

1. Introduction

1.1 Introduction

Quick Share is a one click, pre-configured and easy to use file sharing application meant for sharing files quickly amongst users. Quick Share application uses fast Internet services available today such as 3G, Wi-Fi etc. for connecting the users in order to share a file. Using this application will help its users to get rid of flash storages and other hardware storage devices for transferring files whenever required.

Quick Share desktop application comes with facility to connect a group of users for file sharing purposes instantly. The application is very convenient to use as all the settings required for connecting users would be pre-configured. This allows user to send/receive file with just one click. The initial version of application supports Microsoft WindowsTM platform.

Following are salient feature of Quick Share:

- Simple, clean Graphical User Interface.
- Just a click for sharing a file in the group.
- Database for viewing sent/received files.
- Instantaneous file transfer using effective bandwidth of Internet connection.

1.2 Technologies to be used

- Front-End: Java (J2SE) version 1.7
- IDE: Eclipse Helios IDE 4.2
- Back End: XML

2. Literature Survey

Following are the sources that were studied for developing the application and its underlying research work:

Text references:

[1] Java: The Complete Reference by Herbert Schildt

Chapter 20. Networking

Chapter 21. The Applet class

Chapter 22. Event Handling

Chapter 23. Introducing the AWT: Working with Windows, Graphics, and Text

Chapter 24. Using AWT Controls, Layout Managers, and Menus

Web references:

[1] www.docs.oracle.com/javase/tutorial/networking

[2] www.onjava.com/pub/a/onjava/2003/12/03/p2psockets.html

[3] www.springer.com/engineering/signals/journal/12083

[4] www.tools.ietf.org/html/rfc959

[5] www.thenewboston.org

3. Requirement Specification

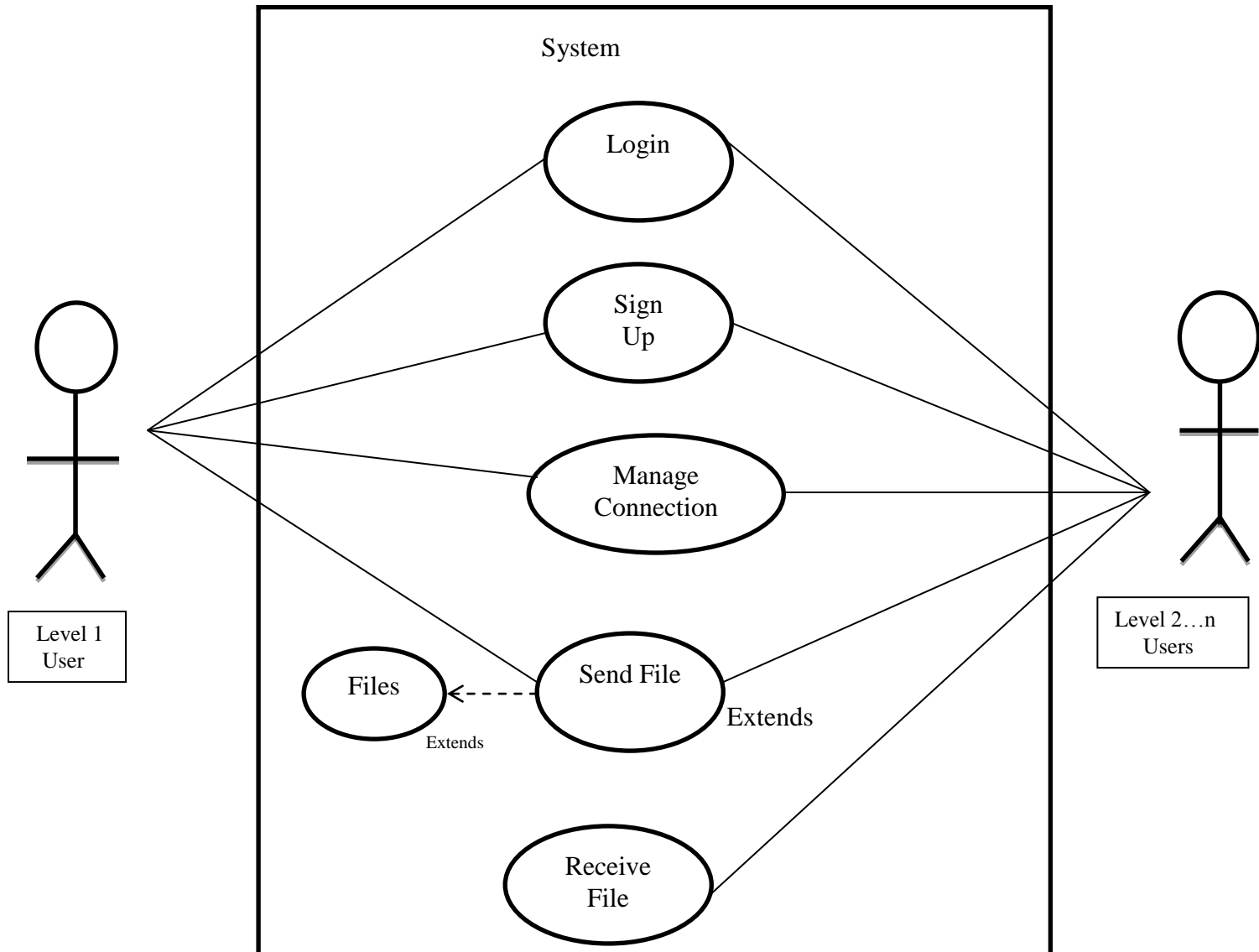
1.1 Functionalities

The main idea behind developing Quick Share is to provide faster file sharing between two or more peers in a network over the Internet. The major functionalities would include:

- Firstly the data/files to be shared would be categorized according to their extension and size. Then transferring the files directly.
- Underlying communication protocols would involve TCP for a reliable and pre-established connection. The sharing of files would be done using FTP between the peers in the network.
- Authorizing the peers involved in the network so that integrity of data is maintained.
- Maintaining the hierarchy of peers in a graph form so that congestion in the network can be minimized and better use of available Internet bandwidth can be achieved.
- Keeping the record of involved peers from the network in the database.

1.2 Use Case Model Diagrams

The Use Case model Diagram for the software is shown below along with all classes and actors in the system.



1.3 Interfaces

Software:

Front End: The GUI for the Front End will be developed in

- Java 1.7 (J2SE) - Level 1 Peer i.e. Server node of hierarchy.

Back End: The database used for storing all the files and necessary information will be

- XML – A structured modeling language.

Hardware:

Level 1 peer or Server Node

Operating System	Processor	Free Disk space	RAM
Windows XP/7 or Higher	Intel Pentium IV 2.27 GHz or High	10 Gigabytes of free Hard disk space	1 GB or more

Level 2 or Peer Nodes

Operating System	Processor	Free Disk space	RAM
Windows XP/7 or Higher	Intel Pentium IV 2.27 GHz or High	10 Gigabytes of free Hard disk space	1 GB or more

Communication:

The communication between peers in the network would be made by connecting them using high speed Internet facilities like DSL or dedicated Wi-Fi network.

Apart from these other components would include:

- A NIC for all nodes.
- HSPA/WCDMA: 900/2100 MHz for all nodes.
- Wi-Fi (802.11b/g) for all devices in in network.
- 3G Internet connectivity.

1.4 General Constraints

The general constraints for running Quick Share would be:

- JRE 7 on all nodes included in the network.
- Availability of high speed Internet connection so that key features of the software i.e. faster file sharing can be fulfilled.

1.5 Supplementary requirements

The additional requirements for developing the software are mentioned below:

Learning resources

Java Practical Learning resources- From various sources

TCP architecture- From RFC 793 as given at www.ietf.org/rfc/rfc793.txt

FTP architecture- From RFC 959 as given at www.ietf.org/rfc/rfc959.txt

4. Architectural Design

4.1 Introduction

Quick Share is a one click, pre-configured and easy to use file sharing application meant for sharing files quickly amongst users. Quick Share application uses fast Internet services available today such as DSL, Wi-Fi etc. for connecting the users in order to share a file. Using this application will help its users to get rid of flash storages and other hardware storage devices for transferring files whenever required.

Quick Share desktop application comes with facility to connect a group of users for file sharing purposes instantly. The application is very convenient to use as all the settings required for connecting users would be pre-configured. This allows user to send/receive file with just one click. The initial version of application supports Microsoft WindowsTM platform.

Following are salient feature of Quick Share:

- Simple, clean Graphical User Interface.
- Just a click for sharing a file in the group.
- Database for viewing sent/received files.
- Instantaneous file transfer using effective bandwidth of Internet connection.

