

A
Project Report On
Virtual Interactive Classroom

As partial fulfillment for the
Award of

BACHELOR OF ENGINEERING
(JUNE 2010)



Monitoring Faculty:

Mr. Darshak G. Thakore

Prepared By:

Ahesanali S. Suthar (06CP220)

Sanjay Y. Rajwadi(06CP233)

Pratik Macwan(07CP294)

Other Guiding Faculties:

Mr. Narendra M. Patel

Mr. Sunil A. Bakhru

Mr. Mosin Hasan

Department of Computer Science & Engineering

Birla Vishwakarma Mahavidyalaya

Vallabh Vidyanagar, GUJARAT.



BVM Engineering College
Vallabh Vidyanagar – Gujarat, India.

CERTIFICATE

This is to certify that **Mr. Sanjay Y. Rajwadi** - ID NO **06CP233**, **Mr. Ahesanali S. Suthar** - ID NO **06CP220** and **Mr. Pratik Macwan** - ID NO **07CP294** of Fourth Level B.E. (Computer Engineering) have successfully completed the final year project of **Virtual Interactive Classroom** towards partial fulfillment of the requirements of the Bachelor of Engineering in Computers at Birla Vishvakarma Mahavidyalaya, Engineering College, Vallabh Vidyanagar during the period January 2010 to June 2010.

Date: _____

Monitoring Faculty

Mr. Darshak G. Thakore

Head of Department

Mr. Prashant B. Swadas

ACKNOWLEDGEMENTS

Our Project – **Virtual Interactive Classroom** has been an acknowledgeable experience. We would like to thereby take this opportunity to thank all the persons who have helped us by contributing their skills, time and uphill struggle for the successful relation of this project.

First off, our special thanks to The **Almighty GOD** by whose power everything is made possible.

Our hearty thanks to our beloved Principal – **Dr. F. S. UMRIGAR SIR** for giving us the opportunity to develop this project.

Our hearty thanks to the Head of Computer Dept. - **Mr. PRASHANT SWADAS SIR**, without whom this project would not have left a launching pad. Thanks to his support and direction all the way through the completion of the project.

We are very fortunate to have **Mr. Darshak G. Thakore SIR** as our Project Guide who helped us when we needed clarification on a concept or a feature, as well as for giving us additional insight into the discipline. His untiring guidance and kind consideration has lead to this success. His valuable efforts and suggestion have contributed a lot to our project development.

We are thankful to **Mr N. M. PATEL SIR, Mr. S. A. BAKHRU SIR, & Mr. MAUSIN, and Mr. MUKESH SHAH SIR** from Bitnet Infosys for supporting us to build this project. Their knowledge and reviews installed in us a love of shaping up this project and helped us avoid some awkward code design decisions.

Besides, we are also thankful to our college staff, which supported us kindly.

Ahesanali S. Suthar(06CP220)

Sanjay Y. Rajwadi(06CP233)

Pratik Macwan(07CP294)

INDEX

Chapter 1: INTRODUCTION

| | | |
|---------|--------------------------|---|
| 1.1 | Abstract..... | 1 |
| 1.2 | Brief Introduction..... | 2 |
| 1.2.1 | Main Objective..... | 2 |
| 1.2.2 | Brief Description..... | 3 |
| 1.2.2.1 | Modules Description..... | 3 |
| 1.2.2.2 | Study Of Technology..... | 4 |

Chapter 2: DESIGN APPROACH AND IMPLEMENTATION

| | | |
|---------|---------------------------------|----|
| 2.1 | Specification..... | 9 |
| 2.1.1 | User Characteristic..... | 9 |
| 2.1.2 | Tools to be used..... | 9 |
| 2.1.3 | Assumption & Dependencies..... | 9 |
| 2.2 | Analysis..... | 10 |
| 2.2.1 | Functional Requirements..... | 10 |
| 2.2.2 | Nonfunctional Requirements..... | 11 |
| 2.2.2.1 | Hardware Requirements..... | 11 |
| 2.2.2.2 | Software Requirements..... | 11 |
| 2.2.3 | Feasibility Study..... | 11 |

| | | |
|---------|------------------------|----|
| 2.3 | Design..... | 12 |
| 2.3.1 | System Design..... | 12 |
| 2.3.1.1 | Data Flow Diagram..... | 12 |
| 2.3.1.2 | Use Case Diagram..... | 14 |
| 2.3.1.3 | Class Diagram..... | 17 |
| 2.3.1.4 | Sequence Diagram..... | 22 |
| 2.4 | Implementation..... | 25 |

Chapter 3: RESULTS

| | | |
|-----|-----------------------------|----|
| 3.1 | Snapshots..... | 26 |
| 3.2 | Testing & Verification..... | 33 |
| 3.3 | Conclusion..... | 36 |
| 3.4 | Future Enhancement..... | 37 |
| 3.5 | References..... | 38 |

CHAPTER: 1 INTRODUCTION

1.1 Abstract

This project VIRTUAL INTERACTIVE CLASSROOM is aimed to develop to handheld software for an administrator and also for a teaching faculty to look after, to manage and to control all the users accessing the network.

This project is a generalized project that can be used in any organization to handle a large network or in any computer lab to interact with different students as a teaching faculty.

This project is basically an extra facility provided to the administrator in order to manage a large network. This project will allows administrator to connect to any machine of the network either to monitor or to have an access to that machine.

In case, if administrator wants to monitor what the user is doing then he can connect to the user machine in a view mode and he/she can track what a particular user is doing at a current time and also he/she can inform that particular user by message.

In case, if an administrator wants to have a file from remote pc or to send a file to remote pc he/she can simply do it by connecting with user in FILE DOWNLOADING mode or FILE TRANSFER mode.

Also, students (users) and teaching faculty (admin) can interact with each other using chatting facility provided in the project and also make interaction with individuals.

1.2 Brief Introduction

1.2.1 Main Objective

- One of the main objective of this project to develop handheld software for an administrator to look after, manage and to control all the users access in the network.
- The other objective of this project is that, in any organization students (users) can connect with teaching faculty (admin) so that teaching faculty can interact with connected students with different facilities provided in the project to solve their problems related to study and also give presentation on related subjects.
- Also different connected students (users) can interact with each other so that they can share their doubts related to any subject and also can interact with the teaching faculty to clear their doubts.
- This project basically used where numbers of machines are connected to each other through network or in any Educational organization where different users like students and faculty are connected.
- This project can be used where the major administrator wants to keep watch on remote computer i.e. what is going on remote computer.
E.g. In the internet lab to keep watch on the users what they are doing.
- Also where some controls on remote machine are required like to shutdown, restart, logoff. This project will provide these facilities.

1.2.2 Brief Description

1.2.2.1 Modules Description:

1. File Transfer/Downloading:

This module provides file transferring and files downloading facility to administrator to transfer file from server machine to client machine or download the file from client machine to server machine by selecting required IP address of the client, the source path and the destination path.

2. Message Broadcast:

This module provides message sending facility to administrator to the connected any particular client or all the connected clients.

3. Chatting:

This module provides chatting facility to the all clients and server to interact with each other by logging into chat room by entering the server IP address and their names.

4. Remote View Mode:

This module provides screen capturing facility to the administrator to watch out any particular client machine's screen.

5. Session View:

This module provides server screen view to the connected clients by the server permission. Server can start the session and stop the session.

6. Access Control:

This module provides facility like shutdown, restart and logoff of the client machine. Only administrator can perform this task.

1.2.2.2 Study of Technology:

- We have decided to make a use of **JAVA RMI (Remot Method Invocation)** to build our application. In JAVA RMI the stub/skeleton interface has been provided which is going to make the client/server application possible by making a call to remote objects which have been running at remote machines. We will discuss about JVA RMI in detail as follow.

Introduction to JAVA RMI

Java RMI allows java object that executes on one machine to invoke a method of java object that executes on another machine. This feature allows building a distributed application.

In the Java distributed object model, a *remote object* is one whose methods can be invoked from another Java virtual machine, potentially on a different host. An object of this type is described by one or more *remote interfaces*, which are Java interfaces that declare the methods of the remote object.

Remote method invocation (RMI) is the action of invoking a method of a remote interface on a remote object. Most importantly, a method invocation on a remote object has the same syntax as a method invocation on a local object.

Basic sequence of execution in RMI occurs as follows:

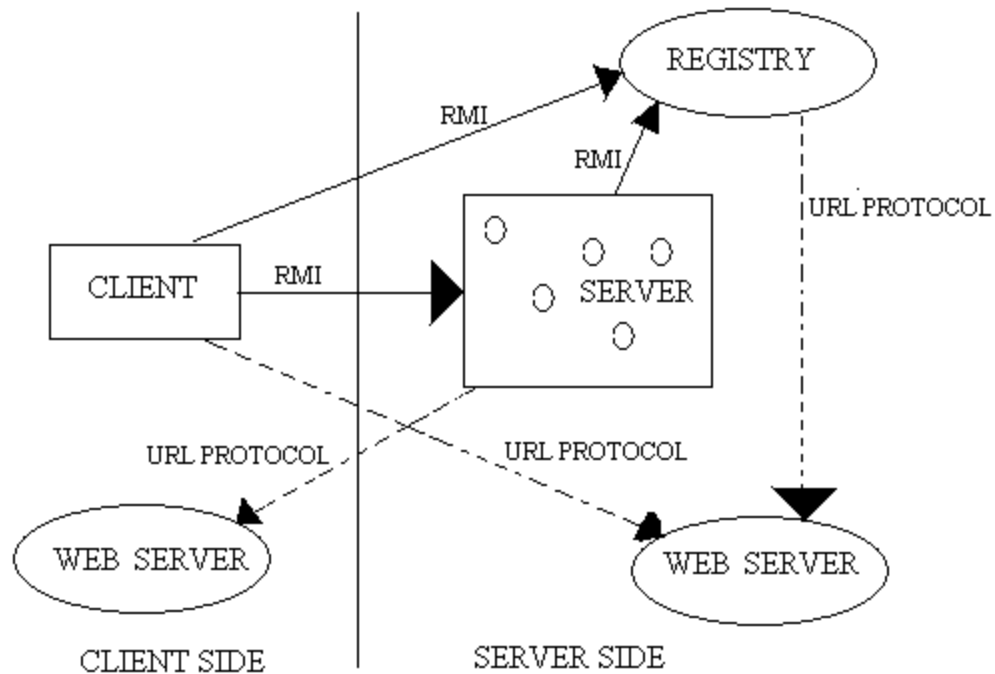
- The server receives a request from client, processes it and returns results back to the client.
RMI applications are often comprised of two separate programs: a server and a client.
- A typical **server application** creates a number of remote objects, makes references to those remote objects accessible, and waits for clients to invoke methods on those remote objects.

