**Report of**

Monitoring and Communication System

Version 1.0

Prepared by Phanindra V.V.D, Shreyan Goswami and Vansh Jain

Department of ICT

24/04/2015

**Index**

1. Introduction.............................................................................................
2. Problem Statement of the Project............................................................
3. Software Requirement Specification.......................................................
4. Data Modelling........................................................................................
5. Analysis Modelling..................................................................................

5.1 Use-Case Diagram…………………………………………………………….....

5.2 Activity Diagram………………………………………………………………...

5.3. Class Diagram……………………………………………………………..........

5.4 Sequence Diagram…………………………………………...………………….

5.5 State-Chart Diagram……………………………………………………….……

1. Software Testing….…………………………………………………….

6.1 Unit Testing……………………………………………………….…………….

6.2 CFGs…………………………………………………………………………….

1. Snapshots of the Project Output………………..………………………
2. Importance of the Project……………………..………………………..

**1. Introduction**

Monitoring and Communication System is designed to help the user in computer labs. It uses TCP/IP protocol to connect two or more PC’s. TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination. It gives users a privilege to directly transfer data to other computers through Intra-Networking. The data will be transferred in a secure format so that no one can sniffer the content.

**2. Problem Statement**

The monitoring and communication system allows users to connect to a single server and send messages, transfer files and screenshots to other users. The system is designed to facilitate easy access to all its features. There are two types of users for the system- Teachers and Students. Both users have almost the same functionalities. The teachers have the facility to request for a screenshot of a student’s workspace. The server forwards this request to the student workstation. The system takes a screenshots of the student’s workspace and sends it to the teacher without the knowledge of the student in question. This allows the teacher to monitor the students. Whenever a teacher connects to the server she will obtain a list of students connected to the server. A student on the other hand obtains a list of connected teachers. This list is updated dynamically. Users can choose the person to whom they want to send data with the restriction that a student cannot communicate with another student. All the interaction is done via GUI.

3. Software Requirements Specification

for

Monitoring and Communication System

Version 1.0

Prepared by Phanindra V.V.D, Shreyan Goswami and Vansh Jain

Department of ICT

24/04/2015

Table of Contents

Table of Contents 1

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 2

1.4 Product Scope 2

1.5 References 2

2. Overall Description 2

2.1 Project Perspective 2

2.2 Product Functions 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 5

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

4. System Features 6

4.1 Sending and receiving messages 6

4.2 Transfer of files 7

4.3 Graphical User Interface (GUI) 7

4.4 Screenshot 7

5. Other Nonfunctional Requirements 8

5.1 Performance Requirements 8

5.2 Security Requirements 8

5.3 Software Quality Attributes 8

6. Other Requirements 9

**Appendix A: Glossary………………………………………………………………………..9**

# Introduction

## Purpose

The purpose of this software is to ease the work of teacher in supervising the activities of student in the lab. It will allow the students to communicate with teacher in an efficient way and also in transmission of data between them.

## Document Conventions

Font-style: Times New Roman

Font size (paragraph): 12

Font size (headers): 14

Header font size: 12

Header font: Times New Roman Bold)

Diagrams: Use case diagram and Activity diagram.

Actors: Student, Professor and Server.

## Intended Audience and Reading Suggestions

The document is intended for developers, project managers, users and testers. The SRS contains various information about the project implementation, its functionalities, design, interface requirements and scope.

The testers can directly jump to the activity diagram to get an idea of how the system will work at any point of its operation. They can have a look at the use case diagram to understand the features implemented by the system. The developer need to have a look at the user classes before proceeding to write code for the system. A high level summary of the product can be found in Section 2 for product managers.

All the abbreviations are listed in the glossary.

## Product Scope

MCS provides an interface to teacher where they can communicate with students concurrently.

It helps teacher to sit at one place and aid students with their doubts. With this the communication becomes more flexible and easier. It gives an opportunity to teacher to monitor student activities anonymously through which they can make sure that all students are utilizing the time wisely. The transfer of data between users becomes easier.

## References

Java 2 Complete Reference Book

<http://www.javaworld.com/>

<http://www.tutorialspoint.com/java/index.htm>

<https://docs.oracle.com/>

# Overall Description

## Project Perspective

For big computer labs, where student strength is more than 50, it is a difficult task for group of teacher to help all the students with their doubts. The time taken for a teacher to come to a student increases as no. of students getting doubt at the same time increases. This project is mainly targeted in solving this singular problem by acting as an interface between student’s PC and teacher’s PC where teacher can help multiple students simultaneously by being at one place.