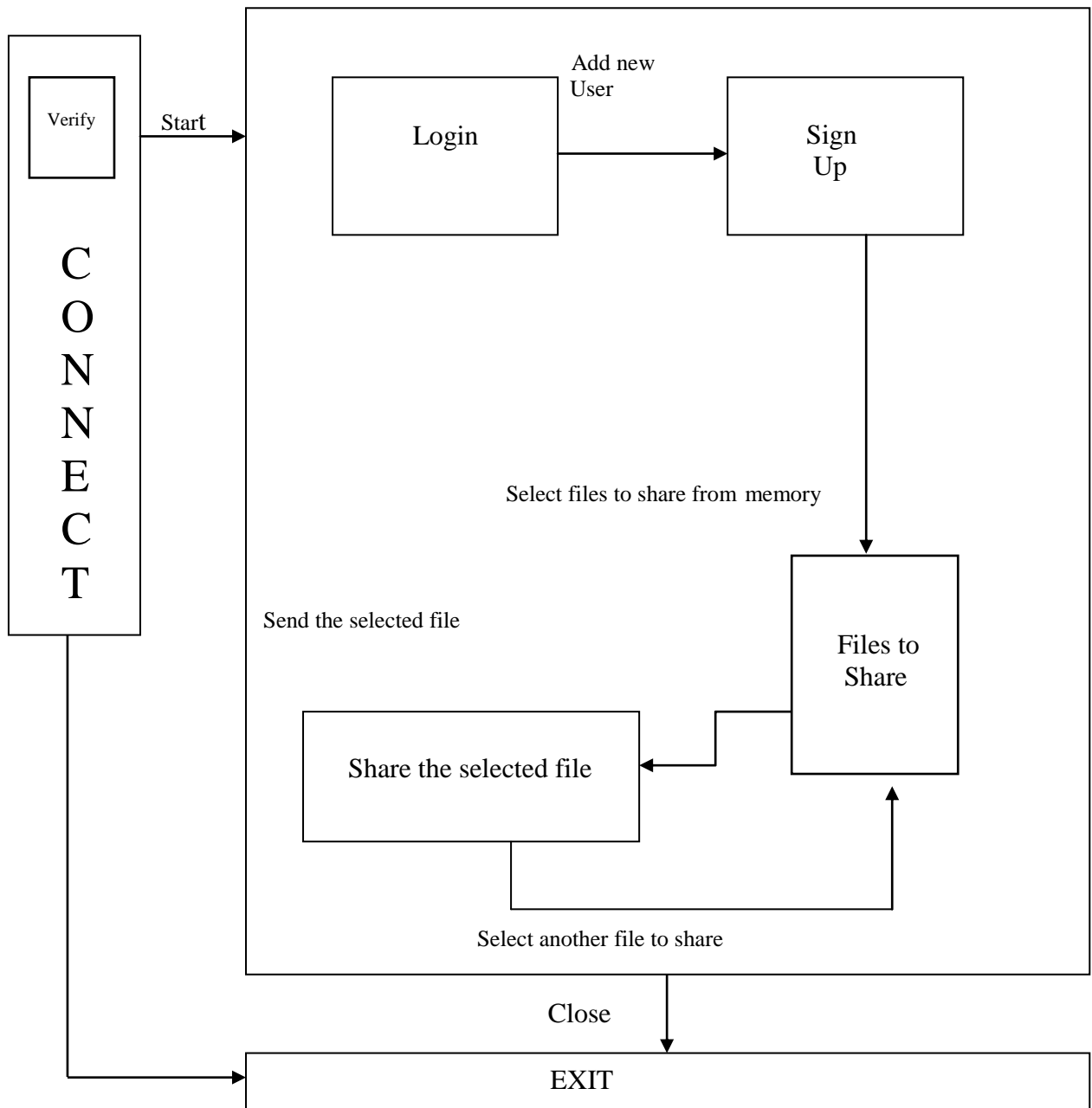
4.2 Functional overview

This section describes the basic functionality of the software. There are various stages involved in functioning of the software according to its architecture. The software is primarily based on FTP as mentioned in RFC 793. Quick Share uses FTP as its core and all the sharing of files is done using this protocol. The communication is done using TCP/IP protocol stack as mentioned in RFC 1180.

- Following are the functionalities of application:
- In the first stage, the file to be shared is selected from database.
- In contrast to other normal applications based on sharing of files, Quick Share uses pre-configured connections with receiver nodes to avoid various intermediate time consuming steps.
- Subsequently, the selected file is then transferred to receiver node using Internet connection.

4.3 Functional Block Diagram

This section deals with various Functional Block Diagram in accordance with architecture of the application.



Starting the application

This block has the functionality to start the application. After starting the application, the user has to authenticate the identity by providing a unique Username and a strong password. The authentication is necessary to secure the contents of database.

Set up the connection

This part establishes the connection available in the geographical area to connect to other users of application and share files. The connection is TCP/IP based Internet connection such as Broadband network.

Login of users

This group is used to login of users with whom files are to be shared. Login is necessary when the user is using for the first time. The number of users per group is fixed in order to speed up download/upload bandwidth.

Sign Up of users

The updation of an existing group helps the user to add new users or remove existing ones according the needs. This functionality extends the scope of application i.e. instead of sending files to a static group of users, the sender can update the group. This helps application to be used at variety of locations.

Selecting the file to send

After the user has set up the connection and has determined the senders to whom files are to be send, the next part comes where file that is to be sent if selected from the database. The database of the application stores the files from physical memory location of the sender's computer. The database also has a log to maintain the list of files that have been shared earlier.

Sending the file

Now after selecting the desired file to be shared, the user can send the file to the receiving group instantaneously.

Exit

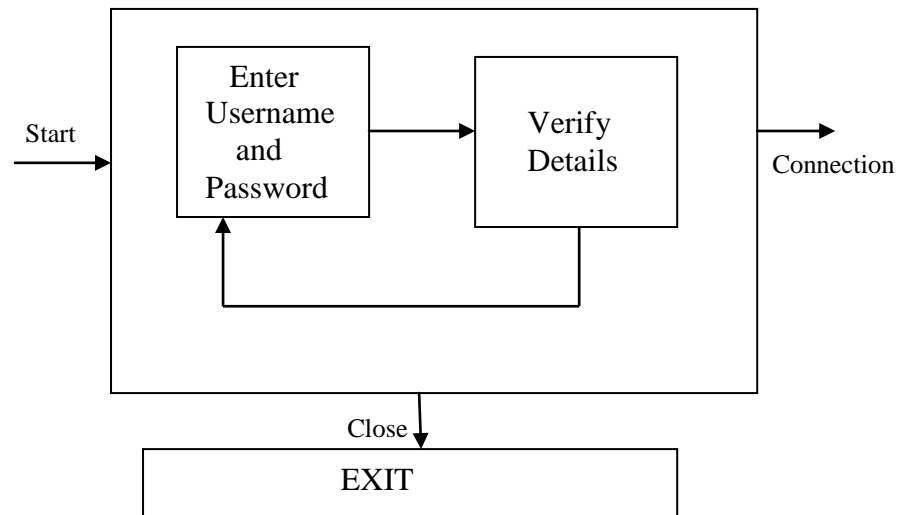
The application can be closed at any instant by the user.

4.4 Functional Partitions

4.4.1 Module Name: Login

The Login module is used to authenticate the user using the application.

4.4.1.1 Functional Block Diagram



4.4.1.2 Module Dependencies

4.4.1.3 Input Dependencies

The information needed to open the application is to be supplied to Login module. This information needed is:

Username: The unique username provides the user to separate his/her profile from other users of the application.

Password: The password is used to authenticate the users before they can use the application.

4.4.1.4 Output Dependencies

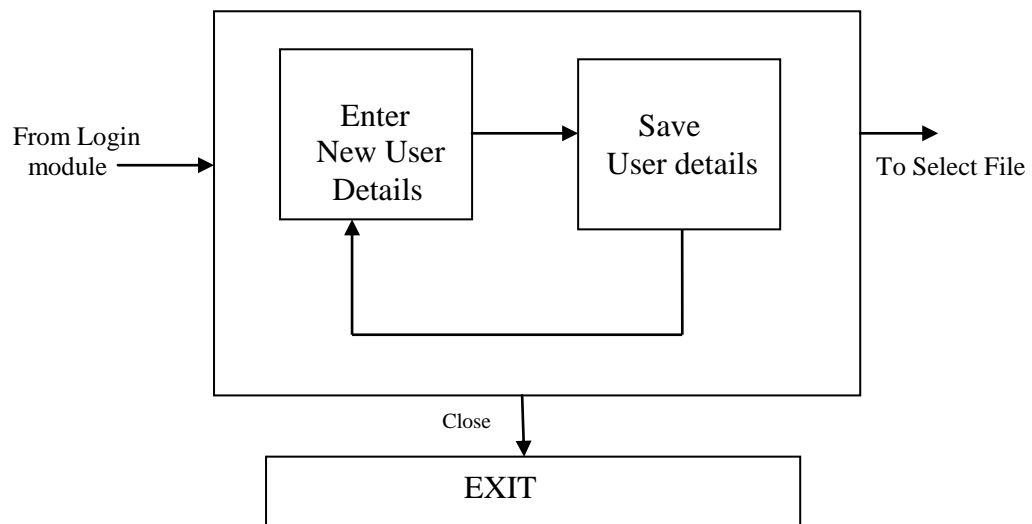
The output dependencies of Login module are described below:

Module	Data	Description
Share	Username, Password	Correct username should be entered to proceed to next module.
Sign Up	Username, Password	Correct username should be entered along with password to start application.

4.4.2 Module Name: Sign Up

The Sign Up module is used to create a set of users with a predefined size; together they form the end-users of the application. The file sharing is done between these user groups. The groups are created so that file can be transferred efficiently in a hierarchical manner from one group to other maintaining efficient use of connection bandwidth.

4.4.2.1 Functional Block Diagram



4.4.2.2 Module Dependencies

4.2.2.1 Input Dependencies

The input given to Create User Group module in order to create a new user group is given below:

Add New User: This option is inputted with a new user name.

Enter User Details: Here all the details necessary to connect a user with application are inputted. These details include IP address as a primary input. Other details can be Group name, Date of creation of group etc.

4.2.2.2 Output Dependencies

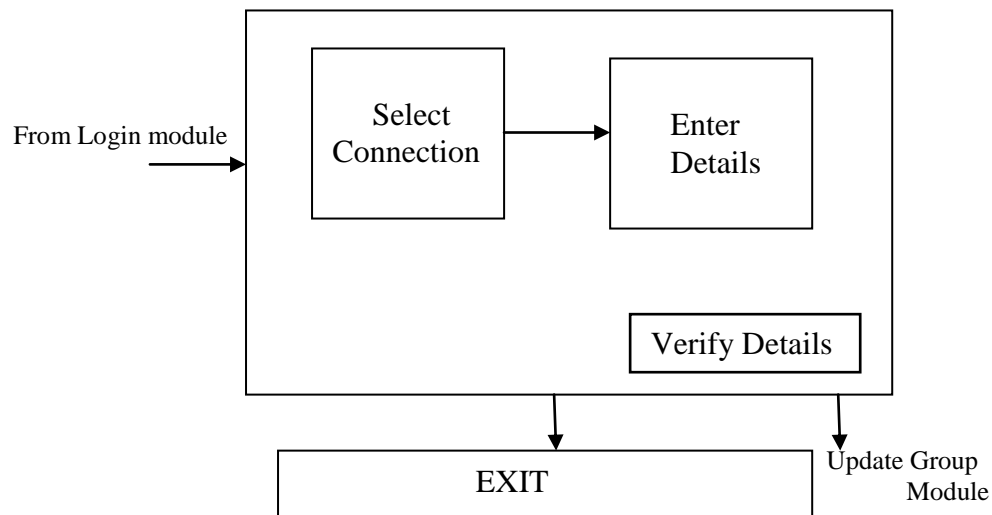
The output dependencies of Create User Group module are described below:

Module	Data	Description
Select File	File name	The name of file which is to be shared should be consistent with database.
Select File	File Details	All the details along with address of the file to be shared should be correct.

4.4.3 Module Name: Set Connection

The Set Connection module is used to connect the application with other users using the application in the network. This module performs the functionality of setting up the connection using available connections.

4.4.3.1 Functional Block Diagram



4.4.3.2 Module Dependencies

4.4.3.2.1 Input Dependencies

The user needs to input following details before proceeding to use the application:

Connection: The available server details are to be entered.

Connection details: The details for connection such as private/public area, Validation details of network etc. are to be supplied to this module.