- A typical **client application** gets a remote reference to one or more remote objects in the server and then invokes methods on them. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. Such an application is sometimes referred to as a distributed object application.

Distributed object applications need to:

- Locate remote objects: - Applications can use one of two mechanisms to obtain references to remote objects. An application can register its remote objects with RMI's simple naming facility, the rmiregistry, or the application can pass and return remote object references as part of its normal operation.
- Communicate with remote objects: - Details of communication between remote objects are handled by RMI; to the programmer, remote communication looks like a standard Java method invocation.
- Load class bytecodes for objects that are passed as parameters or return values:-Because RMI allows a caller to pass pure Java objects to remote objects, RMI provides the necessary mechanisms for loading an object's code as well as transmitting its data.

The illustration below depicts an RMI distributed application that uses the registry to obtain references to a remote object. The server calls the registry to associate a name with a remote object. The client looks up the remote object by its name in the server's registry and then invokes a method on it. The illustration also shows that the RMI system uses an existing web server to load Java class bytecodes, from server to client and from client to server, for objects when needed. RMI can load class bytecodes using any URL protocol (E.g., HTTP, FTP, file, etc.) that is supported by the Java system.



RMI DISTRIBUTED APPLICATION

4-LAYERS OF RMI ARCHITECTURE:

Now we are going to discuss about the details of JAVA RMI architecture, which is a 4-layer Structure defined as below:

1. APPLICATION LAYER:

The layer is composed of basically building an actual client/server application where the definition for high level calls are made to access and export remote objects.

2. PROXY LAYER (STUB / SKELETON LAYER) :

This layer is what application deals with directly. All calls to remote method, marshalling of parameters and return of the objects are done through the proxies.

STUB:

In context of RMI the stub is the java object that resides on the client machine. Its function is to provide the same interface as the remote server. Remote calls initiated by client application are actually directed to stub. Stub work with other parts of the RMI system to formulate the request, which is being sent to the remote machine. Since in RMI objects may be passed as parameters they must be serialized before passing a parameter. When a stub's method is invoked, it does the following:

- Initiates a connection with the remote VM containing the remote object.
- Marshals (writes and transmits) the parameters to the remote VM
- Waits for the result of the method invocation
- Unmarshals (reads) the return value or exception returned
- Returns the value to the caller

SKELETON:

In context of RMI skeleton is the java object that resides on server machine. Working with other parts of RMI system it receives the request, perform the deserialization (if any required) and invoke an appropriate code on server. If response must be return to the client the skeleton will follow the same procedure as what stub had in requesting the service. When a skeleton receives an incoming method invocation it does the following:

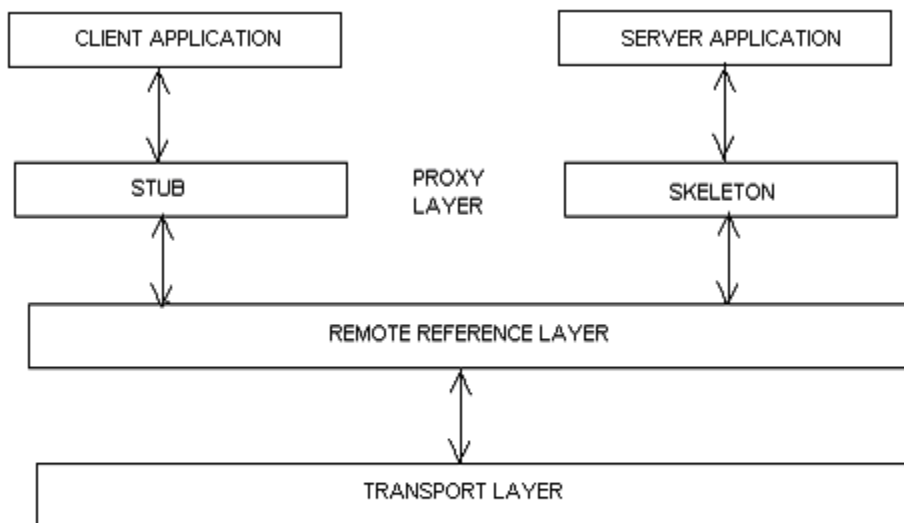
- Unmarshals (reads) the parameters for the remote method
- Invokes the method on the actual remote object implementation
- Marshals (writes and transmits) the result (return value or exception) to the caller.

3. REMOTE REFERENCE LAYER:

It is responsible for semantics of remote invocation. The layer is responsible for handling replicated objects and for performing implementation specific task with remote objects.

4. TRANSPORT LAYER:

The layer is responsible for actual setup of connection between the server and the client machines, transporting data between them and terminating the connection as indicated by application.



4-LAYER RMI ARCHITECTURE

CHAPTER: 2 DESIGN APPROACH & IMPLEMENTATION

2.1 Specification

2.1.1 User Characteristics

User characteristics refer to the users of the system. There are two main users of this system.

Administrator:

Administrator is the main user of the system. Server is started by the Administrator and has the information about all the connected clients. Admin can use all the facilities provided in the system. Admin can interact with the connected clients and also monitoring them.

Clients:

Clients are the secondary user of the system. Client can connect to the server and also use some facilities provided to them. They can interact with each other and server too.

2.1.2 Tools to be used

- Netbeans IDE 6.5M1
- JAVA Development kit JDK 1.6
- OS Environment: Windows XP

2.1.3 Assumption & Dependencies

Assumption:

- It is assume that user of this system have basic knowledge computer & networking.

- It is assume that user's machine have network facility and connected with server.
- It is assume that on server side Administrator application is running and client side Remote Agent application is running.

Dependencies:

- This system depends on the Network connection.

2.2 Analysis

2.2.1 Functional Requirements

- Administrator starts the application and authenticated by the application and ready the server for client requests.
- Client is connected to the server using IP address of the server.
- Sever can transfer or download the file to/from the clients by selecting the IP address of the particular and selecting the source and destination path.
- Users can interact with each other by chat interface, connecting with server by selecting IP address of the server and their user name. They can send a message by entering the message in text field and click on send button.
- Administrator can broadcast a message to all the connected clients.
- Administrator can remote view of the client machine by using remote view facility by selecting the IP address of the client and click on the capture button. Administrator can also make the facility to the client so, the client can see the administrator screen.
- Administrator can turn off, restart and logoff the client machine by the selecting IP address of the client and by clicking on the perform button.

2.2.2 Non-Functional Requirements

2.2.2.1 Hardware Requirements

- 128 MB RAM
- 40 MB swap space
- 50 MHz or higher processor
- NIC card and networking tools.

2.2.2.2 Software Requirements

- GUI based operating system that supports networking.
- JAVA2 runtime environment 1.4.2 or higher for the installed OS

2.2.3 Feasibility Study

2.2.3.1 Technical Feasibility:

We have the necessary tools like Netbeans IDE and Windows XP O.S as a platform to develop and we know JAVA technology so this project is technically feasible.

2.2.3.2 Operational Feasibility:

The application can be used by any technical, non-technical person. No special technical skill is required for operation and maintenance. And hence the project is operationally feasible.

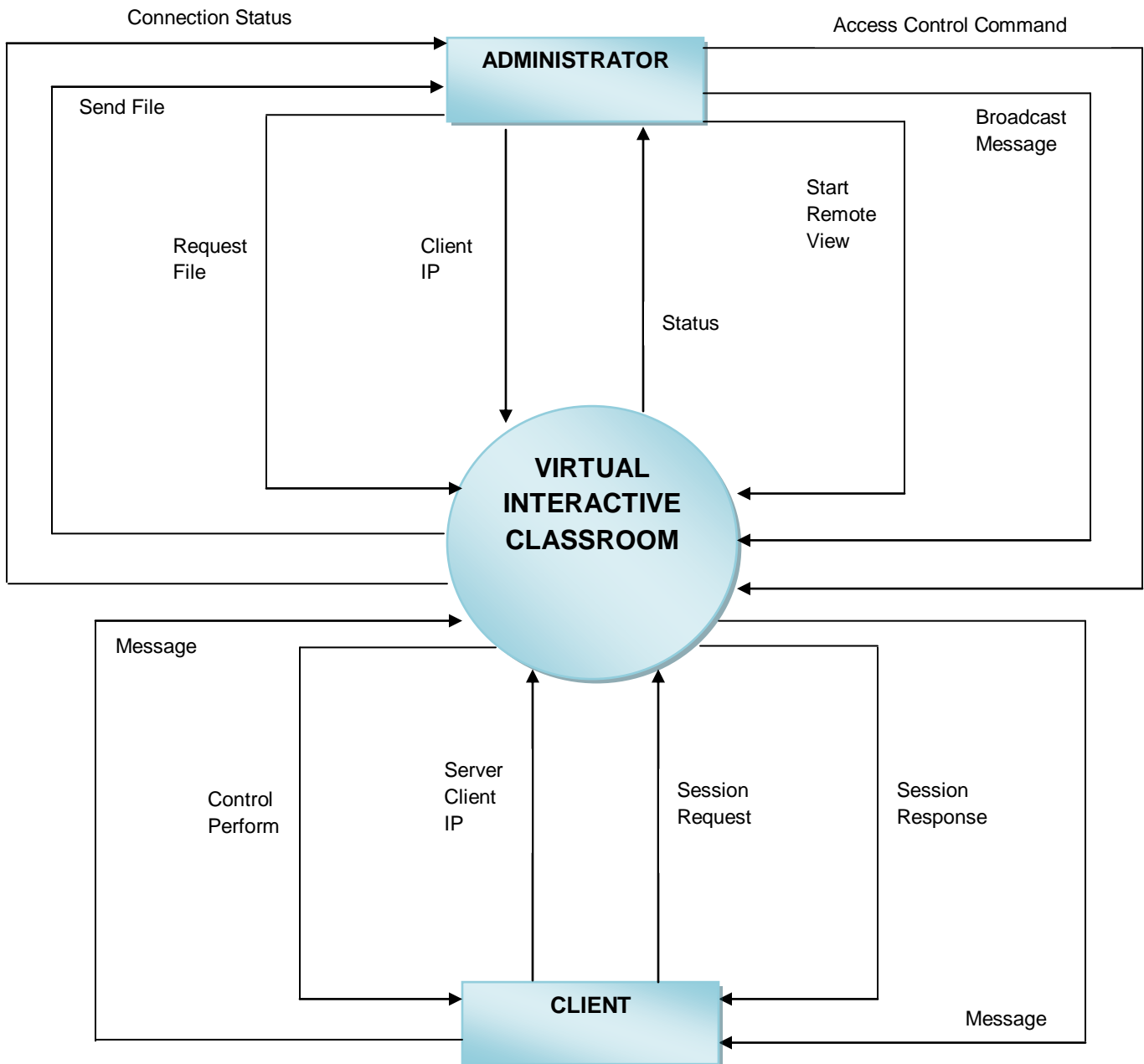
2.2.3.3 Economical Feasibility:

All the tools required for developing the project is readily available in the college and hence there is no requirement to buying any, thereby making it economically feasible also.

