Maria](http://en.wikipedia.org/wiki/Maria_%28storage_engine%29), and InnoDB, which was made the default as of 5.5)
* Partner-developed storage engines ([solidDB](http://en.wikipedia.org/wiki/SolidDB" \o "SolidDB), NitroEDB, [ScaleDB](http://en.wikipedia.org/w/index.php?title=User:MPH007&action=edit&redlink=1), TokuDB, [Infobright](http://en.wikipedia.org/wiki/Infobright) (formerly Brighthouse), [Kickfire](http://en.wikipedia.org/wiki/Kickfire), [XtraDB](http://en.wikipedia.org/wiki/XtraDB), [IBM DB2](http://en.wikipedia.org/wiki/IBM_DB2)).[[35]](http://en.wikipedia.org/wiki/MySQL#cite_note-35) InnoDB used to be a partner-developed storage engine, but with recent acquisitions, [Oracle](http://en.wikipedia.org/wiki/Oracle_Corporation) now owns both MySQL core and InnoDB.
* Community-developed storage engines ([memcache engine](http://en.wikipedia.org/w/index.php?title=Memcache_engine&action=edit&redlink=1" \o "Memcache engine (page does not exist)), [httpd](http://en.wikipedia.org/wiki/Web_server), PBXT, [Revision Engine](http://en.wikipedia.org/w/index.php?title=Revision_Engine&action=edit&redlink=1))
* Custom storage engines

 Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second. (PostgreSQL has an advanced form of this functionality.

* **MySQL Workbench:**

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, and much more. MySQL Workbench is available on Windows, Linux and Mac OS.

## Design

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

## Develop

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage database connections. The Object Browser provides instant access to database schema and objects.

## Administer

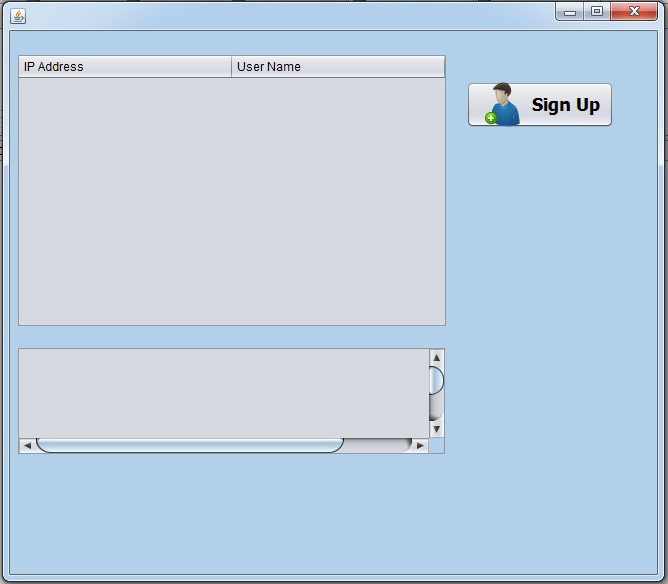
MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, and viewing database health.

## New! Database Migration

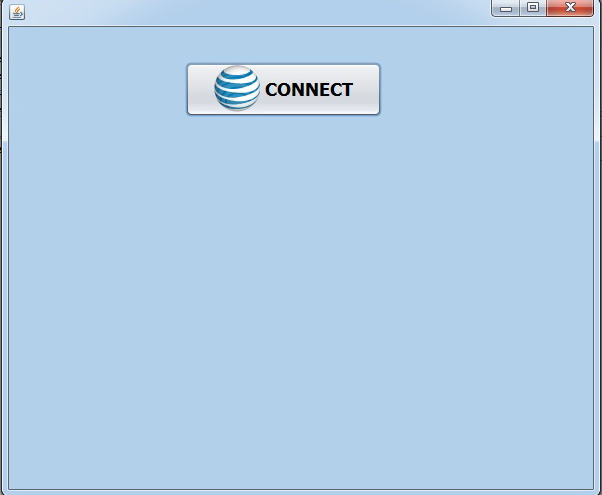
MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Sybase ASE, PostreSQL, and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL both on Windows and other platforms.