4.4.3.2.2 Output Dependencies

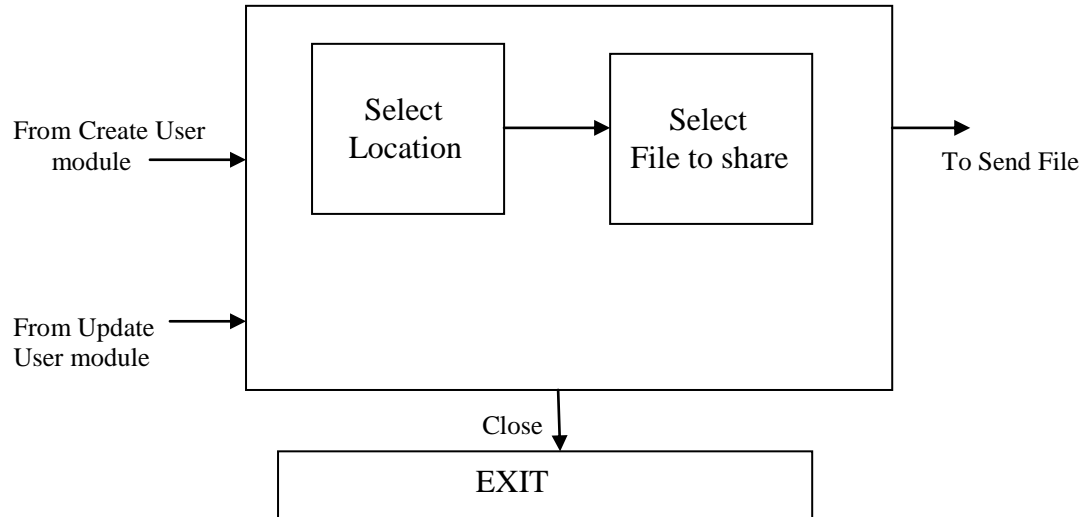
The output dependencies of Set up Connection module are described below:

Module	Data	Description
Login	Connection details	The user details primarily IP address should be consistent.
Sign Up	Connection details	Details like password for protected connection should be entered correctly.

4.4.4 Module Name: Select File

This module is used to provide a functionality under which the user can select the desired files which are to be shared with other user of the application.

4.4.4.1 Functional Block Diagram



4.4.4.2 Module Dependencies

4.4.4.2.1 Input Dependencies

Following details are to be inputted to Select File module in order to proceed to final module:

File location: The location of file where the file is stored in the computer should be provided and is essential to locate the file.

File: The file to be shared using the application is given as input to this module.

4.4.4.2.2 Output Dependencies

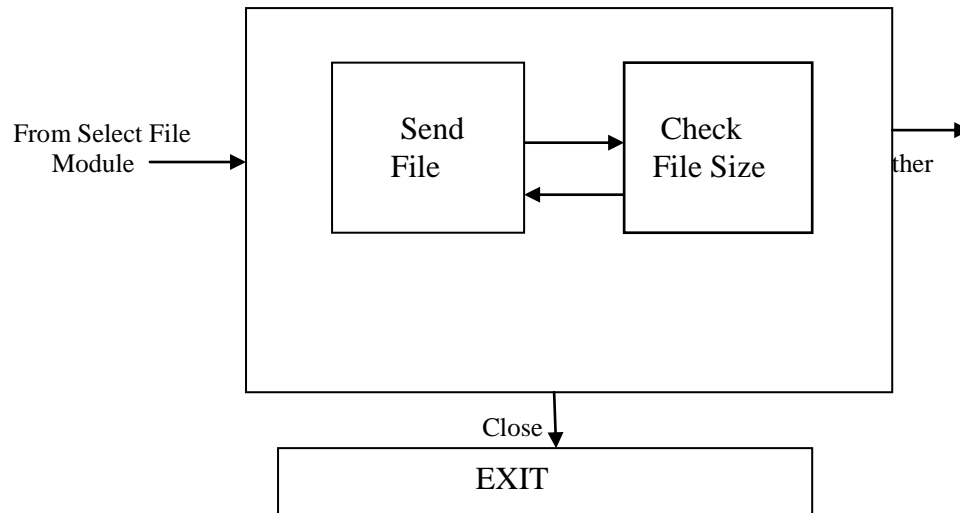
The output dependencies of Select File module are described below:

Module	Data	Description
Share	Filename	Correct filename should be entered to proceed to next module.
Select File	Filename	A file with size less than or equal to prescribed size should be there.

4.4.5 Module Name: Share

The share module is the main core of architecture of Quick Share application. This is due to the fact that once the configuration of setting up the connection, using a user group etc. has been done, then the configured setting are saved until the application is restarted. Now the user needs only single click (to select file) every time a file is to be shared with other users of the application.

4.4.5.1 Functional Block Diagram



4.4.5.2 Module Dependencies

4.4.5.2.1 Input Dependencies

The information to be inputted in this module is:

File: The file to be shared acts as input to this module. There should be at least one selection to transfer/share the file.

4.4.5.2.2 Output Dependencies

The output dependencies of Share module are described below:

Module	Data	Description
Select File	File	A file with size less than or equal to prescribed size should be there.

4.5 Data Description

4.5.1 Data Flow Diagram

The Data Flow Diagrams for the Quick Share Application are drawn below:

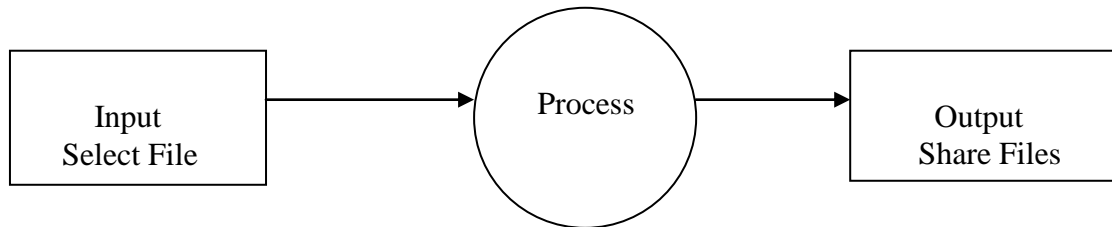


Figure: 4.1 Context Level 0 DFD

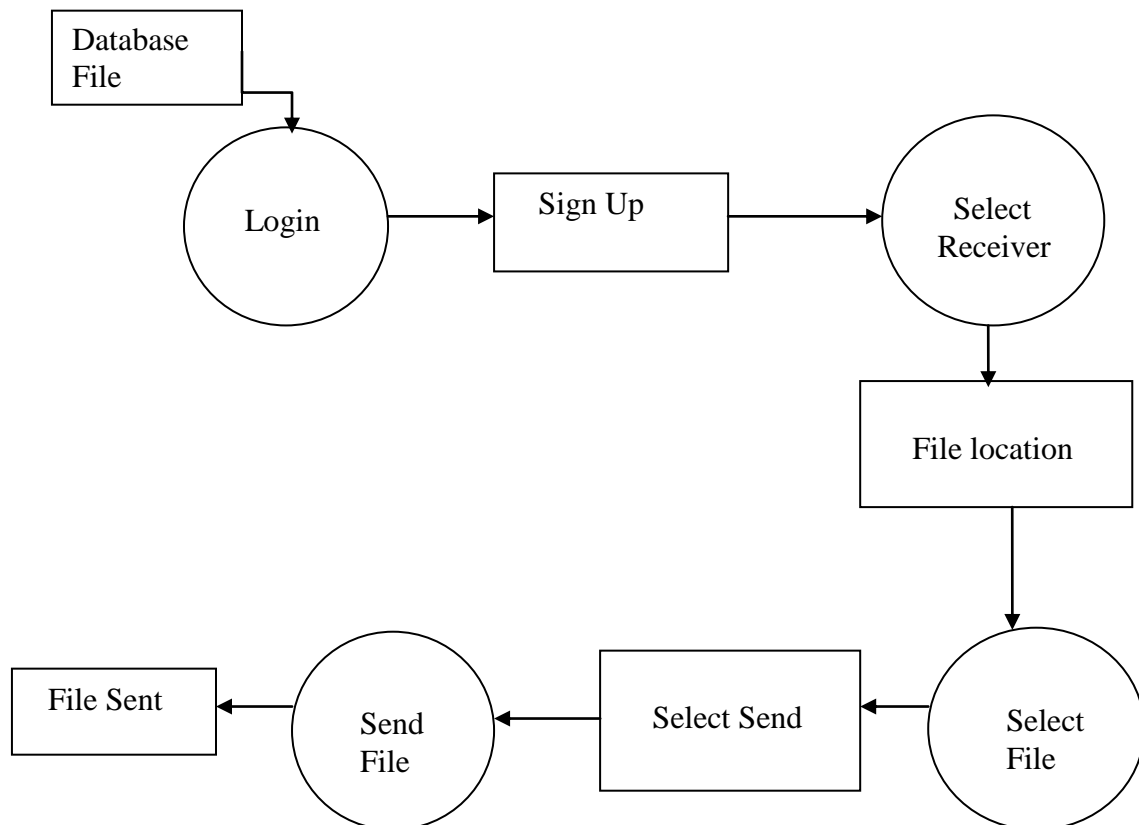


Figure: 4.2 Level 0 DFD

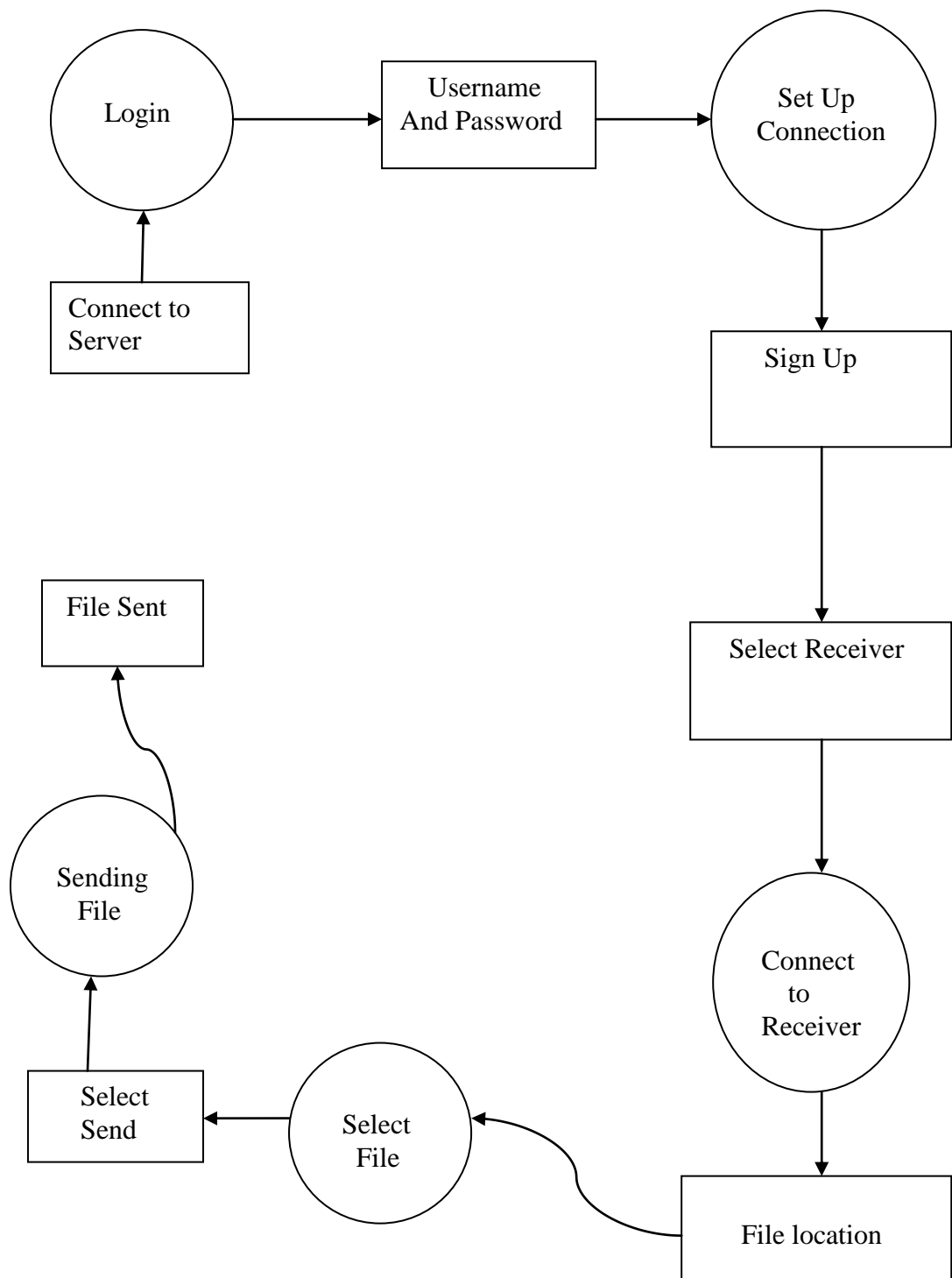


Figure: 4.3 Level 1 DFD

4.5.2 Data Structures Used

The primary data structure used to develop Quick Share application is Class. This is due to the fact that the front end being used for developing the application is Java 1.7 which is a pure Object Oriented programming language.

Following is the list of primary data structures to be used in developing the application:

- Class
- Array

4.5.3 Constant Definition

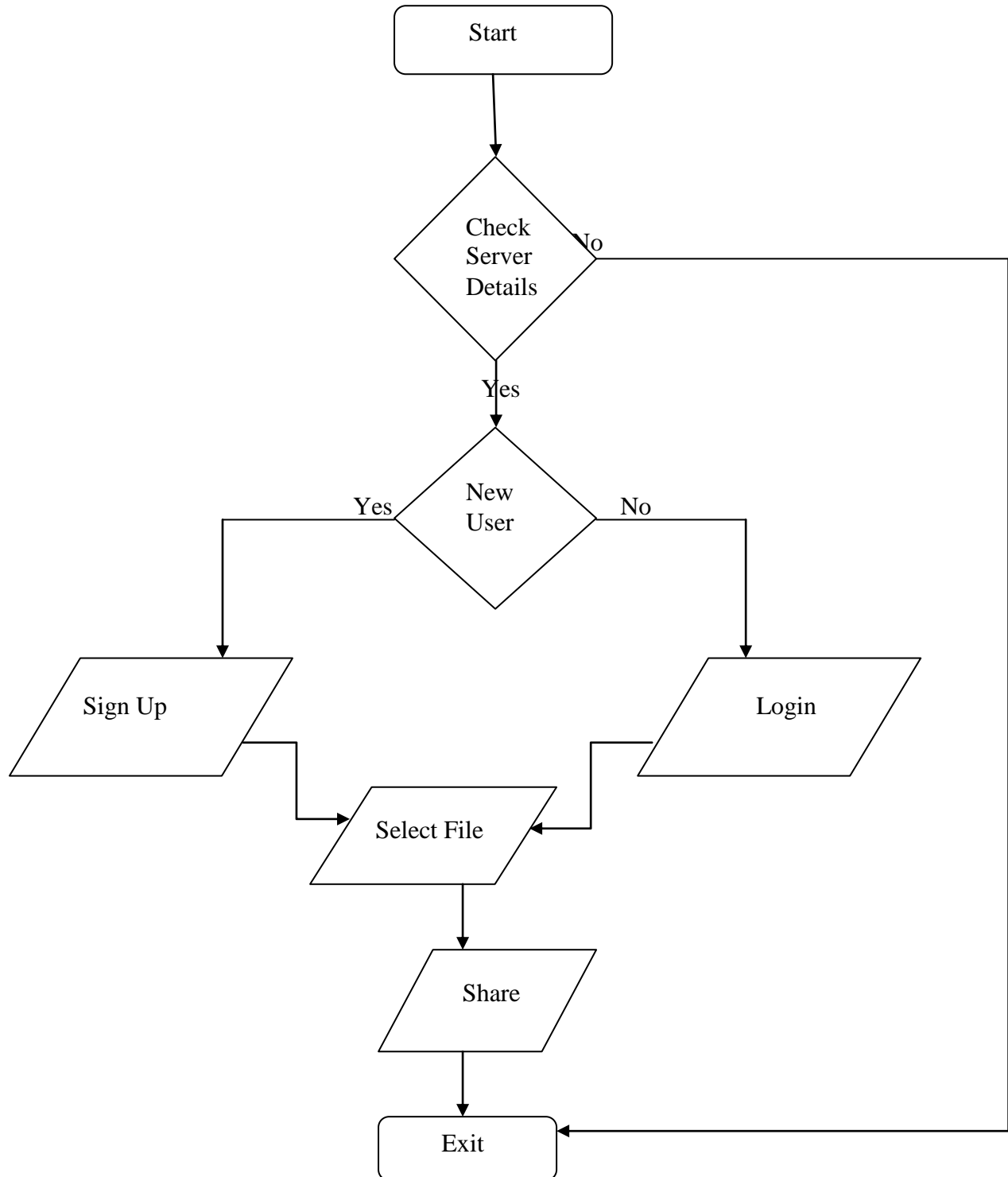
Certain constants are to be defined/declared in code of the application according to need of architecture. They are

- IP_Addr: For IP address of the users in the network that are intended to use the application.
- File_Size: For limiting the size of file to be shared.
- User_Max: Maximum number of users/peers in a group using the application.

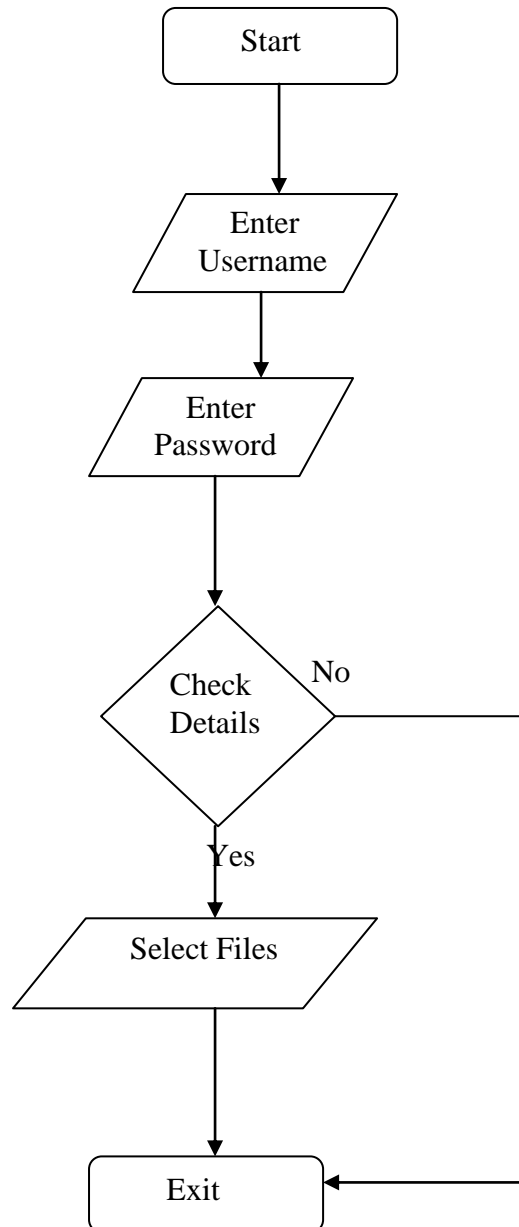
4.5.4 Flow Charts

Flow charts are necessary to visualize the flow of operations and algorithm used to develop the application/software.