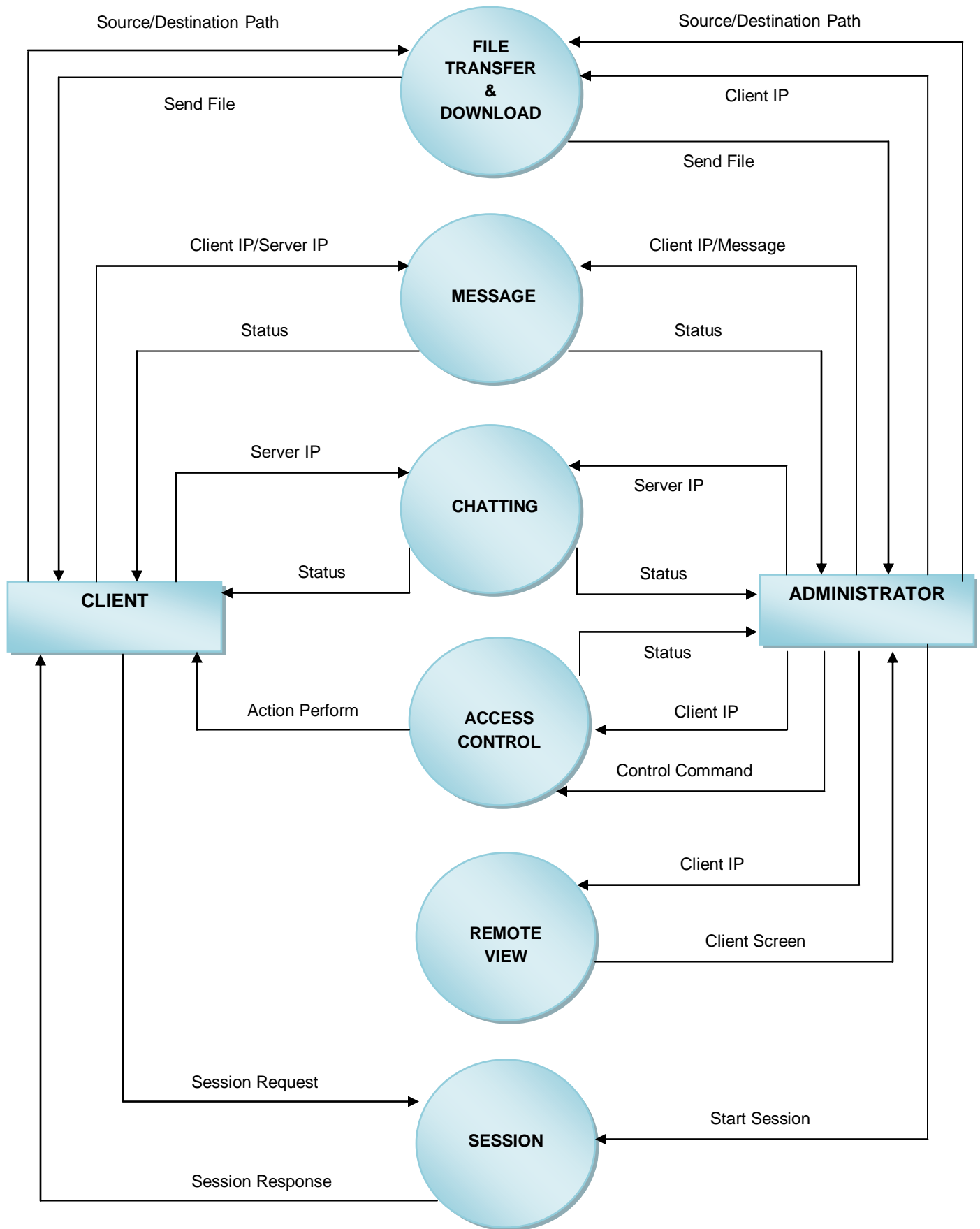
2.3 Design

2.3.1 System Design:

2.3.1.1 Data Flow Diagram:

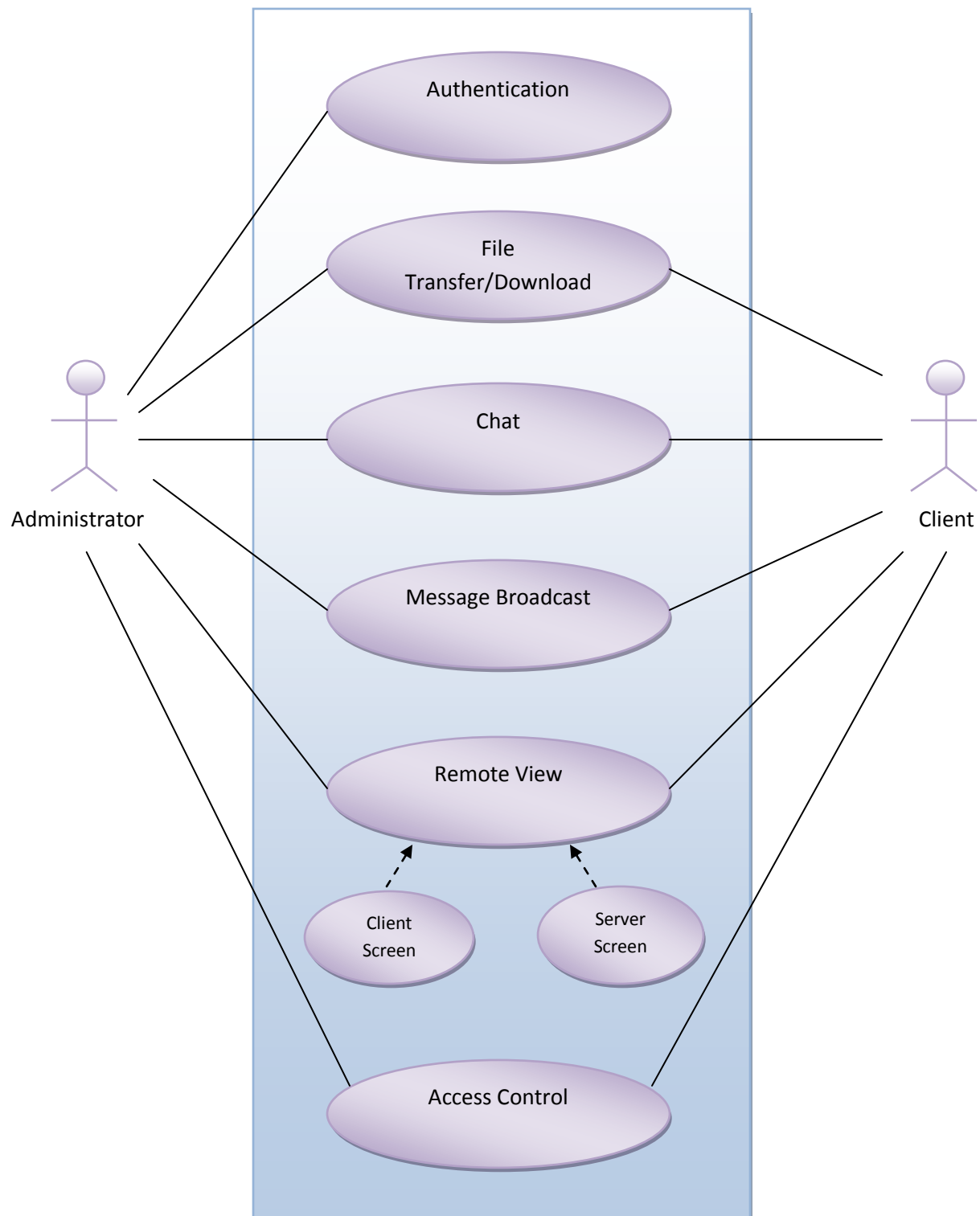


Context Level DFD



LEVEL-1 DFD

2.3.1.2 Use Case Diagram:



Use-Case Diagram

❖ Use-Case Report:

Authentication

| | |
|-----------------------|---|
| Use-case Name | Authentication |
| Actor | Administrator |
| Trigger | Login perform |
| Basic Path | Administrator enters username and password and click on login |
| Post Condition | Administrator will be authenticated successfully and Remote Controller window will appear |
| Exception Path | If authentication failed message will be displayed. |

File Transfer

| | |
|-----------------------|---|
| Use-case Name | File Transfer |
| Actor | Administrator, Client |
| Trigger | Menu Selection |
| Precondition | Remote controller or Remote agent is in running mode. |
| Basic Path | User enters the destination IP address and Select the file to be send and destination path. |
| Post Condition | File will be transferred to the specified location. |
| Exception Path | If the destination IP address is not found or Destination path not found or Source file not found message will be appeared. |

Chat

| | |
|-----------------------|---|
| Use-case Name | Chat |
| Actor | Administrator, Client |
| Trigger | Menu Selection |
| Precondition | Remote controller or Remote agent is in running mode. |
| Basic Path | Users enters the destination IP address and write the message. And click on the send button. |
| Post Condition | Message will be sent to the destination. |
| Exception Path | If the destination path is not found then error message will be appeared. |

Message Broadcast

| | |
|-----------------------|--|
| Use-case Name | Message Broadcast |
| Actor | Administrator |
| Trigger | Menu Selection |
| Precondition | Remote controller or Remote agent is in running mode. |
| Basic Path | Administrator enters the message and broadcast it. |
| Post Condition | Message will be prompted onto clients screen. |
| Exception Path | If any error is there error message will be displayed. |