It also helps teacher in tracking the progress and the activities of the student. By this, there is no need for the teacher to move around the lab looking over the shoulders of students to observe their activities. In the long run this will allow the teacher to get an overview of the students’ problem solving capabilities. By this the teacher can be able to guide students in a personalized way.

## Product Functions

* It allows the user to send and receive messages.
* Transfer of files between student and professor.
* Allow the teacher to request screenshots from the student anonymously.
* It also provides the facility to students to send screenshots to the professor.

## User Classes and Characteristics

The various users that will use the system are students, teachers and developer.

Student - Basic knowledge of using computer. Able to interact with the GUI for data transfer.

Teacher - Basic knowledge of using computer. Interacting with GUI to transfer data. Monitoring the students using the system.

The teacher will have the software running for the entire duration of the lab. If anyone accidently gets disconnected from the server, other people on the network won’t get effected. The teacher can receive messages from multiple students and reply to them one by one.

## Operating Environment

The software will be running on x86 based processors with Windows 7, 8 OS.

## Design and Implementation Constraints

The project specifically uses Socket programming concepts. As the language used is java, it requires Net Beans version 8 or above to easily debug.

* Maximum capacity of the system is Student side - 100 PCs and Teacher side – 100 PCs.
* Length of the message cannot exceed more than 200 characters.
* Multiple files from one client cannot be transferred at a time.

## Assumptions and Dependencies

* To run the software, JDK 1.6 or above will be required.
* All the systems should be connected in intranet through server using LAN interface.
* Server IP address should be static.
* Server and the clients should have a free port no. which is not used by any other application.
* User should know English language to interact with the GUI.
* File size should be limited to 5MB.
* File type is only txt.
* Image type is png.

# External Interface Requirements

## User Interfaces

The interface is designed using Java’s Nimbus UI Manager. It will contain text filed where user has to enter their name and connect button to get connect to server.

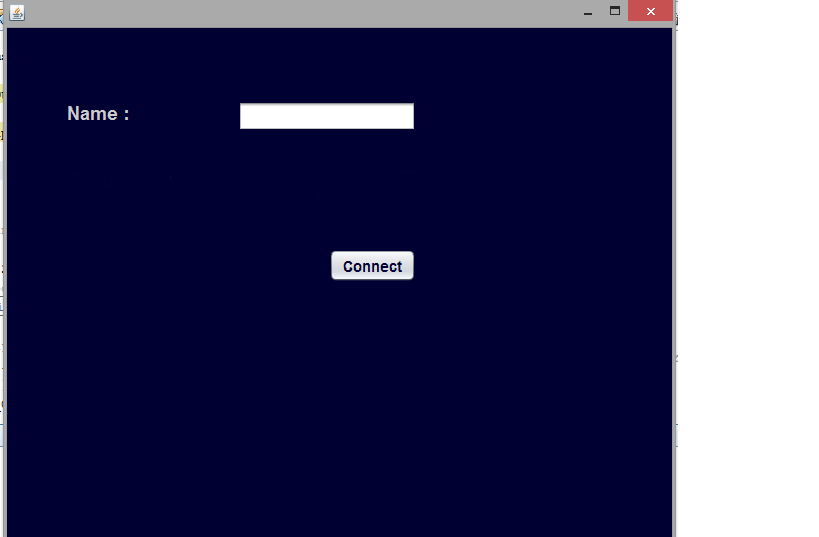
 Fig. 1 – Only for illustration purpose, not the final GUI.

Fig. 2 – Only for illustration purpose, not the final GUI.

In the next panel it will contain a text area where conversation will be displayed. The GUI will have a text field where the user can type a message. Buttons for choosing a file, clicking a screenshot and sending files/screenshot is provided. It will contain a list box from which the user can choose the recipient.

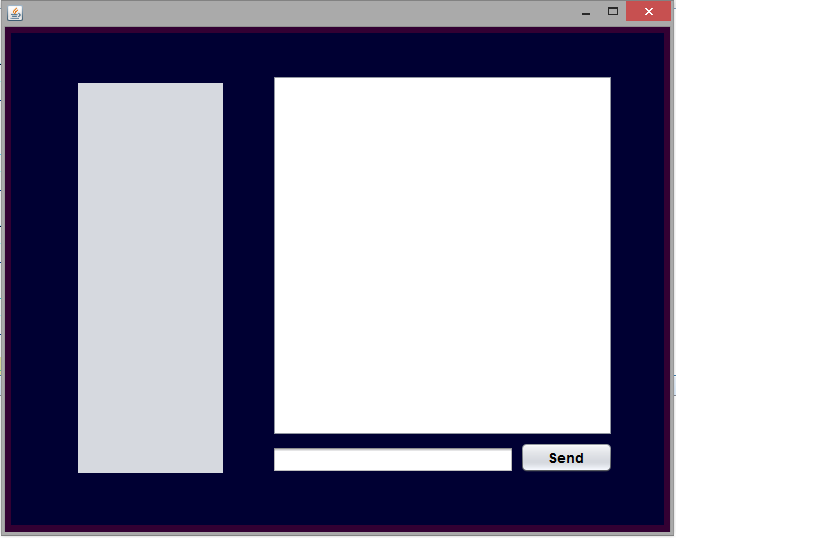


Fig. 2 - Only for illustration

## Hardware Interfaces

Minimum Requirements:

Client Side:

Processor: Intel dual core

RAM: 500MB

Disk Space: 100MB

Server Side:

Processor: Intel dual core

RAM: 2GB

Disk Space: 1GB

Recommended Requirements:

Client Side:

Processor: Intel Pentium I3 or above

RAM: 2GB

Disk Space: 500MB

Server Side:

Processor: Intel xenon or above

RAM: 4GB

Disk Space: 5GB

## Software Interfaces

3.3.1Development End

Programming language – Java

IDE – Net Beans

OS – Windows

## Communications Interfaces

Client and Server will be using TCP/IP Protocol.

# System Features

## Sending and receiving messages

4.1.1 Description and Priority

This feature of the software allows the user to exchange messages with one another through the server. It is the primary feature of the software and is assigned a priority of 9.

4.1.2 Stimulus/Response Sequences

The user first connects to the server and the server assigns each user with a unique ID. The assigned ID depends on whether the user is a teacher or a student. The sender chooses the recipients through a list box. The sender types the message and sends it to the server. The server accepts the message, searches for the recipient and transfers the message along with the source ID to the recipient. The program on the recipient PC will display the message on the tab which corresponds to the sender’s ID.

4.1.3 Functional Requirements

Req.1 The message should not be more than 65535 characters long.

## Transfer of files

### 4.2.1 Description and Priority

This is a feature that allows the users to transfer files between each other. The priority rating is 8.

4.2.2 Stimulus/ Response Sequences

The user chooses the file to be transferred via a dialog box and also the recipient of the file. The program then transfers the contents of the file to the server. The server receives the content and then transfers it to the recipient. The program on the recipient side receives the data and writes the content to a file.

4.2.3 Functional Requirements

Req.2.1 On the recipients side the file is saved to a pre-defined destination.

Req.2.2 Files can be transferred only between student and teacher.

## 4.3 Graphical User Interface (GUI)

4.3.1 Description and Priority

A program should have good interface with which the user interacts with the program. Hence the priority for the system feature is high and is assigned a value of 8.

4.3.2 Stimulus/ Response Sequences

The user will be greeted with an initial login screen where they input their name. This name will be visible to the teacher if the user is a student and vice versa. After they login, they are treated with an interface which allows them to send messages, take a screenshot and transfer files.

4.3.3 Functional Requirements

Req.3.1 The list box cannot have more than 100 items.

Req.3.2 User must login to access the messenger page.

Req.3.3 The user shutdowns the connection and closes the application by clicking on the disconnect button.

## 4.4 Screenshot

4.4.1 Description and Priority

This is a medium priority system feature with a priority value of 6. This allows the user to take a screenshot of the current workspace and save it. The user has the option to send the image to another user at a later point of time.

4.4.2 Stimulus/ Response Sequences

The user clicks a button to take a screenshot of the current workspace. The picture is automatically saved in the PNG format. Now he can choose the file and send it.

4.4.3 Functional Requirements

Req.4.1 The screenshots are save to a pre-defined system location.

Req.4.2 The student can only share the image with the teacher and vice versa.

# Other Nonfunctional Requirements

## Performance Requirements

* The server should be able to handle requests from many clients simultaneously.

## Security Requirements

* Before using the software the user has to enter their name.
* Every user is assigned a unique ID when they are connected to the server. This ID is used to differentiate between multiple users.

## Software Quality Attributes

* If a client is accidently disconnected from the server, the other clients are not affected by it hence making the system more reliable.
* The working of the software is unaffected if more clients are added.

# Other Requirements

Appendix A: Glossary

This section defines all the technical terms used in the SRS.

MCS – Monitoring and Communication System

TCP – Transmission Control Protocol

IP – Internet Protocol

Port – It is a software construct serving as a communications endpoint in a computer's host operating system.

IDE – Integrated Development Environment

LAN – Local Area Connection

Protocol – defines rules and conventions for communication between network devices.

Socket – It is a class defined in JDK which allows to establish connection between two computers.