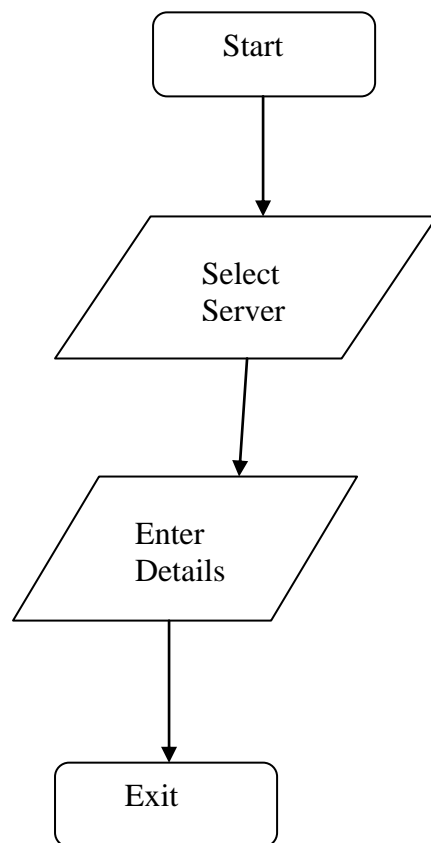
Flow Chart for complete application



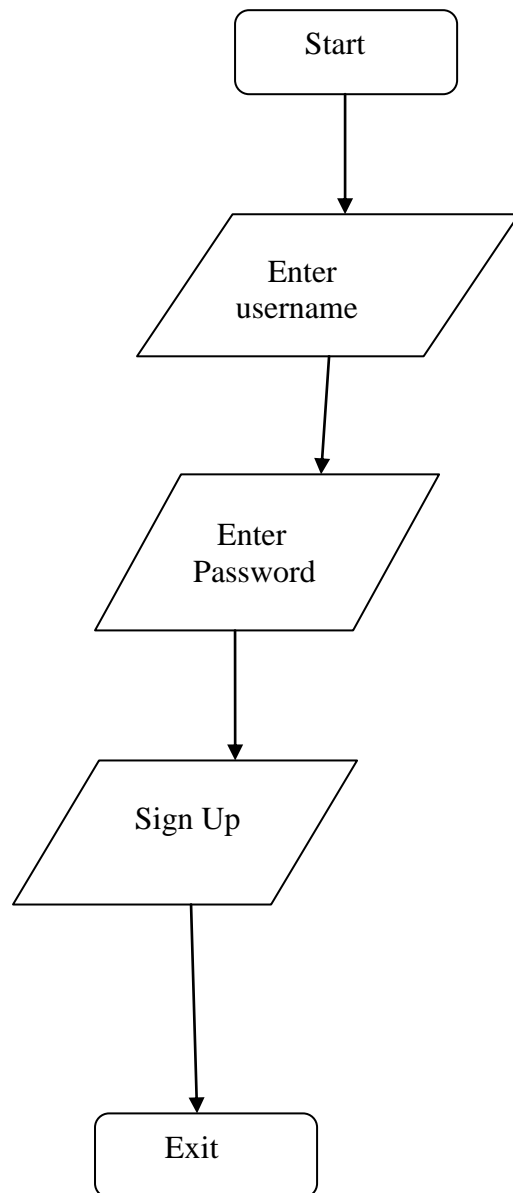
Flow Chart for Login module



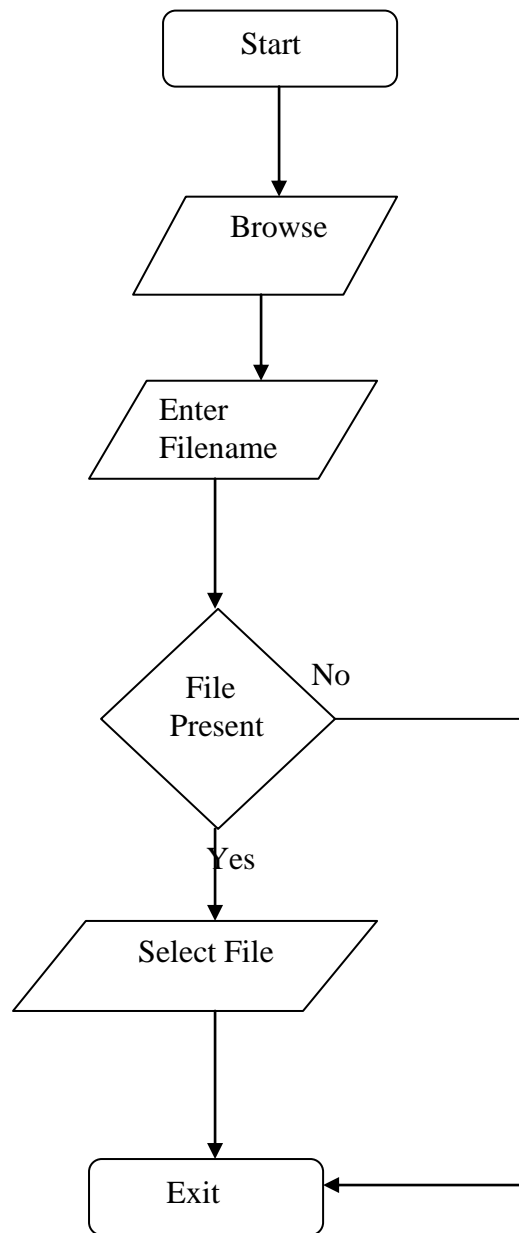
Flow chart of Set up Connection module



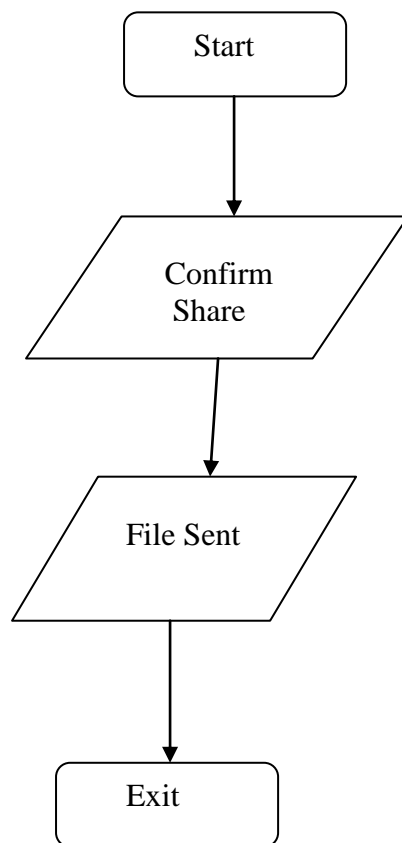
Flow Chart for Sign Up module



Flow Chart for Select File module

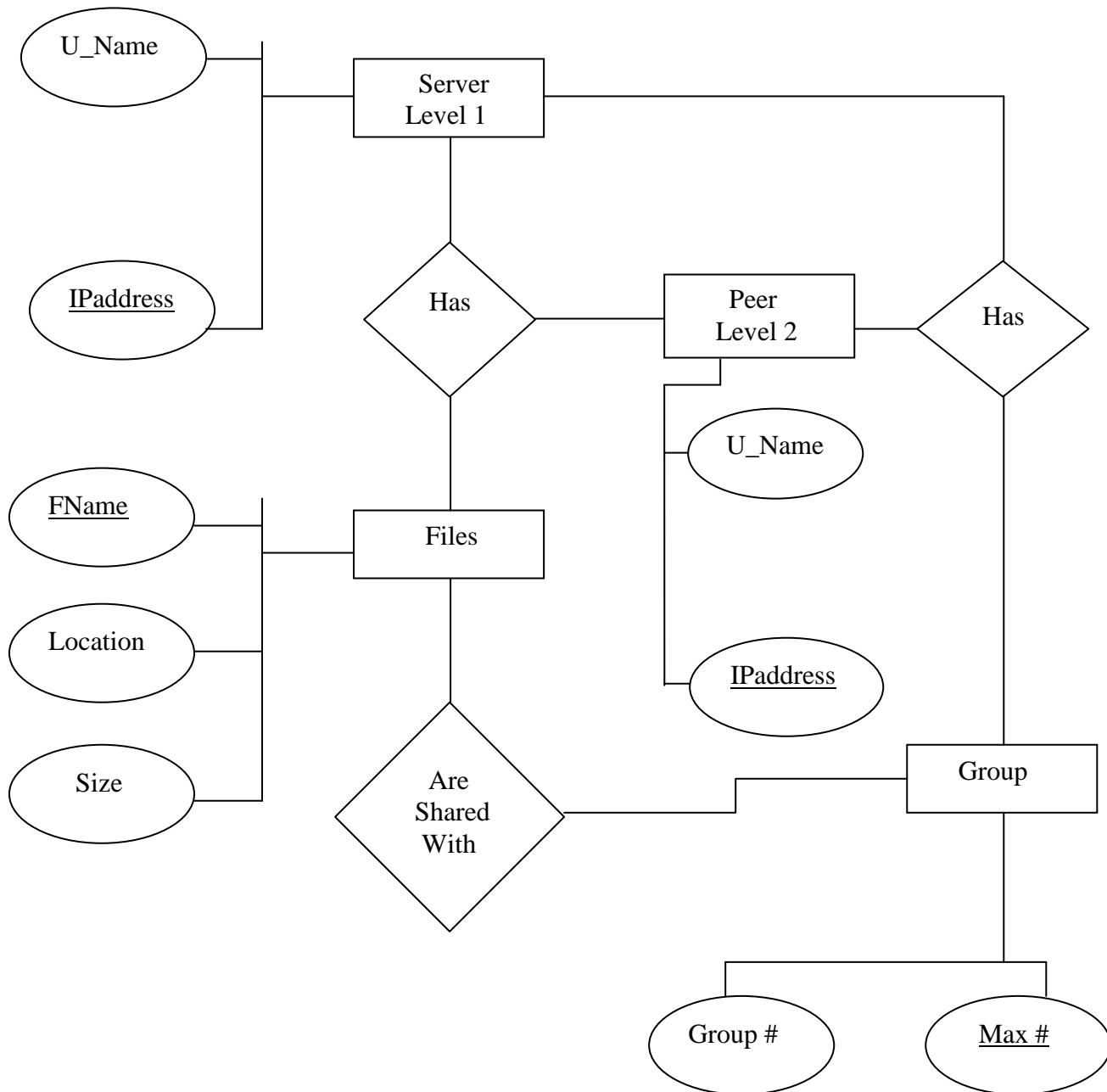


Flow Chart for Share module



4.5.5 E-R Diagram

The Entity Relationship Diagram depicts the relationship between all the real-world entities present in the application/software along with attributes of all individual entities.



4.5.6 Database Description

4.5.6.1 Table Name: Server Level 1

The User Level 1 table is used to store the information related to the level 1 users of the network hierarchy in the application. The information is matched against various conditions for a consistent database of all the users in network.

Tag Name	Synonyms	Description
User	User-Name	Stores the user name

4.5.6.2 Table Name: Peer Level 2

The User Level 2 table is used to store the information related to the level 2 or secondary users of the network hierarchy in the application. The information is matched against various conditions for a consistent database of all the users in network.

Tag Name	Synonyms	Description
User	User-Name	Stores the user name
Password	Secure password	Stores password.

5.1 User Interface Design

This section provides the user interface design for the application.

5.1.2 Form Name: Quick Share Peer

The user interface design for complete application is shown below. Since the main motive of building the application was faster file transfer using bare minimum human intervention, a single form for complete application has been designed.

Layout of Login Form

Quick Share Peer		-	■	X
Host Address:	<input type="text"/>	Host Port:	<input type="text"/>	<input type="button" value="Connect"/>
Username:	<input type="text"/>	Password:	<input type="text"/>	<input type="button" value="Login"/> <input type="button" value="Sign Up"/>
<div></div>				
Files to Share:	<input type="text"/>	<input type="button" value="..."/>	<input type="button" value="Send"/>	

5.1.3 Form Name: Set up Connection

The Set up Connection form is used to configure the entire connection settings related available network in the particular geographical area where application is being used. There is an option to save settings before the application restarts so that pre-configured feature of application is achieved in its architecture.

Layout of Set up Connection Form

Quick Share Server		-	□	X
Database File:	<input type="text"/>	<input type="button" value="Browse"/>	<input type="button" value="Connect"/>	
<hr/>				
<div></div>				

6.1 Module Description

6.1.1 Module Name: Login

The Login module is the basic module of application architecture. It performs validation for user. This is done in order to secure the connection between users. Malicious files can be transferred when application is running by middle person.

6.1.1.1 Class Name: Validate_User

This class contains all the necessary functions that are needed to validate a given input of Username and password by the user to start the application.

6.1.1.2 Class Dependencies

The class Validate_User dependent on Init_Connection class of Set up Connection module as it would be taking user to next step of using application i.e. setting up connection.

6.1.1.3 Class Functions

The functions used in this class are:

- Input_Login ()
- Connection()

6.1.1.3.1 Function Name: *void Input_Login*

This function takes the input from user regarding the Username and Password.