* **SNAPSHOTS**

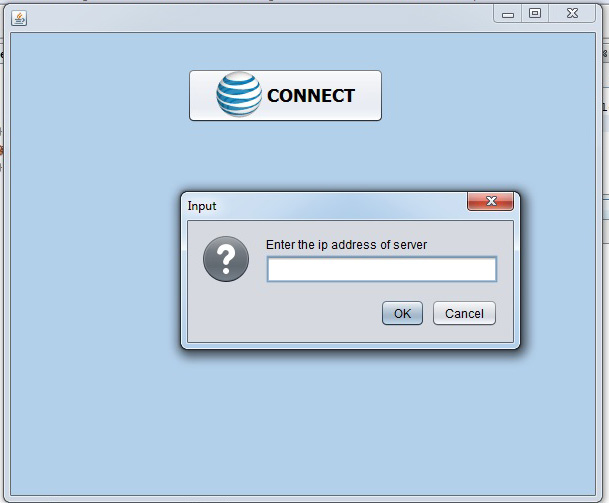
**SERVER**



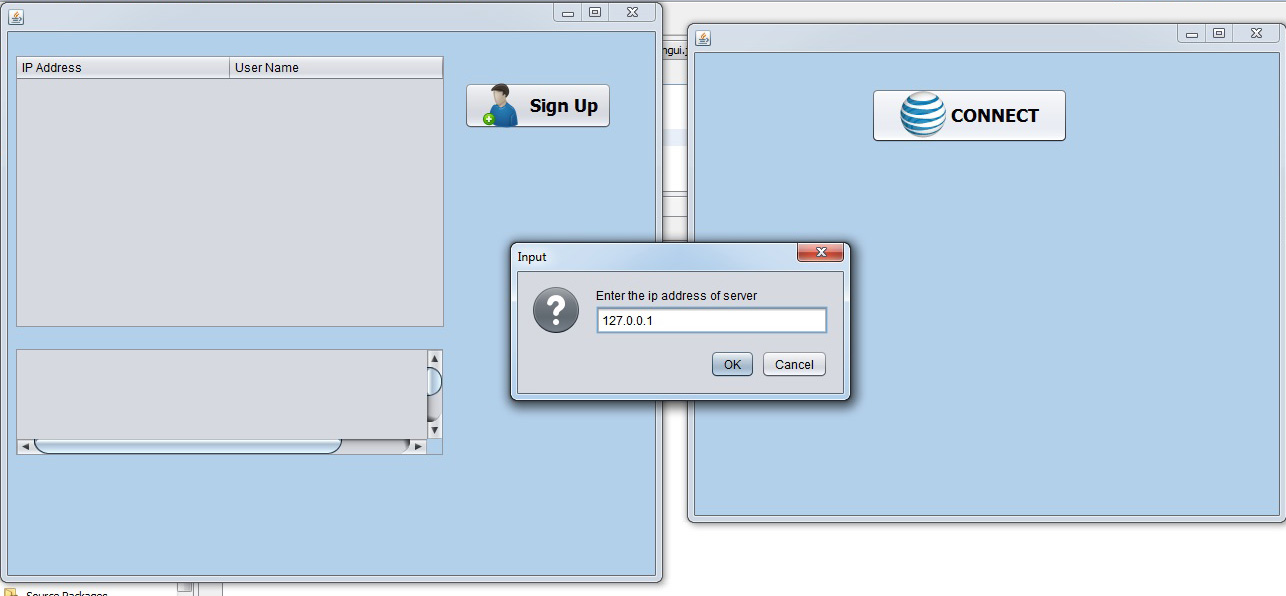
**CLIENT**

****