Remote View

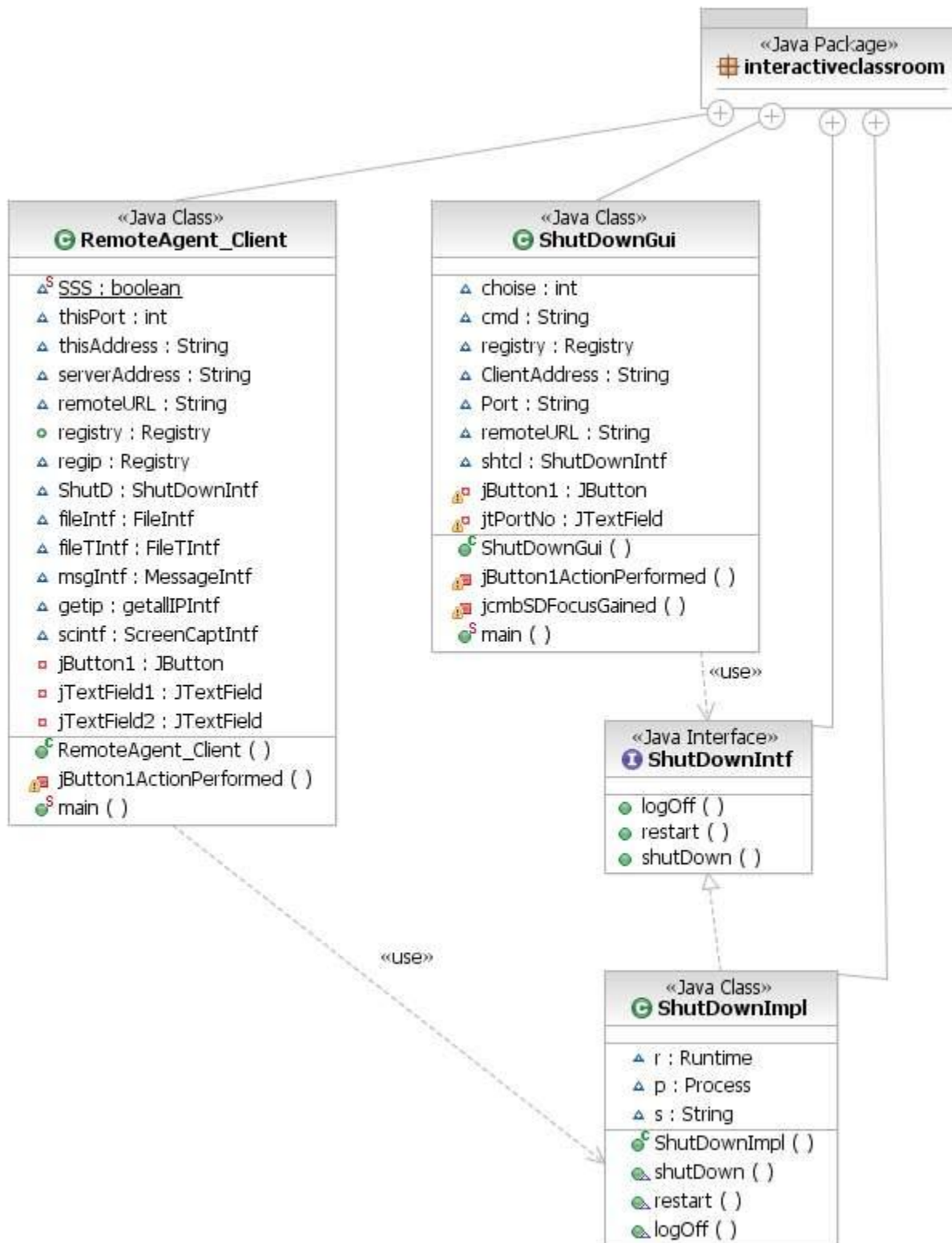
Client Screen Capture

| | |
|-----------------------|--|
| Use-case Name | Client Screen Capture |
| Actor | Administrator |
| Trigger | Menu Selection |
| Precondition | Remote controller or Remote agent is in running mode. |
| Basic Path | Administrator select the IP address of the client and click on the capture button. |
| Post Condition | Client screen will be captured and displayed on Administrator Screen. |
| Exception Path | If client is not found then the error message will be |

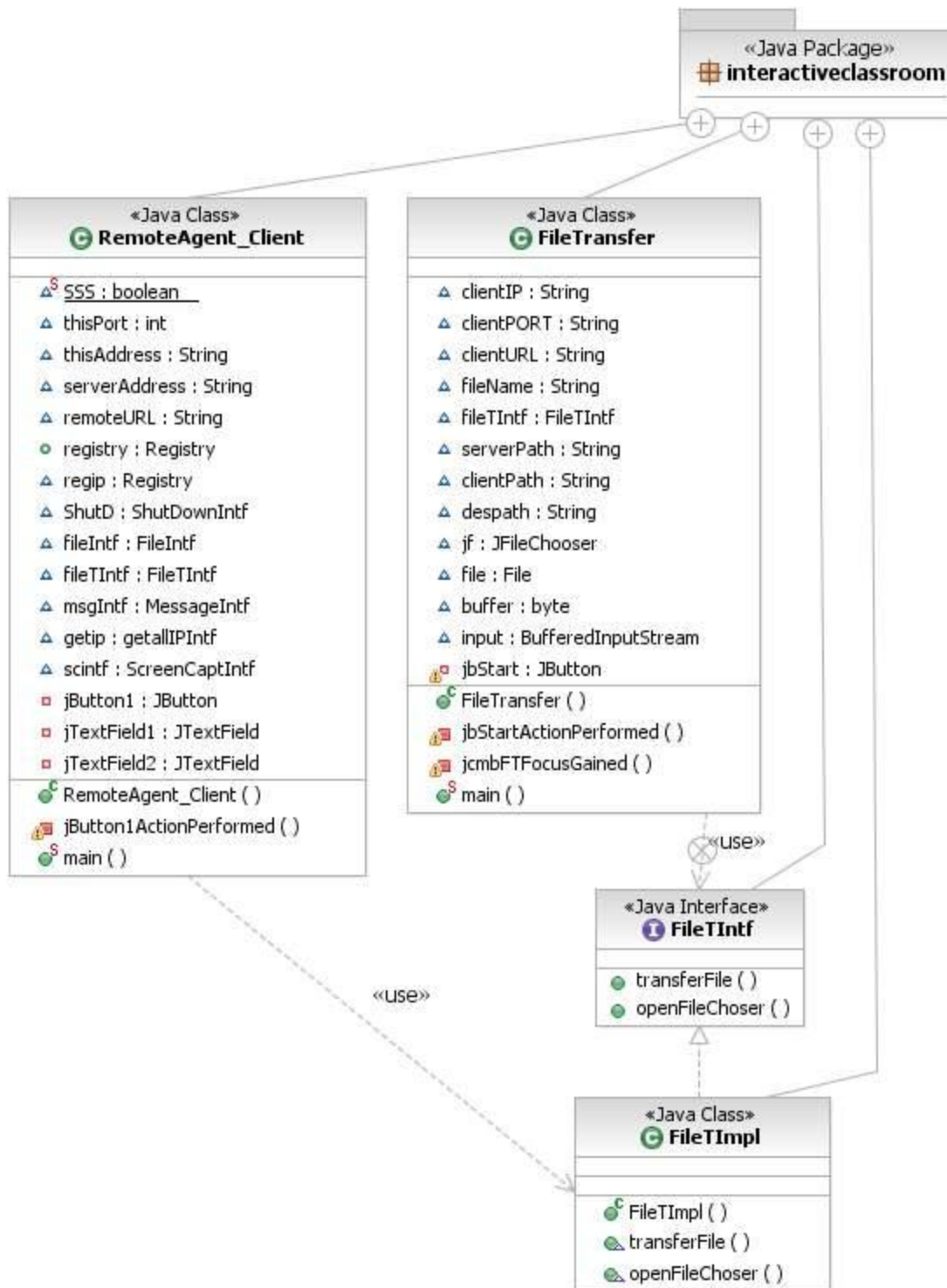
Server Screen Capture Session

| | |
|-----------------------|--|
| Use-case Name | Server Screen Capture Session |
| Actor | Administrator, Client |
| Trigger | Menu Selection |
| Precondition | Remote controller or Remote agent is in running mode. |
| Basic Path | Administrator start the session by clicking on the start session. And clients will be ready by clicking on connect. And administrator stops the session by clicking on the stop session. |
| Post Condition | Administrator screen will be displayed to the all the clients with some time delay. |
| Exception Path | If any client is not found then error message will be appeared. |

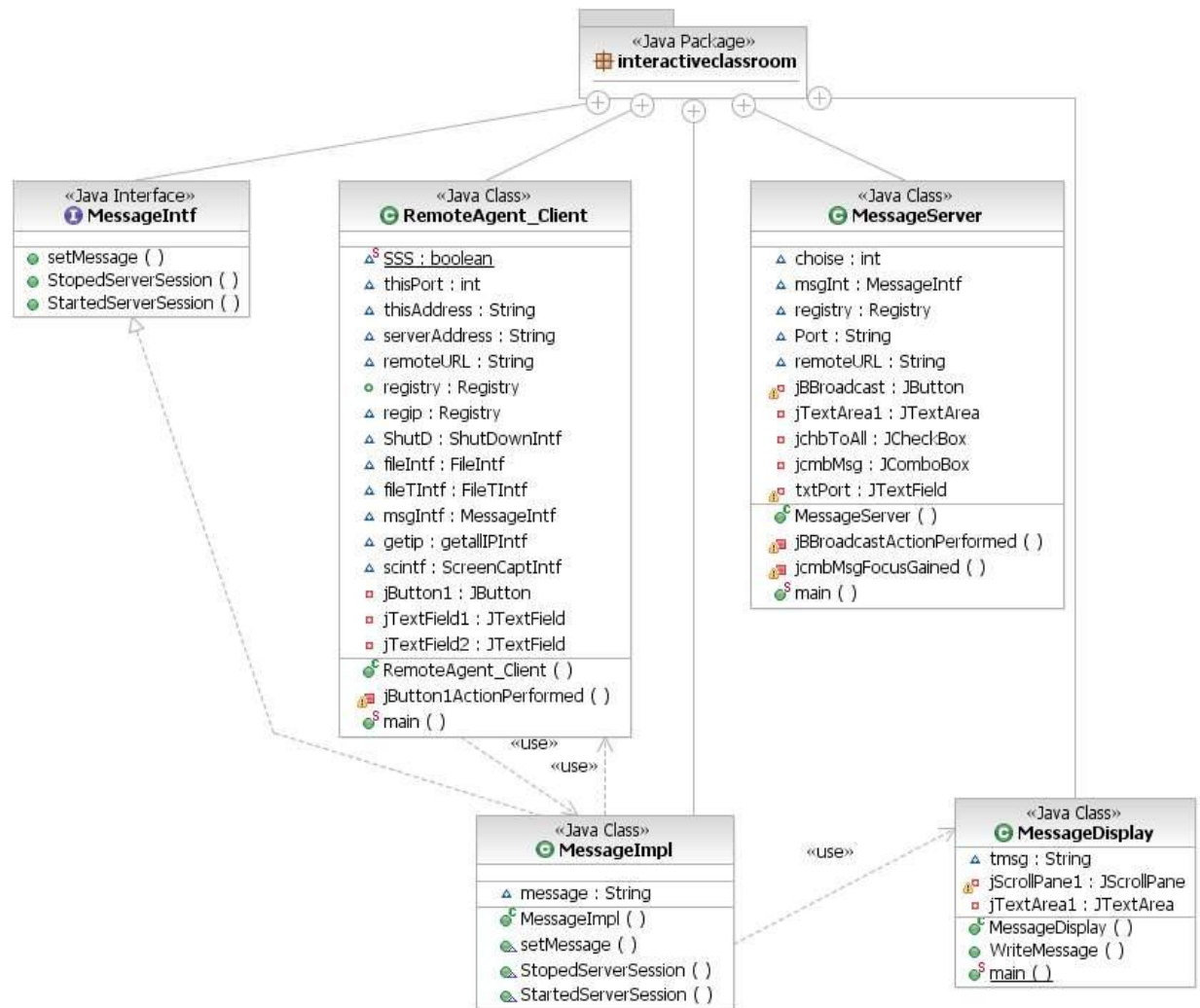
2.3.1.3 Class Diagram:



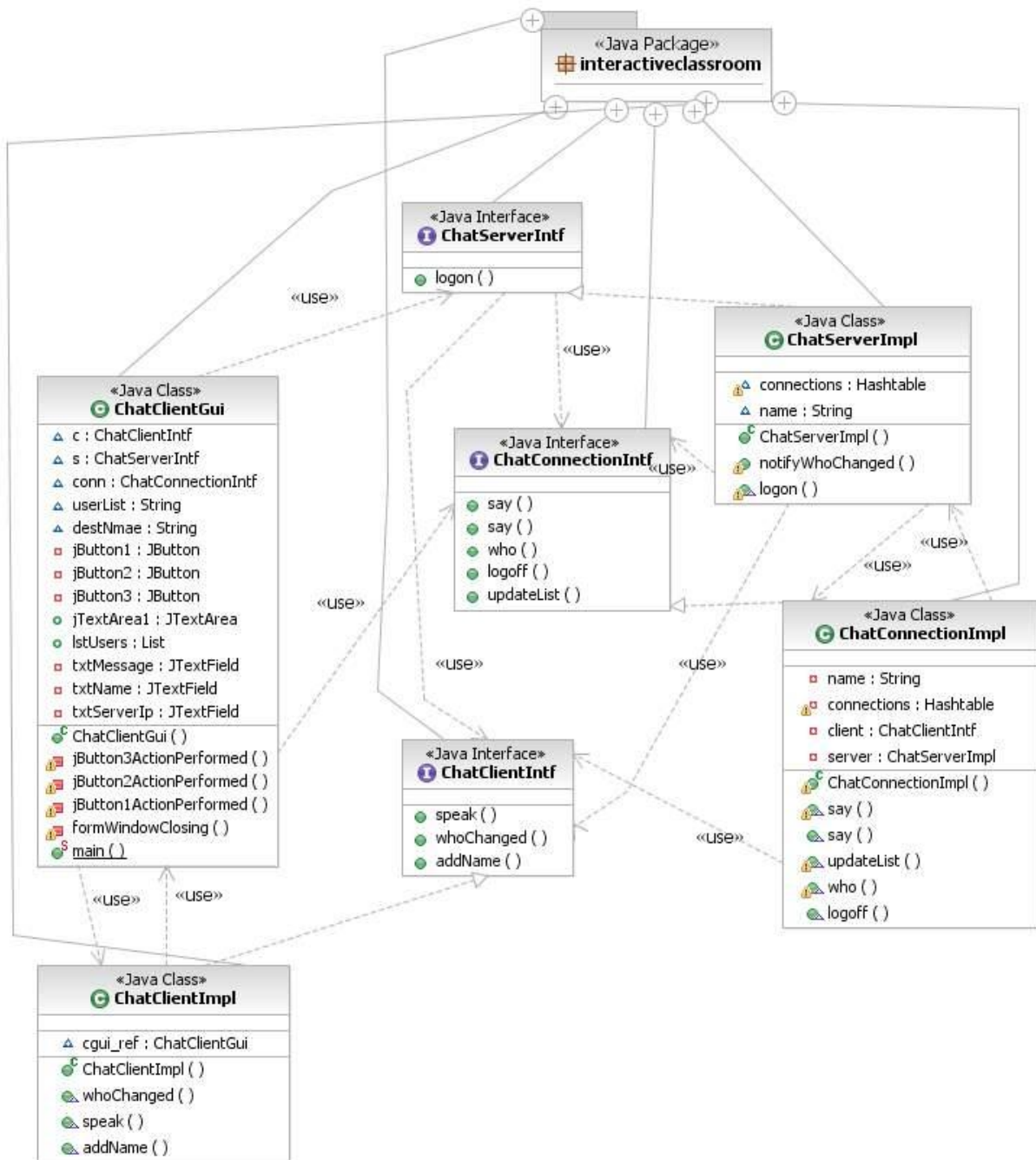
ACCESS CONTROL MODULE Class Diagram



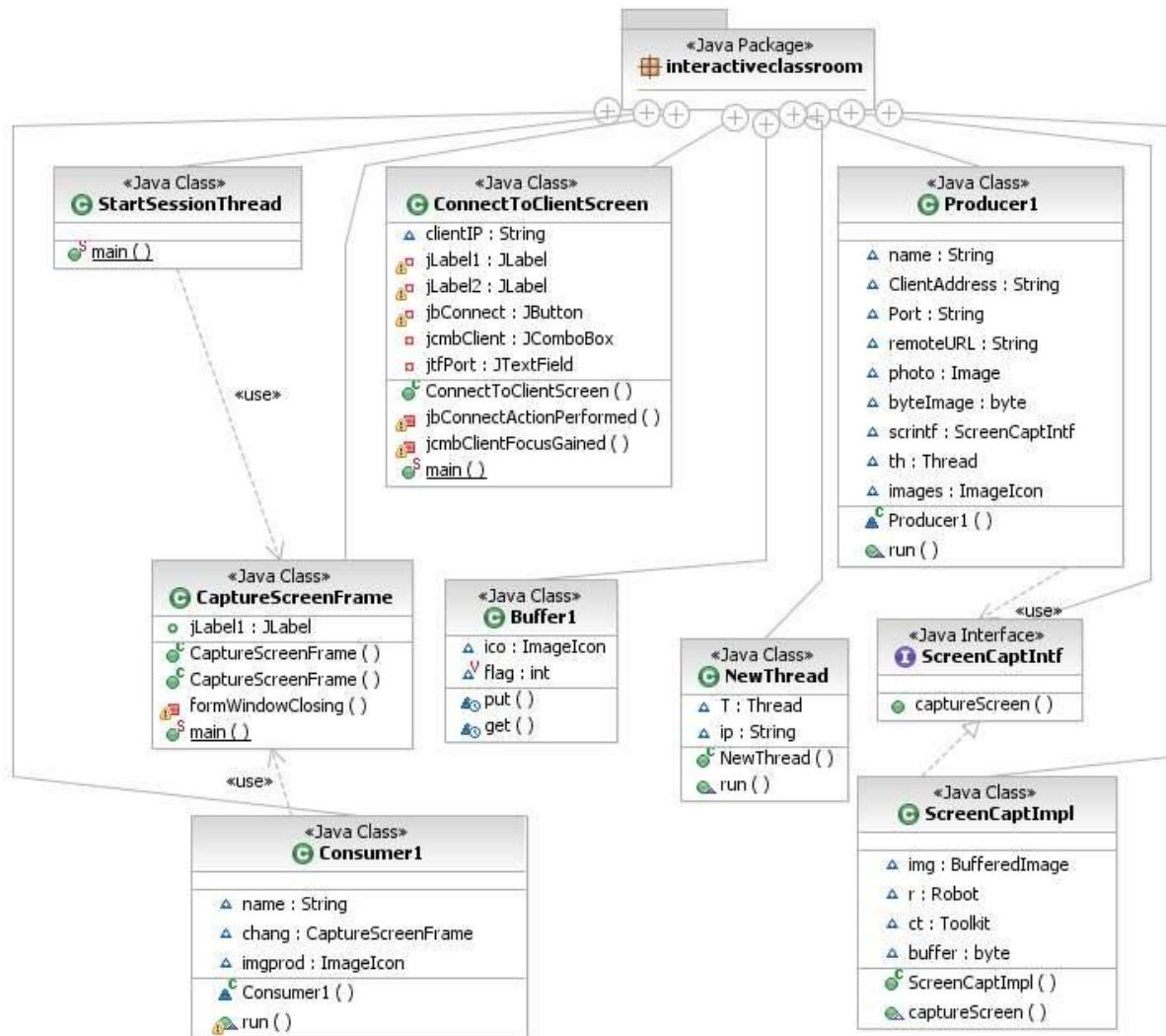
FILE TRANSFER MODULE Class Diagram



MESSAGE BROADCAST MODULE Class Diagram



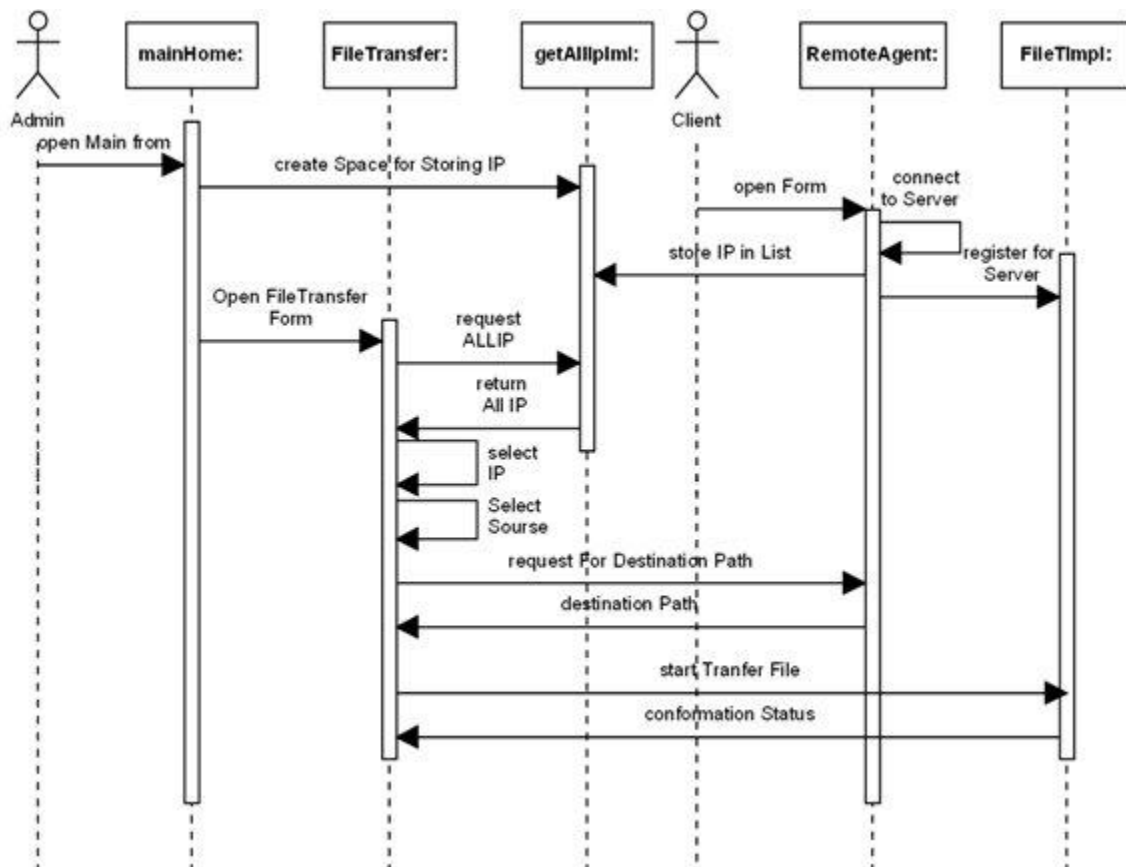
CHATTING MODULE Class Diagram



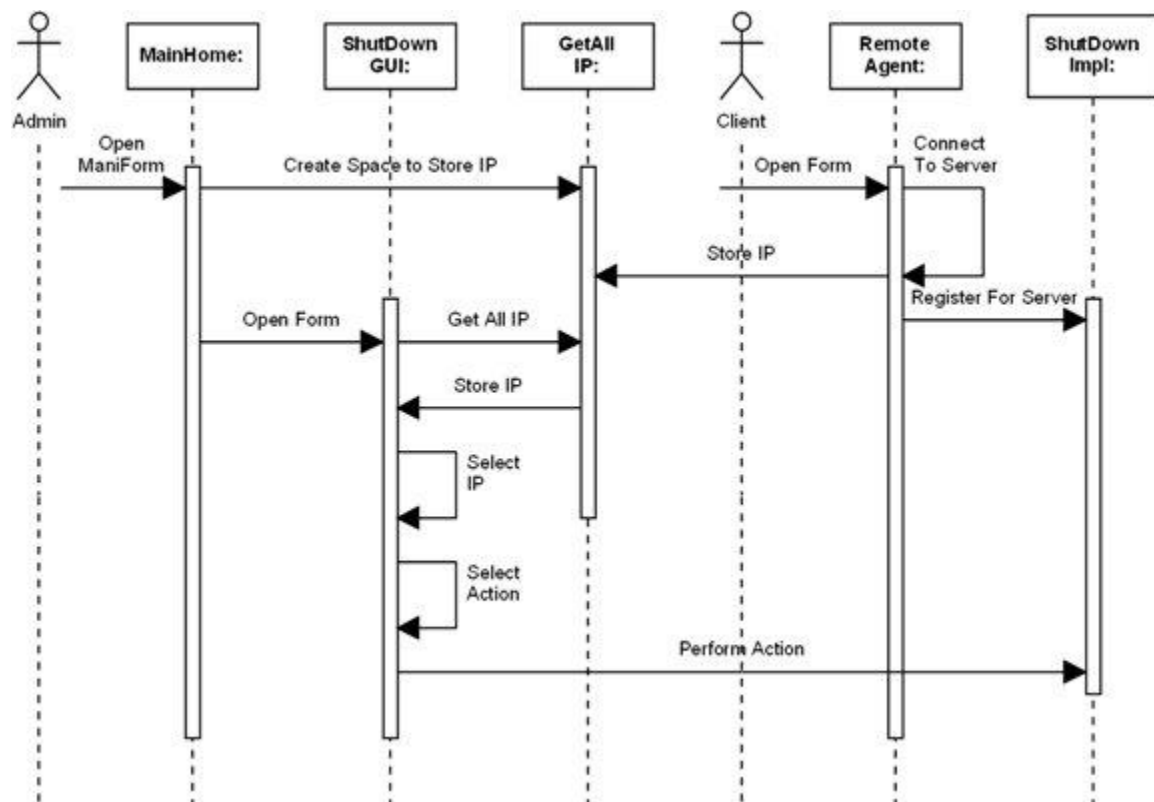
REMOTE VIEW MODULE Class Diagram

2.3.1.4 Sequence Diagram:

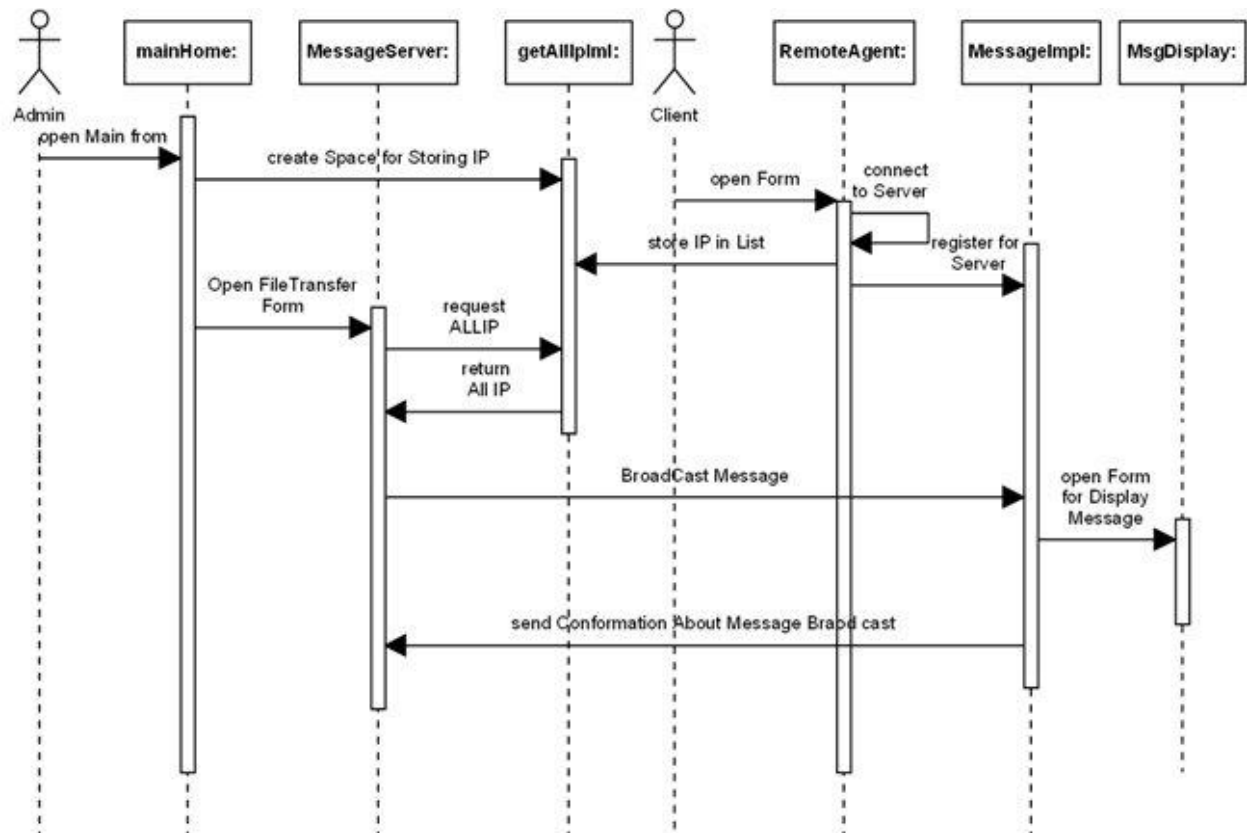
❖ For Use-Case File Transfer



❖ For Use-Case Access Control



❖ For Use-Case Message Broadcast



2.4 Implementation

Basically this project is divided in two parts.

1. Administration (Server)
2. Remote Agent (Client)

We have five modules and we have implemented interfaces for each module. These interfaces are implemented by their relative classes. And relative module uses their relative class.

So the Remote Agent (Client) registers the object for each module. So Administrator (Sever) can use that for remote method invocation on that machine by looking up for the object by registered name and invoke the methods that are implemented by relative class.