**4. Data Modeling**

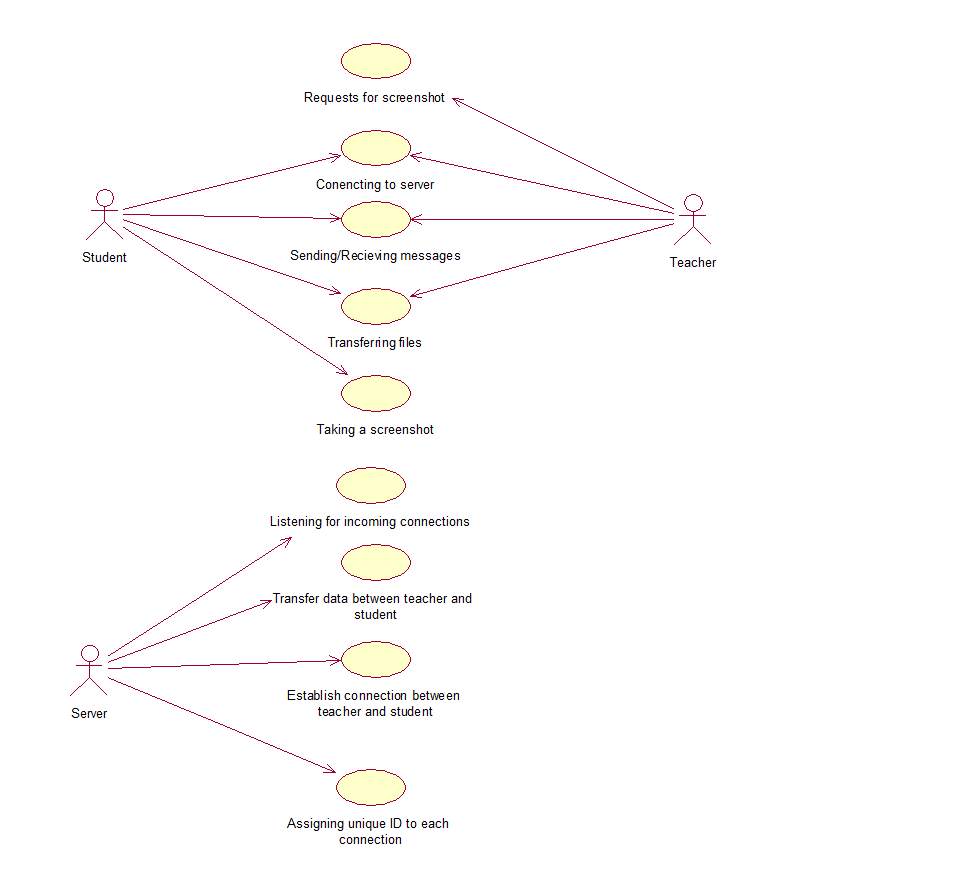
1. **Messages**

The transmitted message are displayed onto the GUI in ‘TextArea Field’. Users can see received and sent messages from other users. Different user messages are displayed onto different Text Areas.

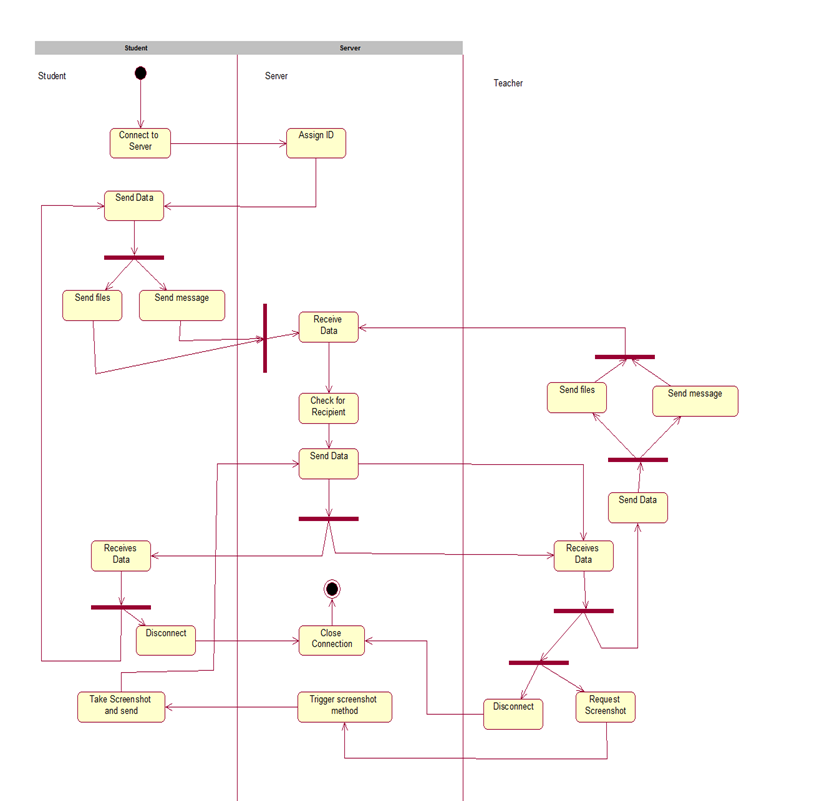
1. **Files and Images**

The received files and images from other users are saved onto a pre-defined memory location in the hard-disk.