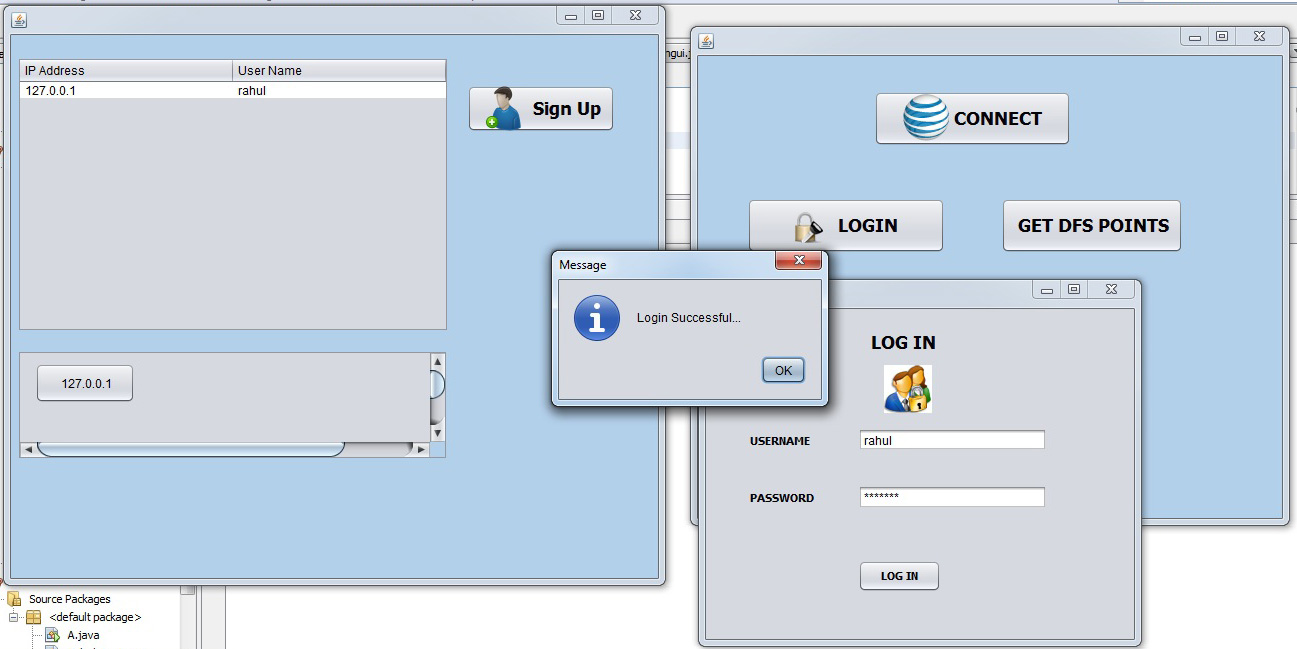
**CLIENT CREATE CONNECTION**

****

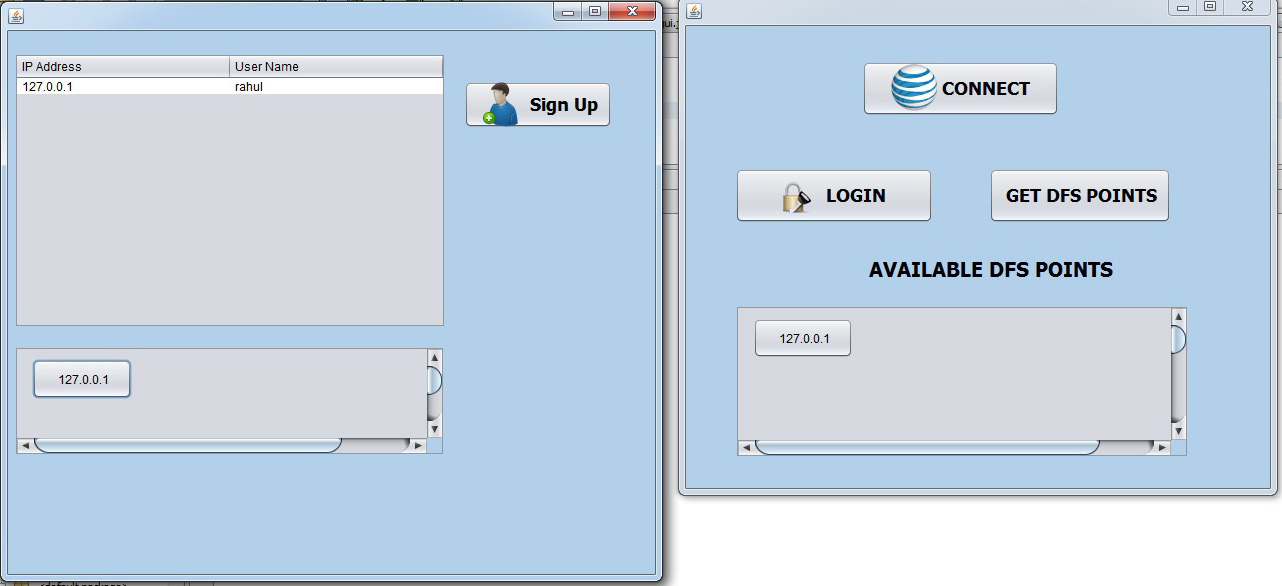
**CLIENT CONNECT WITH IP ADDRESS OF SERVER**