Same way when Remote Agent wants services from Administrator, object for that is registered by Administrator (server). And Client is looking for that object by registered name and invokes the methods.

CHAPTER: 3 RESULTS

Here we have given snapshots of different module related to different user I.e. Administrator & Clients. How the different user can interact with different facilities provided to them via form of the particular operation. So they can interact with the other clients & server.

We have also covered the different types of testing and testing methods like white box testing, black box testing and also System testing & Integration testing. In System testing covers alpha testing, beta testing and acceptance testing. Integration testing is done when all the module is tested at unit test.

At last we have also mention the conclusion related to different functionalities of the project and the future enhancement of the project how the project can be implemented in more sophisticated way. And also the references are related to project is described.

3.1 Snapshots

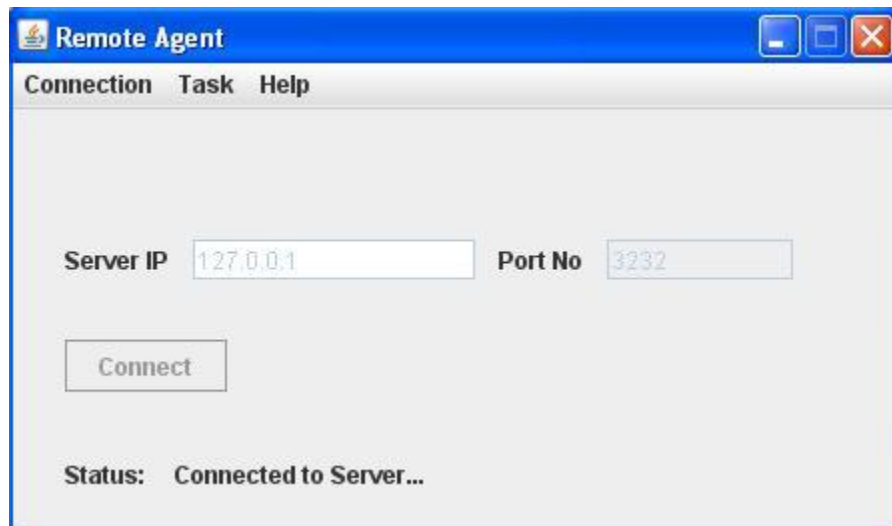
❖ Administrator:

This is the Administrator's Frame who has the central Control.