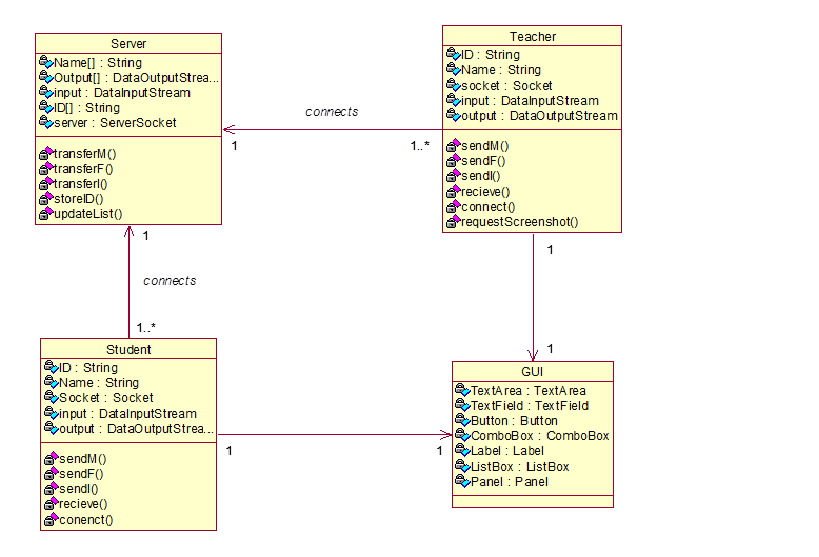
**5. Analysis Modeling**

**1. Use Case Diagram**

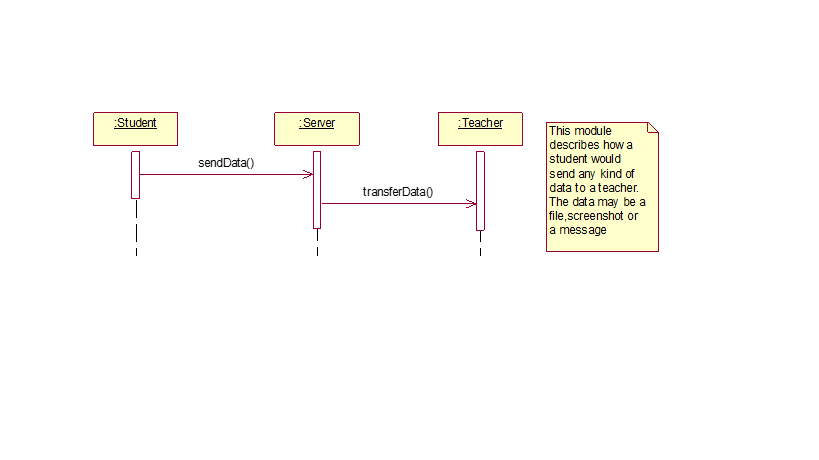
**2. Activity Diagram**

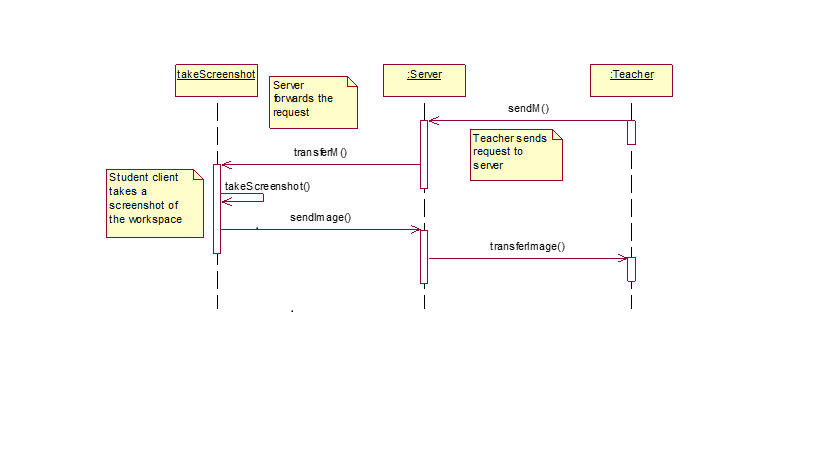


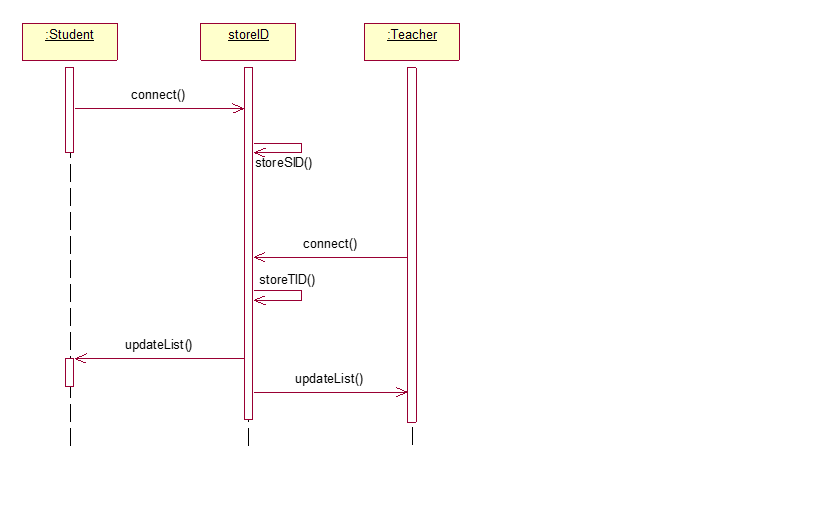