6.1.1.3.1.1 Declaration

void Input_Login (string UName, string Password)

6.1.1.3.1.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	UName	Username of the user using the application
String	Password	Password of the user using the application

6.1.1.3.1.3 Output Parameters

Variable Type	Variable Name	Variable Description
String	Status_User	Displays the Login status of the user.

6.1.1.3.1.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.1.3.1.5 Pseudo code

The pseudo code for the function is:

```
Step 1. Read Username
Step 2. Read password
Step 3. Check Username
Step 4. Check Password
Step 5. If Username and Password match;
    Then move to next form
    Else
        Request Correct Username and Password
```

6.1.1.3.2 Function Name: *ToSetConnection_Page* ()

This function takes user to next form of the application for creating or updating group for sharing the file.

6.1.1.3.2.1 Declaration

```
string ToSetConnection_Page(string Status)
```

6.1.1.3.2.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	Status_User	Status of the login for the application.

6.1.1.3.2.3 Output Parameters

Variable Type	Variable Name	Variable Description
Integer	Next_form	Displays the form for creating of updating user group by checking condition initialized earlier.

6.1.1.3.2.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.1.3.2.5 Pseudo code

The pseudo code for the function is:

```
Step 1. Read status of user login.
Step 2. Check status.
Step 3. If status is equal to old;
    Then move to Update user group form.
    Else
        Move to create user group form.
```

6.1.2 Module Name: Sign Up

The Sign Up module is used to create a set of users with a predefined size; together they form the end-users of the application. The file sharing is done between these user groups.

6.1.2.1 Class Name: Create_UserGroup

This class contains all the necessary functions that are needed to Create a new group of users with whom file is to be shared.

6.1.2.2 Class Dependencies

The class Create_UserGroup is dependent on Init_Connection class of Set up Connection module as it would be requiring user to set up connection before creating a user group.

6.1.2.3 Class Functions

The functions used in this class are:

- void Input_User ()
- ToSendFile_Page()

6.1.2.3.1 Function Name: void Input_User ()

This function takes the input from user regarding the new user group and users being added for the first time.

6.1.2.3.1.1 Declaration

void Input_User (string User_Name, string IP_address)

6.1.2.3.1.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	User_Name	Username of the user added to the new group.
String	IP_address	IP address of the user added to the new group.

6.1.2.3.1.3 Output Parameters

Variable Type	Variable Name	Variable Description
String	User_Name	Adds new username to the database.
String	IP_address	Adds IP address to the respective user.

6.1.2.3.1.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.2.3.1.5 Pseudo code

The pseudo code for the function is:

Step 1. Read User's name.
Step 2. Read IP address of added user's computer.
Step 3. Check unique identity of username.
Step 4. Check IP address consistency.
Step 5. If User's name and IP address are consistent;
Then move to next form
Else
Request Correct User's name and IP address.

6.1.2.3.2 Function Name: *ToSendFile_Page* ()

This function takes user to next form of the application for selecting file to be shared with group users created earlier.

6.1.2.3.2.1 Declaration

string ToSendFile_Page(string SendFile_address)

6.1.2.3.2.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	SendFile_address	Address of the file i.e. the file location in memory of computer.

6.1.2.3.2.3 Output Parameters

Variable Type	Variable Name	Variable Description
File	File_ToSend	Selects the file to be sent to group users created earlier.

6.1.2.3.2.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.2.3.2.5 Pseudo code

The pseudo code for the function is:

Step 1. Read file address.
Step 2. If address is correct;
Select File and;
Then move to Send File form.
Else
Request correct file address.

6.1.3 Module Name: Set up Connection

This module performs the functionality of setting up the connection using available connections.

6.1.3.1 Class Name: Connect_User

This class contains all the necessary functions that are needed to set up a new connection.

6.1.3.2 Class Dependencies

The class Connect_User is dependent on Validate_User class of Login module as users' needs to authenticate themselves before using application.

6.1.3.3 Class Functions

The functions used in this class are:

- void Check_Connection()
- void Connect_User()

6.1.3.3.1 Function Name: void Input_User ()

This function checks the available connections in the area where application is being used. The connections can be Internet connectivity like Wi-Fi, Ethernet based Broadband Connection etc.

6.1.3.3.1.1 Declaration

void Check_Connection (Boolean IP_Status)

6.1.3.3.1.2 Input Parameters

Variable Type	Variable Name	Variable Description
Boolean	IP_Status	Status of active Internet connection.

6.1.3.3.1.3 Output Parameters

Variable Type	Variable Name	Variable Description
String	Display_Connect	Displays available connection name.

6.1.3.3.1.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.3.3.1.5 Pseudo code

The pseudo code for the function is:

Step 1. Check open socket of user's computer.
Step 2. Connect to available connection using IP address.
Step 3. If User's name and IP address are consistent;
 Then move to next form
 Else
 Display Error message.

6.1.3.3.2 Function Name: *void Connect_User ()*

This function is used to connect the user to the Internet.

6.1.3.3.2.1 Declaration

`void Connect_User(string IP_Addr, Boolean IP_Status)`

6.1.3.3.2.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	IP_Addr	IP Address of the computer on which application is running.
Boolean	IP_Status	Status of Internet connection. True if connection is established successfully.

6.1.3.3.2.3 Output Parameters

Variable Type	Variable Name	Variable Description
String	Message	Displays message if connected successfully.

6.1.3.3.2.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.3.3.2.5 Pseudo code

The pseudo code for the function is:

Step 1. Check Internet status
Step 2. If status is True;
 Display message
 Else
 Display error.

6.1.4 Module Name: Select File

The Select File Module deals with handling the files in the memory of computer device on which the application is running.

6.1.4.1 Class Name: Select_File

This class contains those functions which fetches the file from its location or memory location inputted by the user.

6.1.4.2 Class Dependencies

The class Select_File dependent on Validate_User class of Login module as well as Update_UserGroup class of Update user module there should be a well-defined user group with whom selected file is being shared.

6.1.4.3 Class Functions

The functions used in this class are:

- Select_File ()

6.1.4.3.1 Function Name: *void Select_File*

This function selects the desired file from memory of the computer.

6.1.4.3.1.1 Declaration

void Select_File (string File_Name)

6.1.4.3.1.2 Input Parameters

Variable Type	Variable Name	Variable Description
String	File_Name	File name of the file to be shared.

6.1.4.3.1.3 Output Parameters

Variable Type	Variable Name	Variable Description
File	Selected_File	File is selected from the memory.

6.1.4.3.1.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.4.3.1.5 Pseudo code

The pseudo code for the function is:

Step 1. Read Filename
Step 2. Check File path
Step 3. If File name and Path are consistent;
Then move to next form
Else
Request Correct Filename and Path.

6.1.5 Module Name: Share

The Share module deals with sending of file selected in Select File module.

6.1.5.1 Class Name: Share_File

This class contains the functions which send the selected file from sender's socket to receiver's socket using Socket programming.

6.1.5.2 Class Dependencies

The class Share_File is dependent on Select_File class of Select File module since there should be valid file selected prior to sending it to the users defined in the group.

6.1.5.3 Class Functions

The functions used in this class are:

- ToSend_Page ()

6.1.5.3.1 Function Name: *ToSend_Page()*

This function activates the Send option on selection of a valid file from the memory. The users are directed to send form from where they send file to group by clicking on send button.

6.1.5.3.1.1 Declaration

void ToSend_Page (Boolean Send_Status)

6.1.5.3.1.2 Input Parameters

Variable Type	Variable Name	Variable Description
Boolean	Send_Status	Stores the truth value of the status of Send option of the application. If true, then the user can send file successfully.

6.1.5.3.1.3 Output Parameters

Variable Type	Variable Name	Variable Description
File	Selected_File	File selected from the memory is then sent using TCP/IP socket server to client's socket.

6.1.5.3.1.4 Return values

Variable Type	Variable Name	Variable Description
Void	N/A	N/A

6.1.5.3.1.5 Pseudo code

The pseudo code for the function is:

Step 1. Read status of Send option.
Step 2. If status variable is True;
 Then enable Send option.
 Else
 Display Error.

5. Test Planning

5.1 Introduction

Quick Share is a file sharing application based on friend to friend (a modified form of P2P) file transfer architecture. The application is intended to solve cumbersome problem of file sharing between two or more users involving use of flash disks, cabled devices etc.

The Quick Share file sharing application is based on two key functions. They are

- Simple, pre-configured User Interface
- Faster file transfer

This section discusses the test planning phase of the project implementation.

5.2 System Test

Here we have provided all the testing criteria and test specifications for the project.

5.2.1 System passing Criteria

The main passing criterion for the system is that the system should be able to transfer a given file from one peer to other peer without any errors or corruption in file. The criteria is said to be fulfilled when a file has been transferred in the given time limit also.

5.2.2 Functionalities Test

Requirement ID / Test ID	Module	Features
QS_TC1	Login	Here the validation of Username and Password given by the user will be done.
QS_TC2	Sign Up	Here a new user registers for participating in the network.
QS_TC3	Set Connection	This module is used to select available connections. The module would be tested for proper establishment of a WAN/LAN connection.
QS_TC4	Select File	In this module the user has to select a file. The file must be of a size less than specified limit.
QS_TC5	Share	In this module the user can send file. The file should be transferred successfully.

5.2.3 Functionalities to be not tested or verified

The alpha version of the application is exempted from following tests:

- Transfer of file of large sizes
- Dynamic update of IP addresses according to location.
- Compatibility of application with other platforms.

5.3 Test Setup

This section mentions about the hardware and software requirements for testing purposes.

5.3.1 Hardware

The Hardware requirements for the application are:

- User Interface: Keyboard, Mouse.
- Network component: Network Interface Card.
- Processor: Pentium IV or Higher.
- Hard Disk: At least 10 GB of free space.
- RAM: 1GB or more.

5.3.2 Software

The software requirements are:

- Operating System: Windows XP or Higher.
- Java Runtime Environment 1.6 or Higher.
- IDE/Platform: Eclipse Helios 4.2
- XML support.

5.3.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC1	28/01/2013	11/02/2013	Testing the Login Module.
QS_TC2	11/02/2013	25/02/2013	Testing the Sign Up module.
QS_TC3	25/02/2013	10/03/2013	Testing the Set Connection Module.
QS_TC4	10/03/2013	24/03/2013	Testing the Select File Module.
QS_TC5	24/03/2013	1/04/2013	Testing the Share File Module.

5.4 Test Case Specification

This section deals with detailed test case specification for the project testing phase.

5.4.1 Test Case: QS_TC1

5.4.1.1 Objectives

The test case QS_TC1 is designed to check the Login functionality of the application.

5.4.1.2 Module / Class /Function

Module: Login

Class: Validate_User

Functions:

Input_Login ()

ToSetConnection_Page ()

5.4.1.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC1	28/01/2013	11/02/2013	Login module would be bug and error free if application is started when provided with user credentials.