****

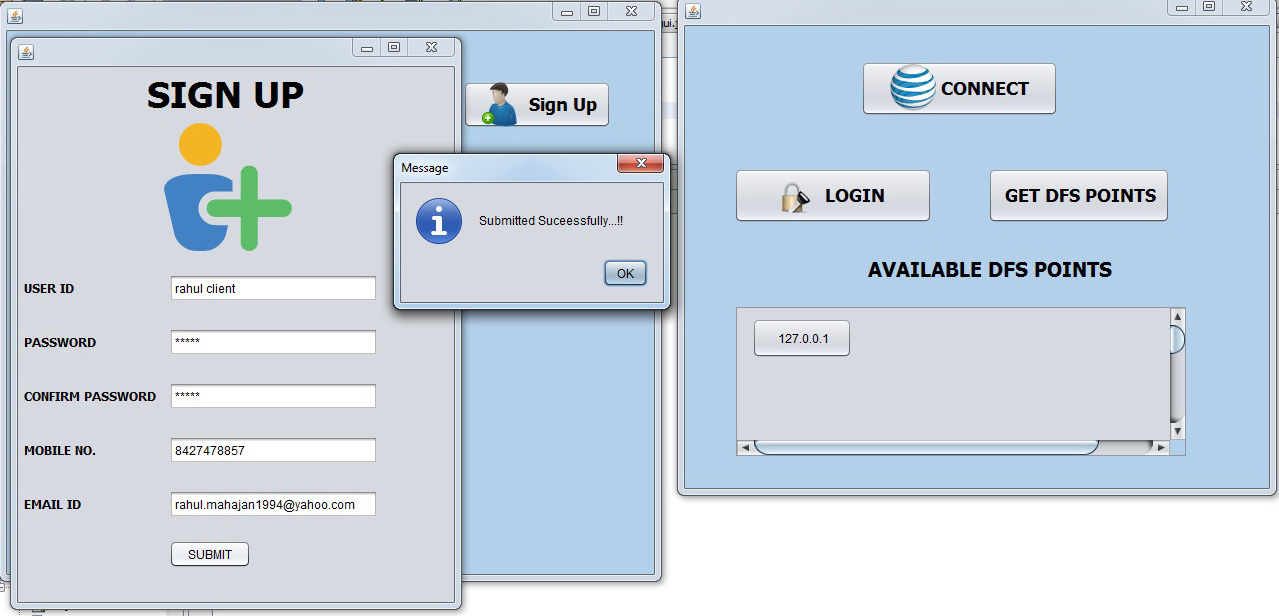
**LOGIN BY CLIENT THROUGH PASSWORD**

****

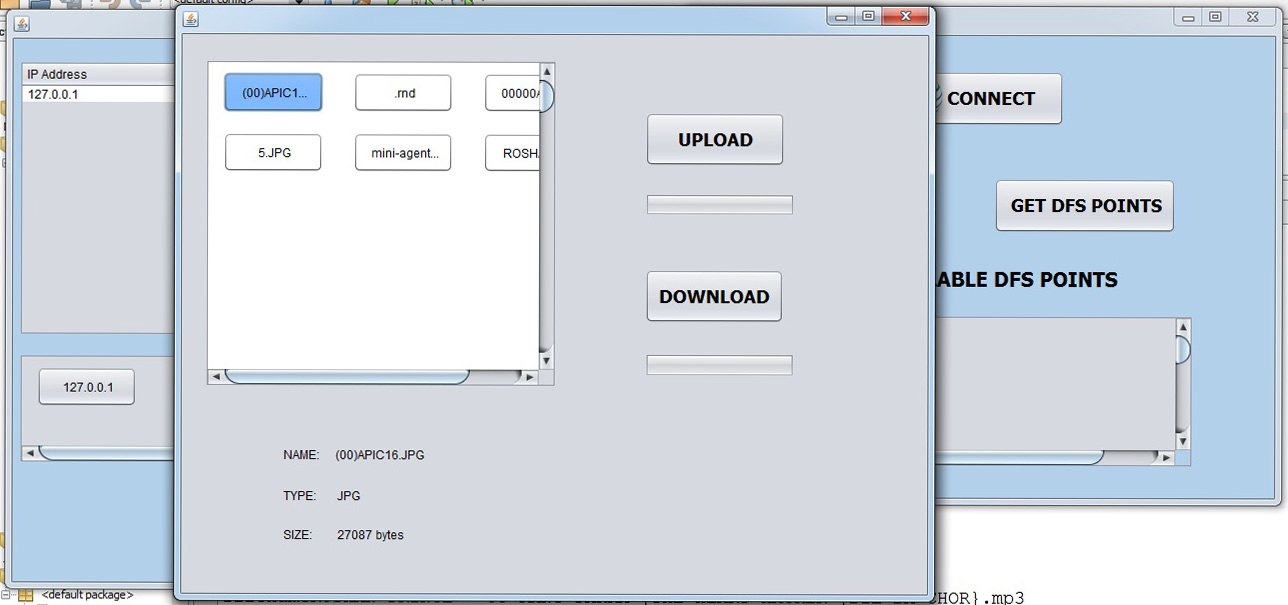
**CLIENT ARE SUCCESSFULLY CONNECTED**

****

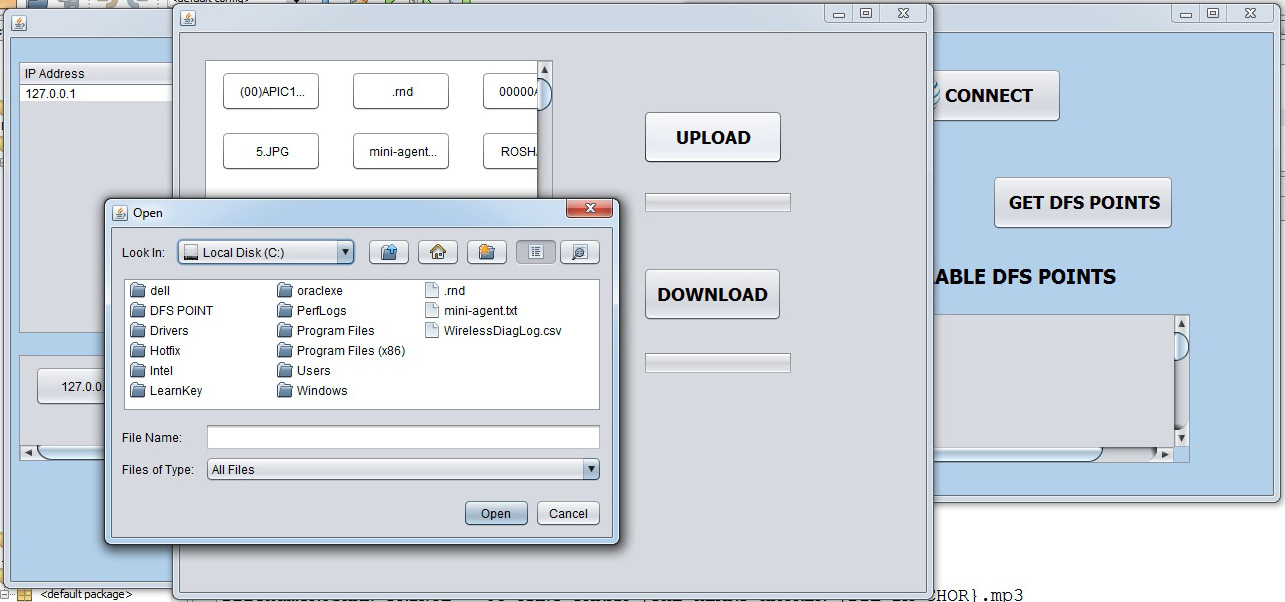
**ADD NEW CLIENT BY SERVER**

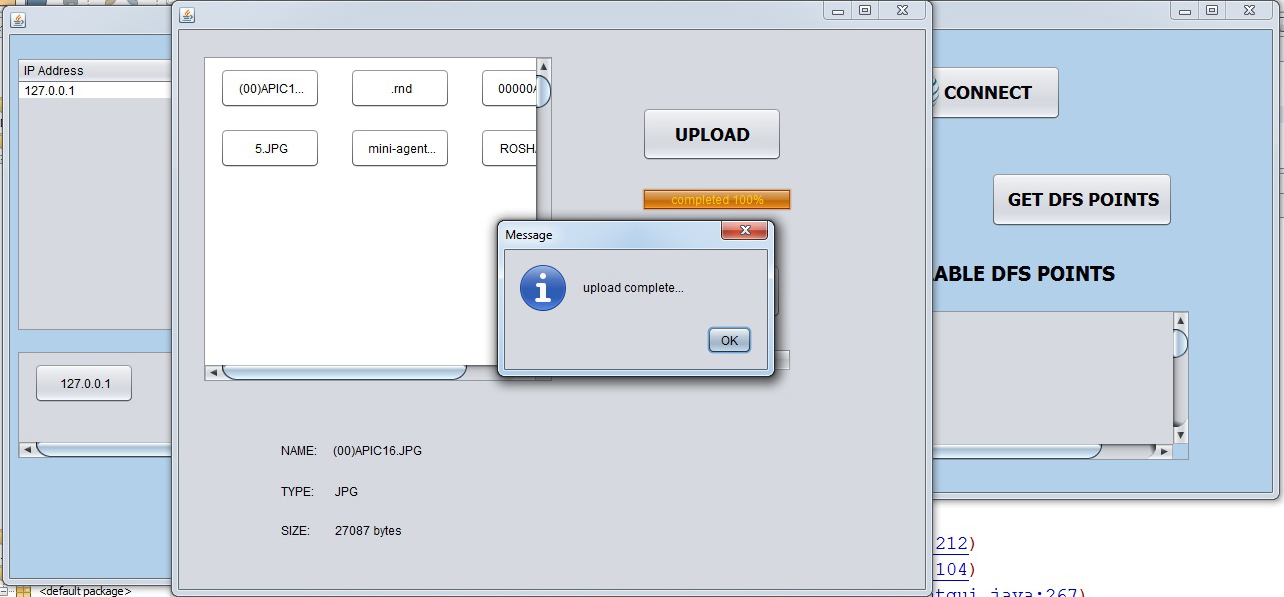
****

**CLIENT VIEW SHEAR FOLDER OF SERVER**

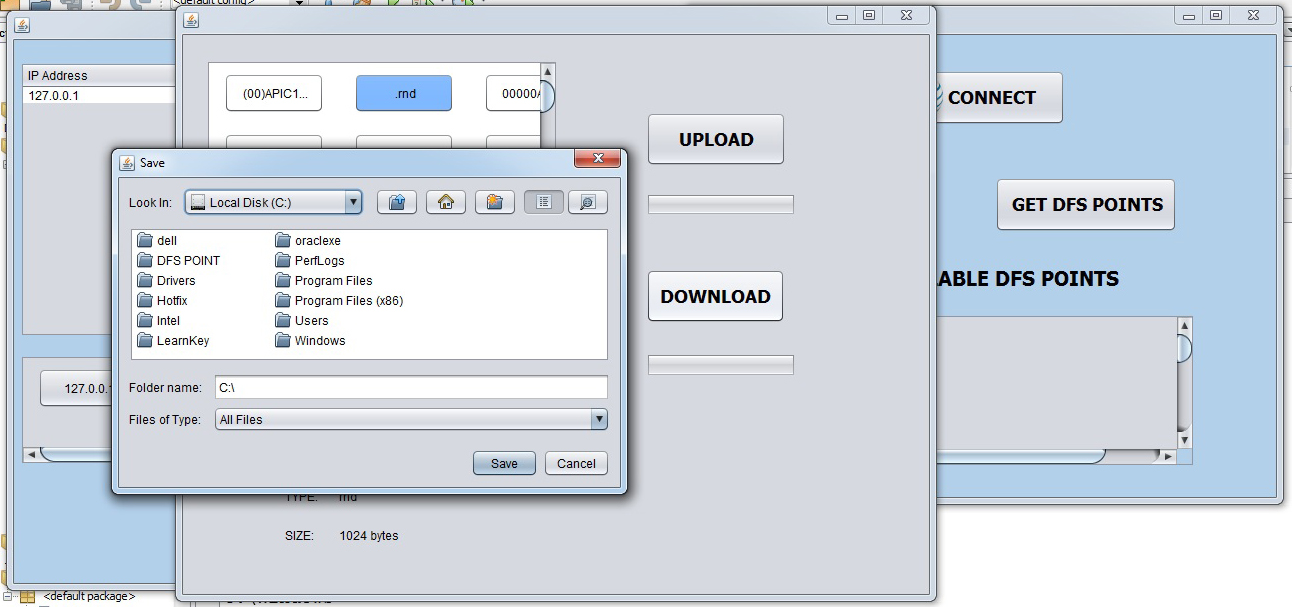
****

**UPLOADING DATA BY CLIENT**

****

****

**DOWNLOAD DATA BY CLIENT**

****

Chapter 5- Conclusions and Future Scope

**CONCLUSION**

Project can communicate various client to a server for data sharing. There are many clients and only one server because server side contain server socket and simple socket so, server accept connection pass to the simple socket then the sever socket is always free that are se to accept connection .

In client side only simple socket are used so server contain many client due to server socket but that clients are connection establish by server through login a new client .after