❖ Remote Agent Client:

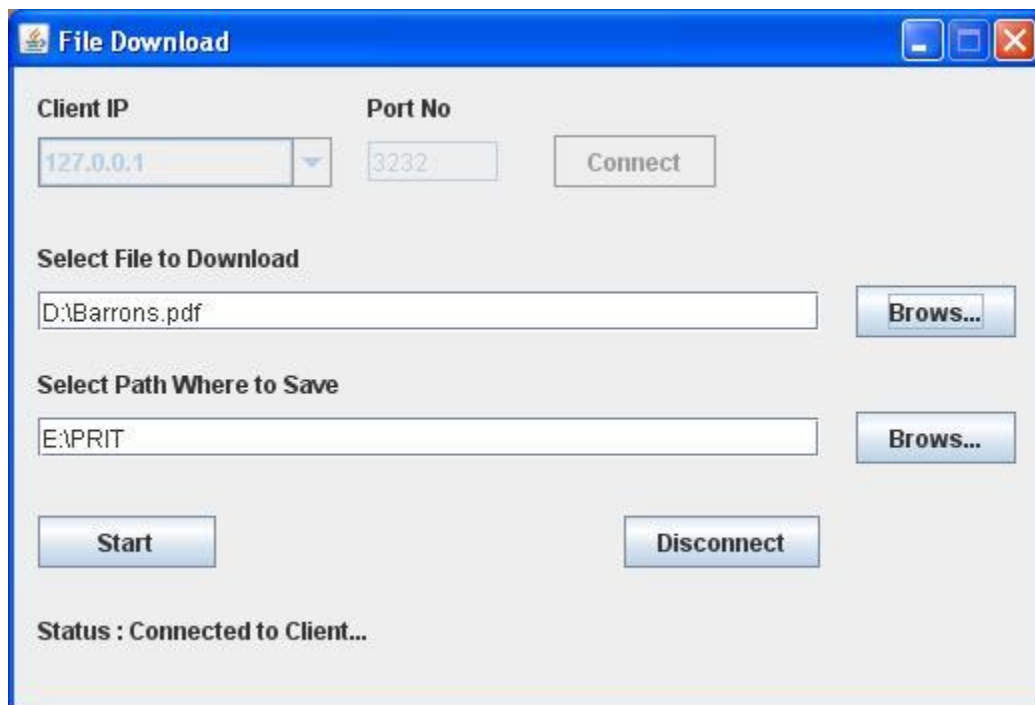
This is the Remote Agent Client's Frame.



The screenshot shows a window titled "Remote Agent" with a menu bar containing "Connection", "Task", and "Help". The main area contains a "Server IP" text box with the value "127.0.0.1" and a "Port No" text box with the value "3232". Below these is a "Connect" button. At the bottom, the status is displayed as "Status: Connected to Server..."

❖ File Download/Transfer:

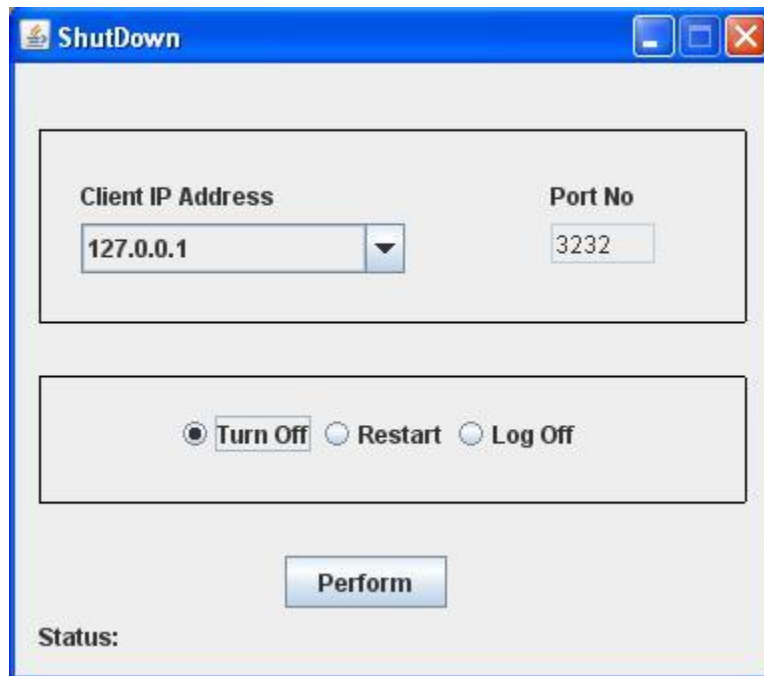
This is the File Download/Transfer Form for Administrator.



The screenshot shows a window titled "File Download" with a menu bar containing "Connection", "Task", and "Help". The main area contains a "Client IP" text box with the value "127.0.0.1" and a "Port No" text box with the value "3232". Below these is a "Connect" button. The "Select File to Download" section has a text box with the value "D:\Barrons.pdf" and a "Brows..." button. The "Select Path Where to Save" section has a text box with the value "E:\PRIT" and a "Brows..." button. At the bottom, there are "Start" and "Disconnect" buttons. The status is displayed as "Status : Connected to Client..."

❖ Access Control (Shut Down):

This is the Shutdown Form for Administrator.



The screenshot shows a Windows-style window titled "ShutDown". Inside the window, there are two input fields: "Client IP Address" with the value "127.0.0.1" and a dropdown arrow, and "Port No" with the value "3232". Below these fields, there are three radio buttons: "Turn Off" (selected), "Restart", and "Log Off". At the bottom of the window, there is a "Perform" button and a "Status:" label.

❖ Message to Server:

This is the Message to Server Form for Remote Agent Clients.



The screenshot shows a Windows-style window titled "Message To Server". Inside the window, there is a text area containing the message: "Hello Sir..", "Please send me the file on", "E:\Sam\Docs Path.", and "Thank You...!!!". At the bottom of the window, there is a "Send" button.

❖ Message Broadcast:

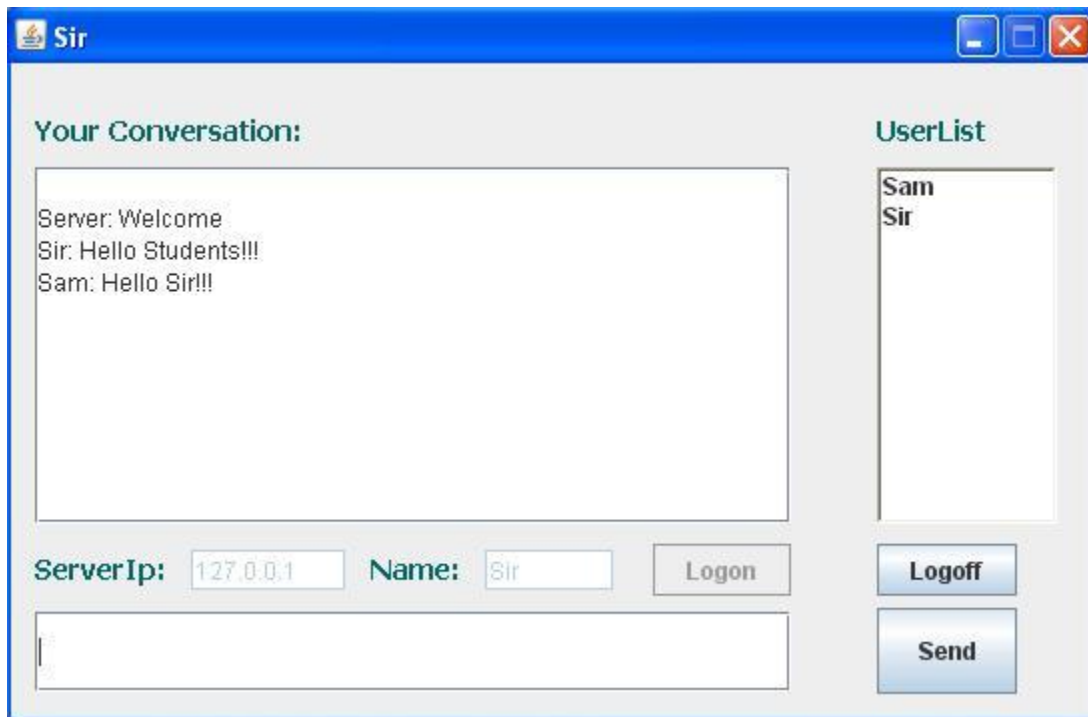
This is the Message Broadcast Form for Administrator.



The screenshot shows a Windows-style application window titled "Message Broadcast". The window has a blue title bar with standard minimize, maximize, and close buttons. Inside the window, there are two input fields at the top: "ClientIp" with the value "127.0.0.1" and a dropdown arrow, and "Port" with the value "3232". Below these, there is a checkbox labeled "Send To All" which is checked. To the left of a text area is the label "ServerMessage". The text area contains the message: "Hi....", "How R U????", and "Hope u can understand well.....!!". Below the text area is a button labeled "BroadCast". At the bottom of the window, there is a label "Status:" followed by a blank space.

❖ Chatting:

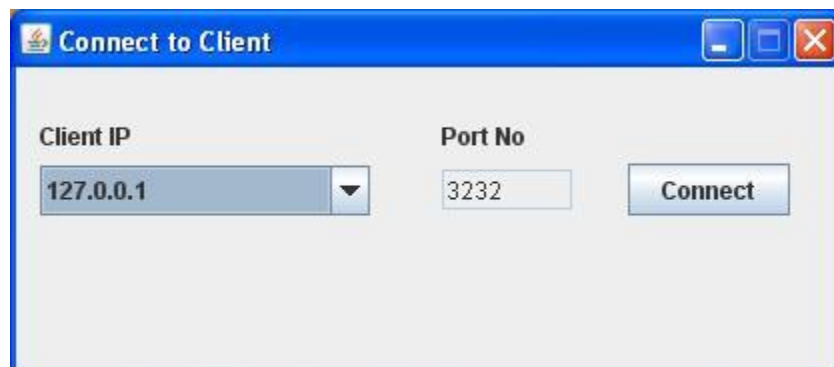
This is the Chatting Form for Administrator & Remote Agent Clients.



The screenshot shows a window titled "Sir" with a blue title bar. Inside, there are two main sections: "Your Conversation:" on the left and "UserList" on the right. The "Your Conversation:" section contains a text area with the following text: "Server: Welcome", "Sir: Hello Students!!!", and "Sam: Hello Sir!!!". Below this text area are input fields for "ServerIp:" (containing "127.0.0.1") and "Name:" (containing "Sir"), followed by a "Logon" button. Below the "Name:" field is a large empty text input box. The "UserList" section contains a list box with "Sam" and "Sir" listed. Below the list box are "Logoff" and "Send" buttons.