5.4.1.4 Expected Output or Results

The expected results from the Test Case are:

Test Case	Result
QS_TC1	User can run the application after a successful Login. If the credentials are incorrect, application cannot be used.

5.4.1.5 Test Passing Criteria

Test Case	Passing Criteria
QS_TC1	User is directed to next UI form only if the credentials inputted are correct as given in database earlier.

5.4.2 Test Case: QS_TC2

5.4.2.1 Objectives

The test case QS_TC2 is designed to check the Sign Up module functionality of the application. The Sign Up option helps the users to create a list of peers in a group when they are using the application for the first time.

5.4.2.2 Module / Class /Function

Module: Sign Up

Class: Sign_Up_User

Functions:

Input_User ()

ToSendFile_Page ()

5.4.2.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC2	11/02/2013	25/02/2013	Sign Up module would be bug and error free if peer is bale to register at any time. Database should be updated accordingly.

5.4.2.4 Expected Output or Results

The expected results from the Test Case are:

Test Case	Result
QS_TC2	The user network is created. The file to be sent can now be shared with users of the above network.

5.4.2.5 Test Passing Criteria

Test Case	Passing Criteria
QS_TC2	Number of peers added in the network should not exceed the limit as per level.

5.4.3 Test Case: QS_TC3

5.4.3.1 Objectives

The test case QS_TC3 is designed to check the Set Connection functionality of the application.

5.4.3.2 Module / Class /Function

Module: Set Connection

Class: Set_Connection_Server

Functions:

ToSendFile_Page ()

5.4.3.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC3	25/02/2013	10/03/2013	Set Connection module would be bug and error free if user is able to connect with server.

5.4.3.4 Expected Output or Results

The expected results from the Test Case are:

Test Case	Result
QS_TC3	The peers can connect to central server.

5.4.3.5 Test Passing Criteria

Test Case	Passing Criteria
QS_TC3	Correct IP address of the server should be used by the user along with the port number.

5.4.4 Test Case: QS_TC4

5.4.4.1 Objectives

The test case QS_TC4 is designed to check the Select file functionality of the application. This option helps the user to select the files that are to be shared among peers.

5.4.4.2 Module / Class /Function

Module: Select file

Class: Select_User

Functions:

Browse_File

5.4.4.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC4	10/03/2013	24/03/2013	Select file module would be bug and error free if user can select desired file using file browser.

5.4.4.4 Expected Output or Results

The expected results from the Test Case are:

Test Case	Result
QS_TC4	Available file are passed to Share module for sending to peers in subsequent steps.

5.4.4.5 Test Passing Criteria

Test Case	Passing Criteria
QS_TC4	The available files in the peer system should be displayed correctly.

5.4.5 Test Case: QS_TC5

5.4.5.1 Objectives

The test case QS_TC5 is designed to check the main functionality of the application i.e. to send the file from one peer to other peer.

5.4.5.2 Module / Class /Function

Module: Share

Class: Share_File

Functions:

ToSend_Page ()

5.4.5.3 Test Configuration

Test Case ID	Starting Date	Completion Date	Execution
QS_TC5	24/03/2013	1/04/2013	Share module would be bug and error free if the selected file is transferred to all the peers of group successfully.

5.4.5.4 Expected Output or Results

The expected results from the Test Case are:

Test Case	Result
QS_TC5	The file is transferred to other peers in the group created by the users.

5.4.5.5 Test Passing Criteria

Test Case	Passing Criteria
QS_TC5	The file should be sent successfully and without any errors.

6. Work Distribution

Week	Module Name	Task Involved	Start Date	End Date	Responsible Person
Week 1	Login module And Sign Up module	<ul style="list-style-type: none"> Implementing a Login module And Sign Up module Creating a database table for storing user credentials. Front End Design. Database Design. Unit Testing. 	28/1/2013 29/1/2013 31/1/2013 2/2/2013 4/2/2013	29/1/2013 31/1/2013 2/1/2013 3/2/2013 4/2/2013	Kunal Bohra
Week 2	Sign Up module	<ul style="list-style-type: none"> Implementing a Sign form. Study of Network connection system call functions for windows platform. 	4/2/2013 6/2/2013	6/2/2013 11/2/2013	Kunal Bohra
Week 3	Set Up Connection	<ul style="list-style-type: none"> Front End Design. Database Design. Unit Testing. 	11/2/2013 13/2/2013 17/2/2013	13/2/2013 17/2/2013 18/2/2013	Kunal Bohra

Week 4	Integration of Modules	<ul style="list-style-type: none"> Integrating Login, Sign Up and Set up Connection modules as per flow diagram of system. Integration of User interfaces of modules implemented. Integration testing of completed modules. 	18/2/2013 21/2/2013 23/2/2013	21/2/2013 23/2/2013 25/2/2013	Kunal Bohra
Week 5	Files to Share	<ul style="list-style-type: none"> Implementing Files to Share form and database design. Study of peer to peer implementation in java. 	25/2/2013 28/2/2013	28/2/2013 3/3/2013	Kunal Bohra
Week 6	Files to Share	<ul style="list-style-type: none"> Front End Design. Integration of File to Share Module Unit testing. 	3/3/2013 5/3/2013 8/3/2013	5/3/2013 8/3/2013 10/3/2013	Kunal Bohra
Week 7	Integration of all modules along with Share module implementation	<ul style="list-style-type: none"> Combining Share module with system 	10/3/2013	12/3/2013	Kunal Bohra

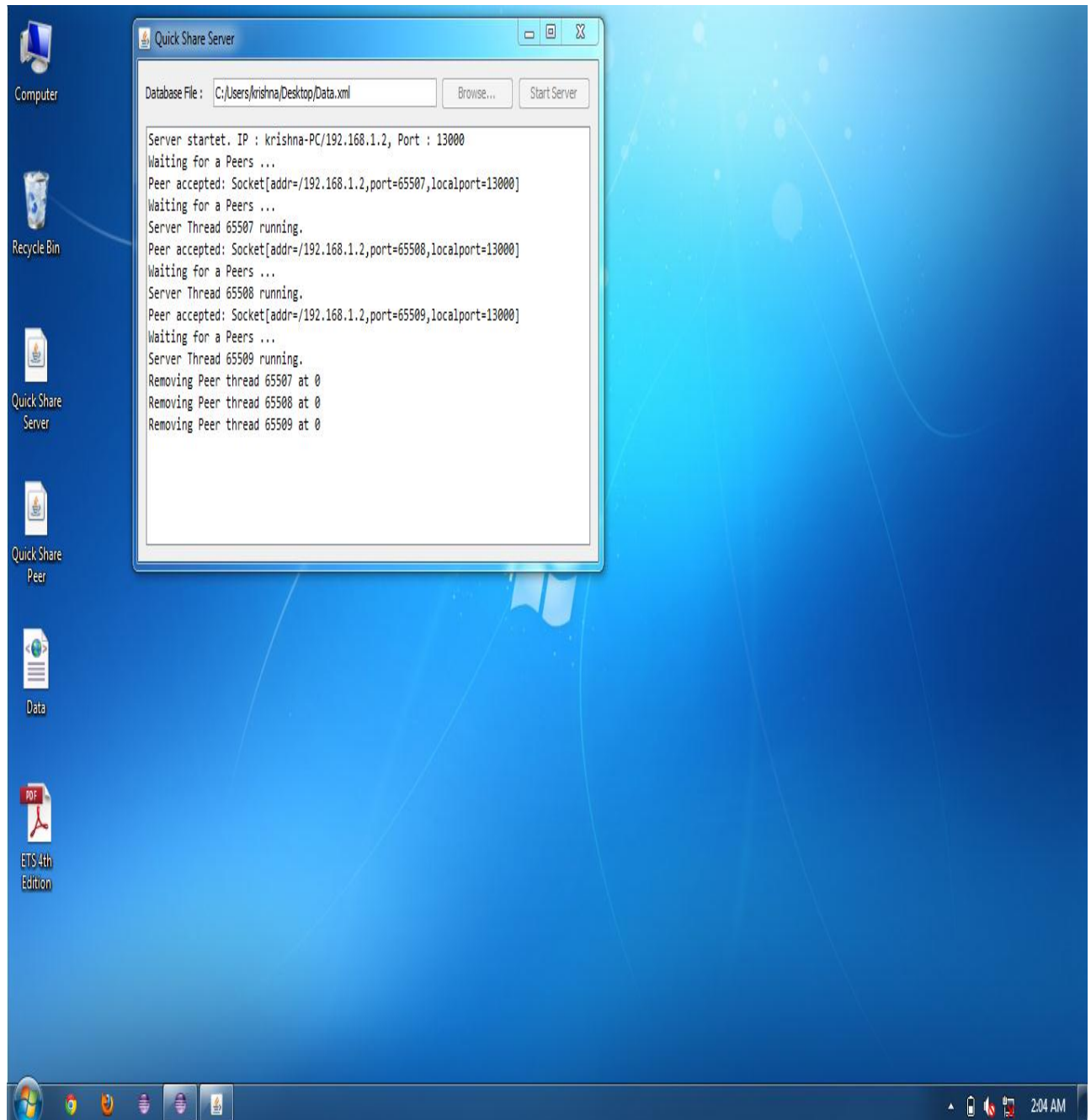
		<p>according to design plan.</p> <ul style="list-style-type: none"> • Combining user interface of all modules. • Integration testing of modules implemented. 	<p>12/3/2013</p> <p>16/3/2013</p>	<p>16/3/2013</p> <p>17/3/2013</p>	
Week 8	Integration Testing and System Testing.	<ul style="list-style-type: none"> • Integration testing of system. • System testing. 	<p>26/3/2013</p> <p>30/3/2013</p>	<p>29/3/2013</p> <p>2/4/2013</p>	Kunal Bohra