**3. Class Diagram**

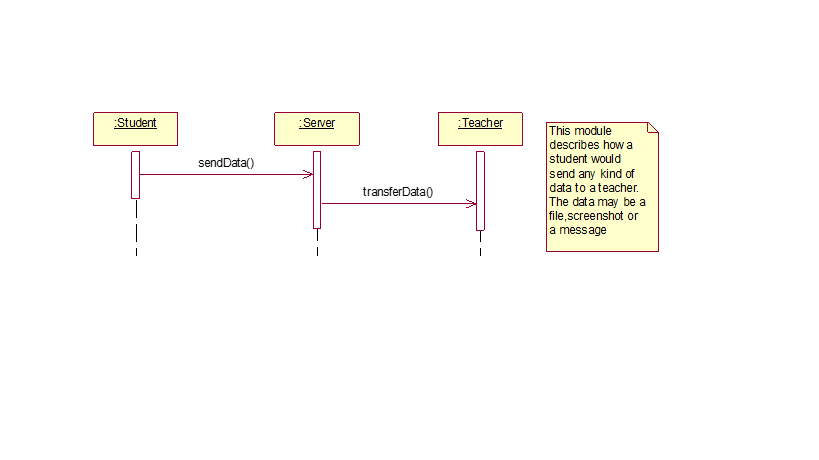


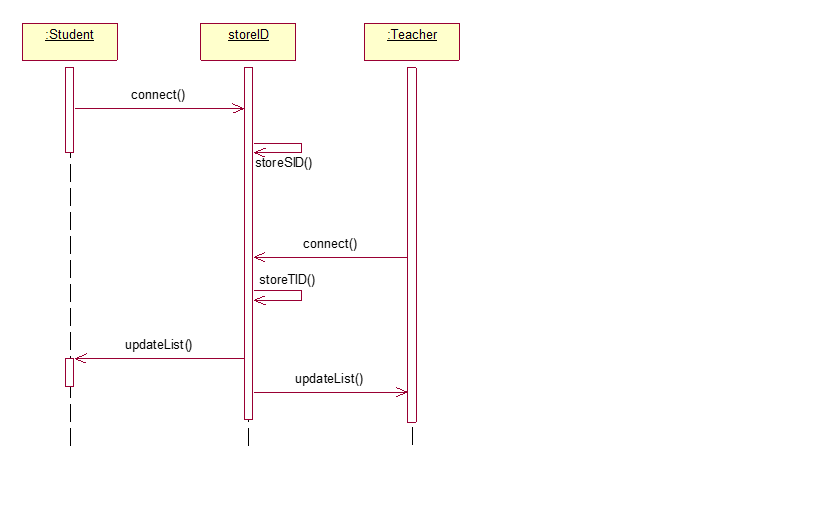
**4. Sequence Diagram**

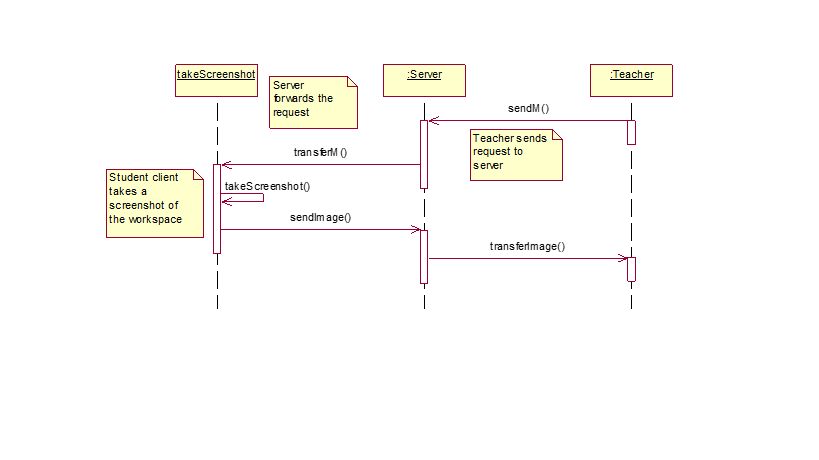




****





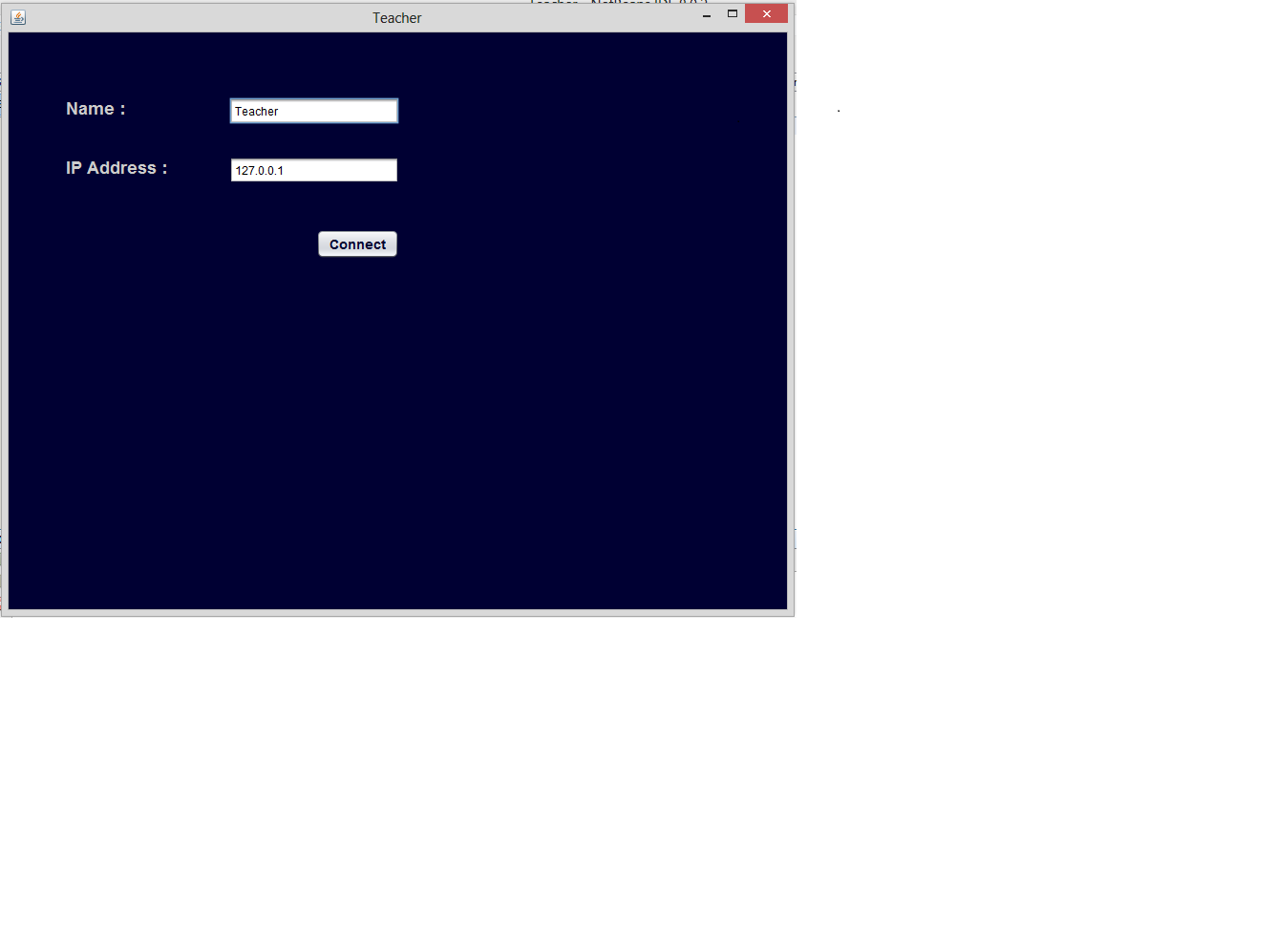


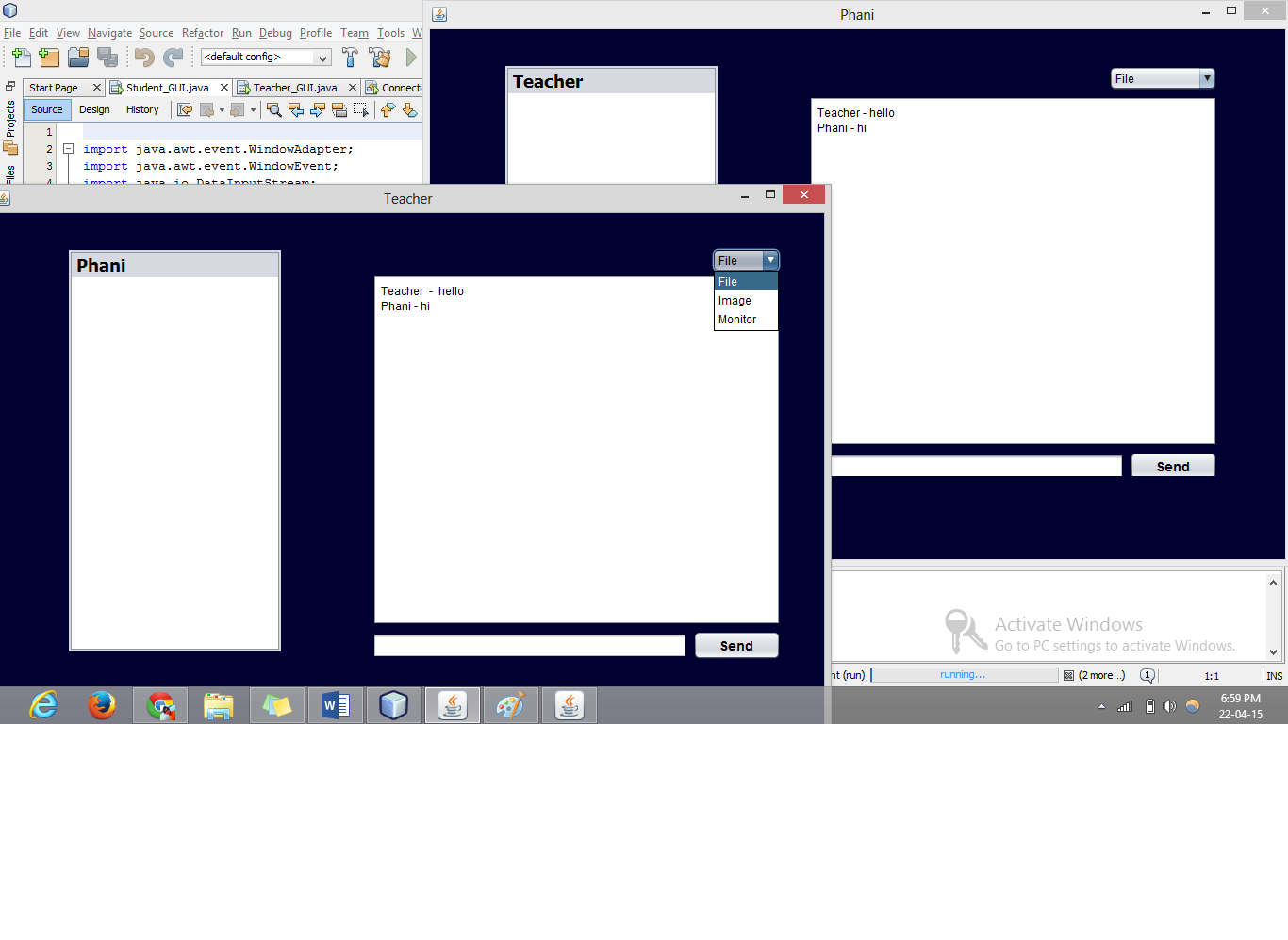
**5. State Chart Diagram**





**7. Snap shots of MCS**





**8**. **Importance of MCS**

This project is mainly used to reduce the efforts of teacher in computer labs. It gives them an opportunity to clarify the doubts of multiple students simultaneously by being at one place. At the same time, student doesn’t need to go to teacher or wait for them to ask queries. Students and teacher can use this system to transfer files and images. The teacher can monitor students without leaving their desk. This is done by allowing teachers to obtain a screenshot of students’ PC.