❖ Connect to Client for Remote View:

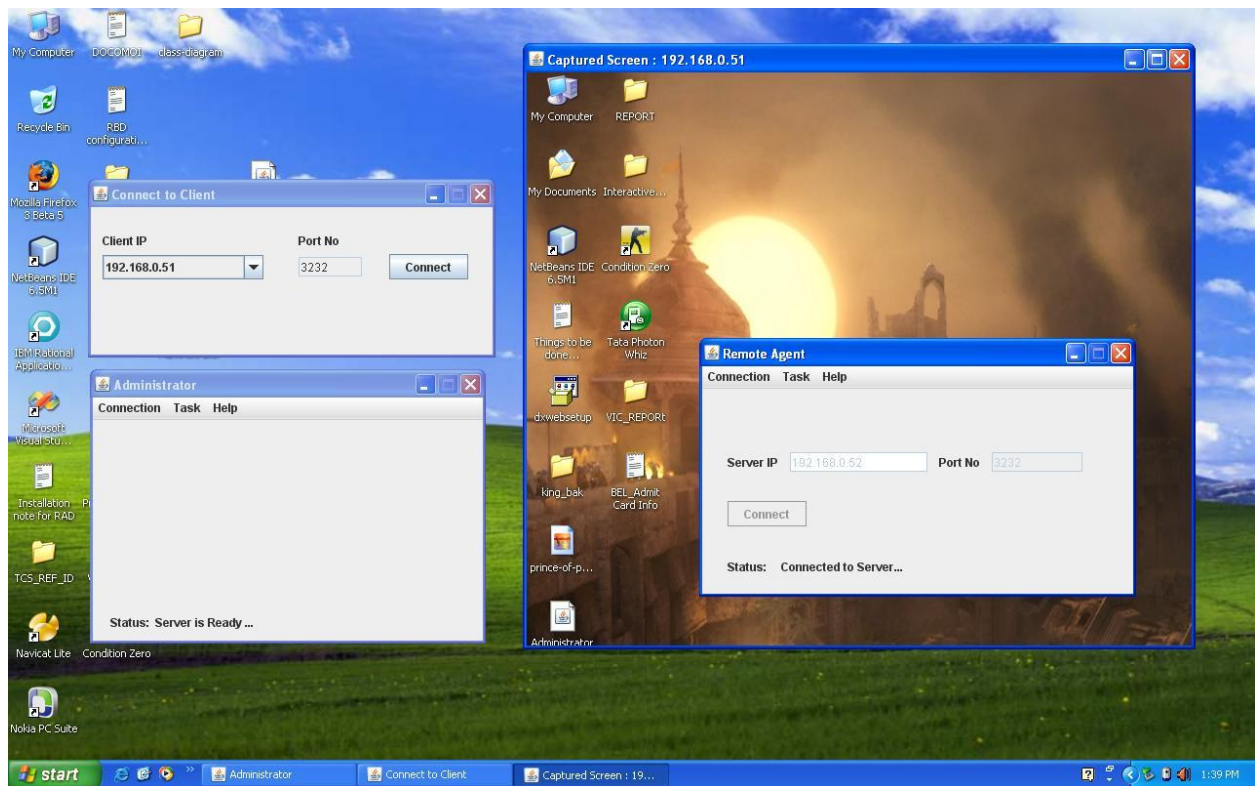
This is the Connect to Client form for Administrator for Remote View.



The screenshot shows a window titled "Connect to Client" with a blue title bar. Inside, there are two input fields: "Client IP" with a dropdown menu showing "127.0.0.1" and a "Port No" field with the value "3232". To the right of these fields is a "Connect" button.

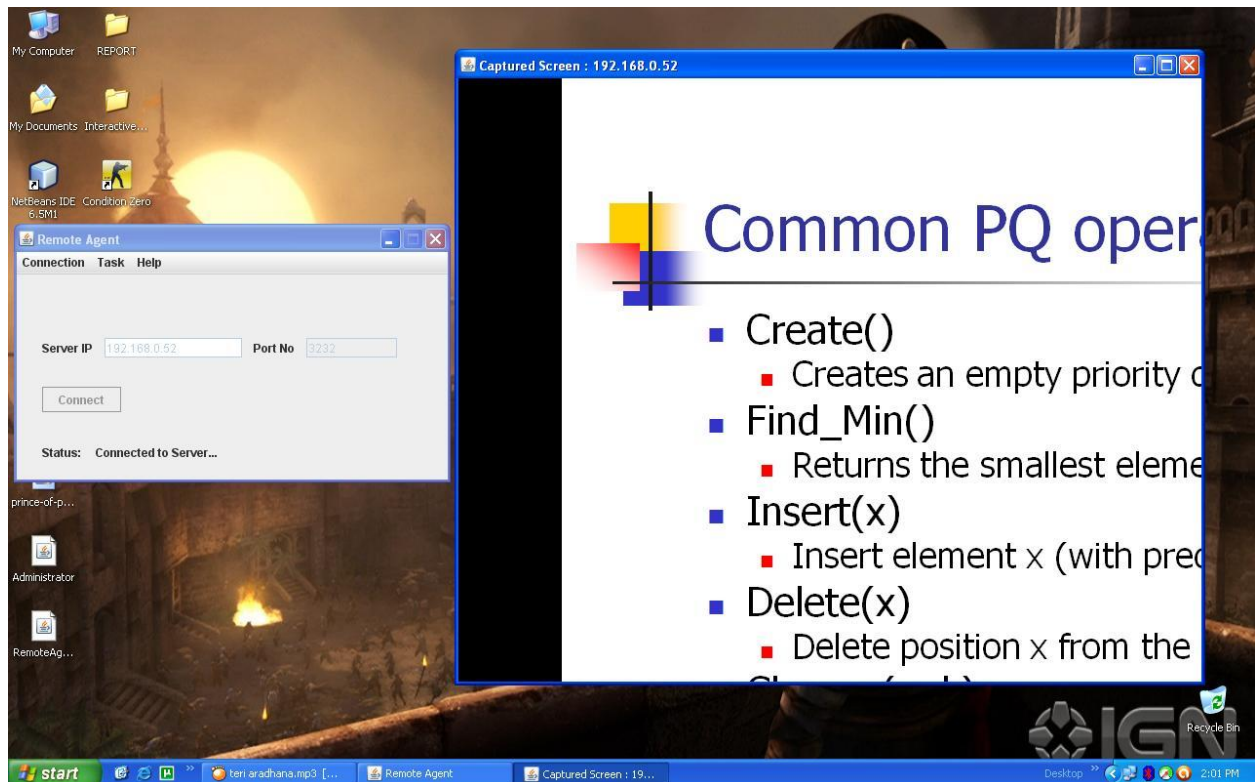
❖ Remote Agent Client Screen View:

This is the Remote Agent Client Screen View for Server. It displays what the Client is doing at current time.



❖ Server Screen View for Clients:

This is the Screen Capture View form for Remote Agent Clients to see the Sever Screen when server gives the permission.



3.2 Testing & Verification

What is Testing:

Testing a program consisting of a subjecting a program to a set of test inputs and observing if the programs behave as expected. If the program fails to behave as expected then the condition under which failure occurs are noted for a later debugging and correction. The following are some commonly used term associated with testing.

- A failure is a manifestation of an error (or defect or bug) but the mere presence of an error may not necessarily lead to a failure.
- A test case is a triplet [I, S, O] where,
 - I – Data input to the system,
 - S - The state of the system at which the data is input,
 - O - Expected output of the system.
- A test suite is the set of all test cases which a given software product is to be tested.

Testing Plan:

- Testing is the last stage before release of a system. During the testing we try to make sure that software does exactly what is supposed to do.

Testing Strategy:

❖ Unit Testing:

The developed software has three main modules:

1. File Transfer/Downloading
2. Message Broadcast

3. Chatting
4. Remote View
5. Session View
6. Access Control

Here, all the modules have been tested individually for the different inputs and expected output has been received from the application.

❖ Integration Testing:

After unit testing, all the modules are interacted and tested for the different set of inputs and desired output is obtained from the system. In this way the complete application has been tested, all the modules integrated together.

❖ System Testing:

Here in the application, “Virtual Interactive Classroom”, we have tested the whole application with all possible sets of inputs and expected outputs have been received.

- Alpha Testing:

Alpha testing refers to the system testing carried out by the test team within the developed organization.

- Beta Testing:

Beta testing is the system testing performed by a selective group of friendly customers.

- Acceptance Testing:

It is the system testing performed by the customer to determine whether to accept or reject the delivery of the system.

Testing Method:

Testing Methods are categorized basically in two forms:

White-Box Testing :

- Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function, is referred to as White-Box testing.

Black-Box Testing :

- Knowing the internal working of a product, tests can be conducted to ensure that all gears mesh i.e. internal operations are performed according to specifications and all internal components have been adequately exercised.

Verification:

Finally, we obtain various verified results that show the proper functioning of the modules of the Virtual Interactive Classroom.

All the tests carried on lead to minor problems and errors which we solved at that stage to obtain an almost error-free but of-course a limitation oriented Load Balancing System.

3.3 Conclusion

We have achieved the task of this project that is to manage a large network and create an environment like virtual interactive class room.

Also all the connected clients (students) can interact with each other and Admin (teaching faculty) through the chatting module.

We have successfully tested the all the functionalities of the modules like in file transfer and downloading module we have successfully downloaded the file from the clients to the server and successfully transferred the file from server to the clients.

Message broadcasting is also successfully worked form server and client can also send the message to server through message to server facility.

Server can shutdown, restart and logoff the desired client machine. This module is specifically worked in windows OS environment.

Server can see the client screen at current time easily and different clients can see server screen if server allows. So now all the students can easily see the presentation taken by the teaching faculty on their on machine.

So now this application works in any machine and provides an easy way to the teaching faculty and students to interact with each other.

3.4 Future Enhancement

This defines the thing that can be included in the project in order to make it professional and have been not included due to lack of the time.

- Fast response time on low latency networks including slow dial up connections.
- Full-featured secure file transfer lets you access files on the remote computer and securely transfer them between local and remote computers in any direction.
- Clipboard transfer automatically or manually synchronizes the clipboards of the remote and local computers making it easy to copy and paste text between two machines.
- Remote PC shortcuts let you create file-based shortcuts to remote computers. Double clicking a shortcut takes you directly to the remote computer.
- Resolution change option lets you view the entire screen of the remote computer
- Command-line interface lets you automate such tasks as installing the software, controlling the server, or running the client.
- Server icon in the system's tray provides single click access to the server window and all of its settings.
- Video streaming of remote & voice transferring machine screen can be implemented.

3.5 References

Books:

- ✓ Herbert Schildt, “*The Complete Reference JAVA2*”, 5th Edition, Tata McGraw-Hill.
- ✓ Troy Bryan, “*Remote Method Invocation JAVA*”, 2nd Edition

Websites:

- ✓ www.java.sun.com
- ✓ www.google.com
- ✓ www.hpjava.org