that client contain his private key password. Login through that password are client are logically and physically connected to server communicate for data shearing between them the server shear the folder dfs to client .the client download and upload the data for client to server and server to client this process is called upload and download.

So, the project can successful to communication between client and server. Actually the server can give data but data can upload by client , server cannot know so, that are sure the data upload by client to server that are firstly permission by server.

**FURTHER SCOPE OF PROJECT IN FUTURE**

* Add authority in uploading of data through client.
* Shear full hard disk and particular folder by dynamically.

.

* **MAINTENANCE**

After a system has been executed and produced satisfactory results, it’s stored as a software package or in system library. The needs of an organization may change with time and a lot more may be expected from the system and this needs either development of a new system or modifications in the existing program. Modification in the system may also be required if it fails to working changed environments, which may be caused by the use of better machines.

The system maintenance means continuous modification and updating of the system to meet the requirements of the users. It’s certainly cheaper than developing a new system. Technical documentation plays an important role in the system maintenance.

Maintenance is the enigma of system development. It holds the software industry captive, typing up programming resources. Analysts and programmers spend far more time maintaining programs then they do writing them. Maintenance can be classified as corrective, adaptive or perfective.

**Maintenance System**

Software system maintenance covers a wide range of activities, including correcting coding and design errors, updating documentation and test data and upgrading user support. It is the process of changing the system to maintain its ability to survive. The system design can adopt any of the under given system maintenance strategies.

**Corrective Maintenance**

The design software can be maintained by corrective maintenance that is concerned with fixing reported errors in the software. Coding errors are relatively cheap to fix as compared to design errors and requirement errors. The requirement errors are more expensive, as redesigning of the entire system is to fix them.

**Adaptive Maintenance**

The design software can be maintained by adaptive maintenance if case there is a need to change the system environment such as a different hardware platform or for using it with different operating system.

**Predictive Maintenance**

The design system can be maintained by predictive maintenance if there is need to implement new functional requirements. They are generated due to the changing requirements of the software customers as their organization or business changes.