7. Results

Working Snapshots

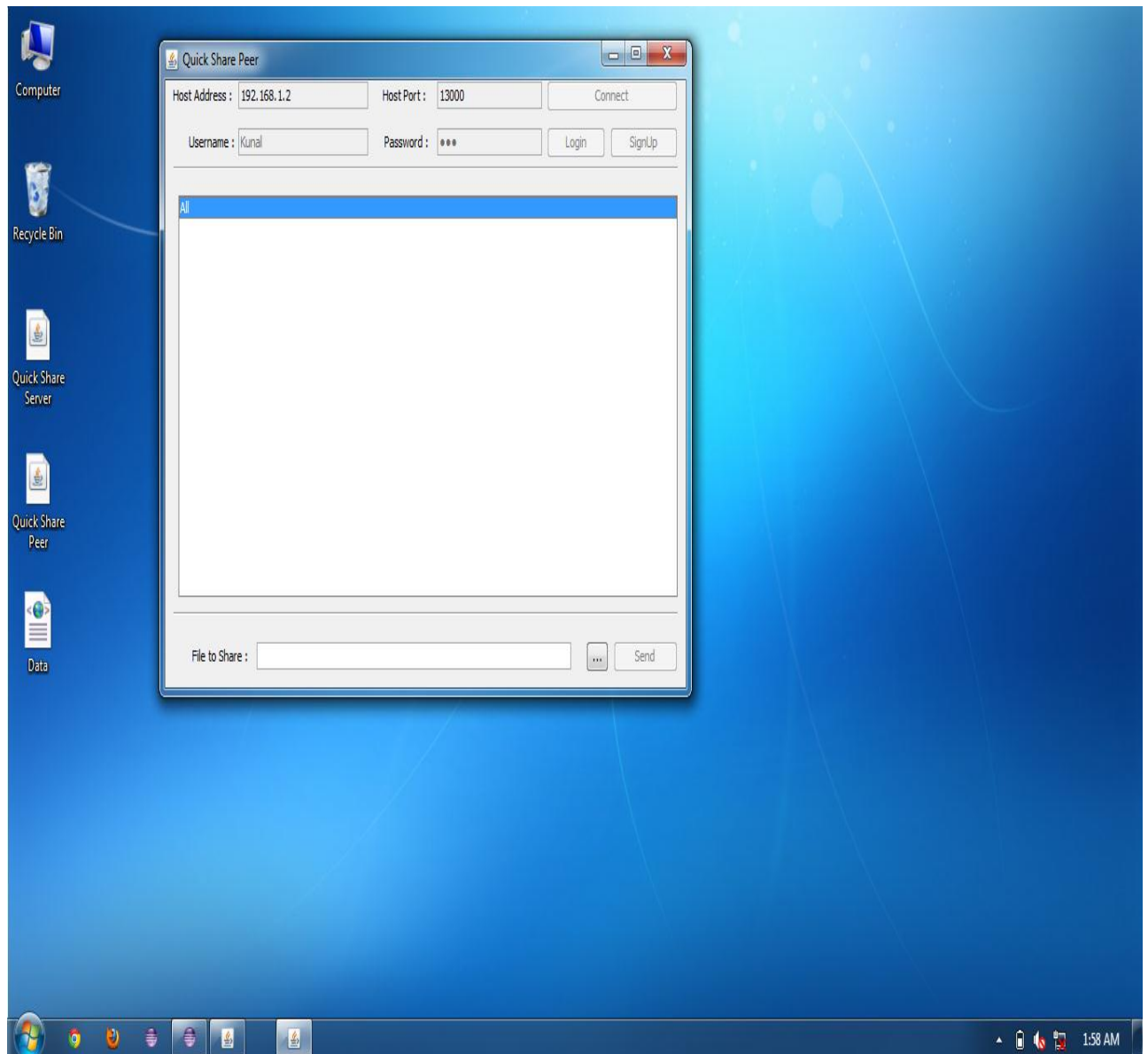
- **Quick Share Server Window**

This snapshot shows the main running Quick Share server and its starting point when supplied with database XML file.



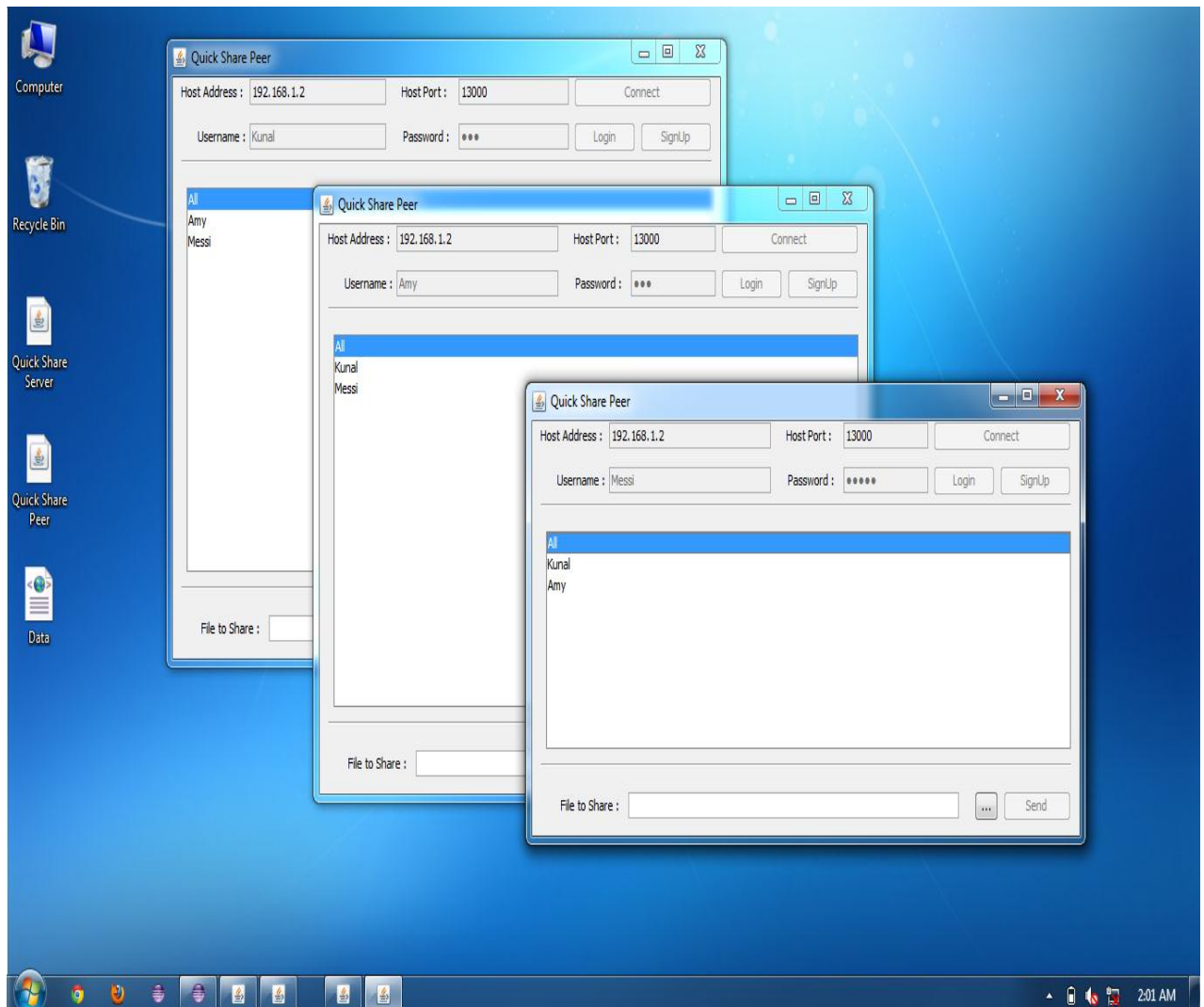
- **Quick Share Peer Window**

Here the running Quick Share Peer application window is shown.



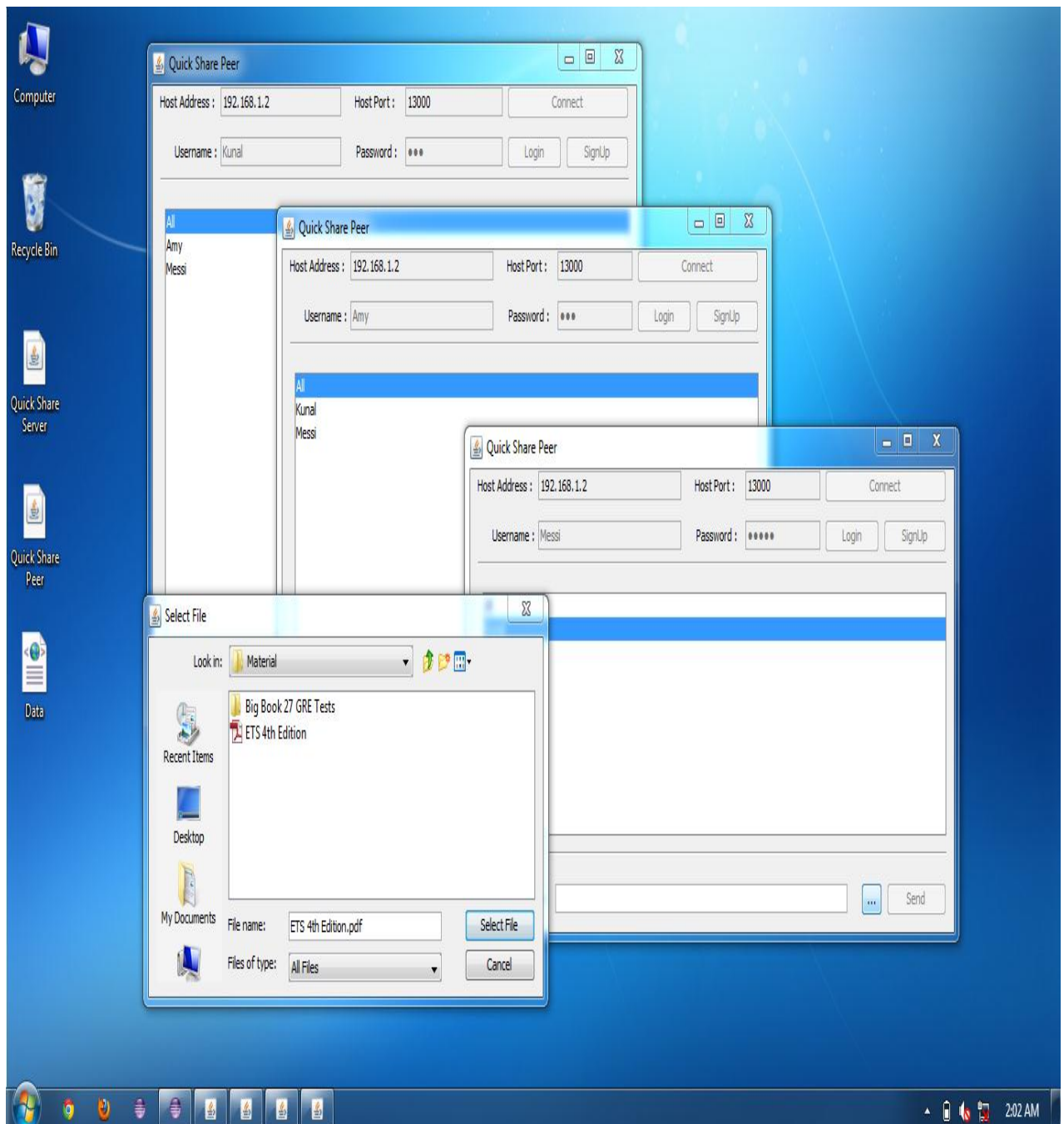
- **Quick Share Peer**

In this snapshot, a peer login or sign up in the network after connecting to the server and see the list of available users in the network.



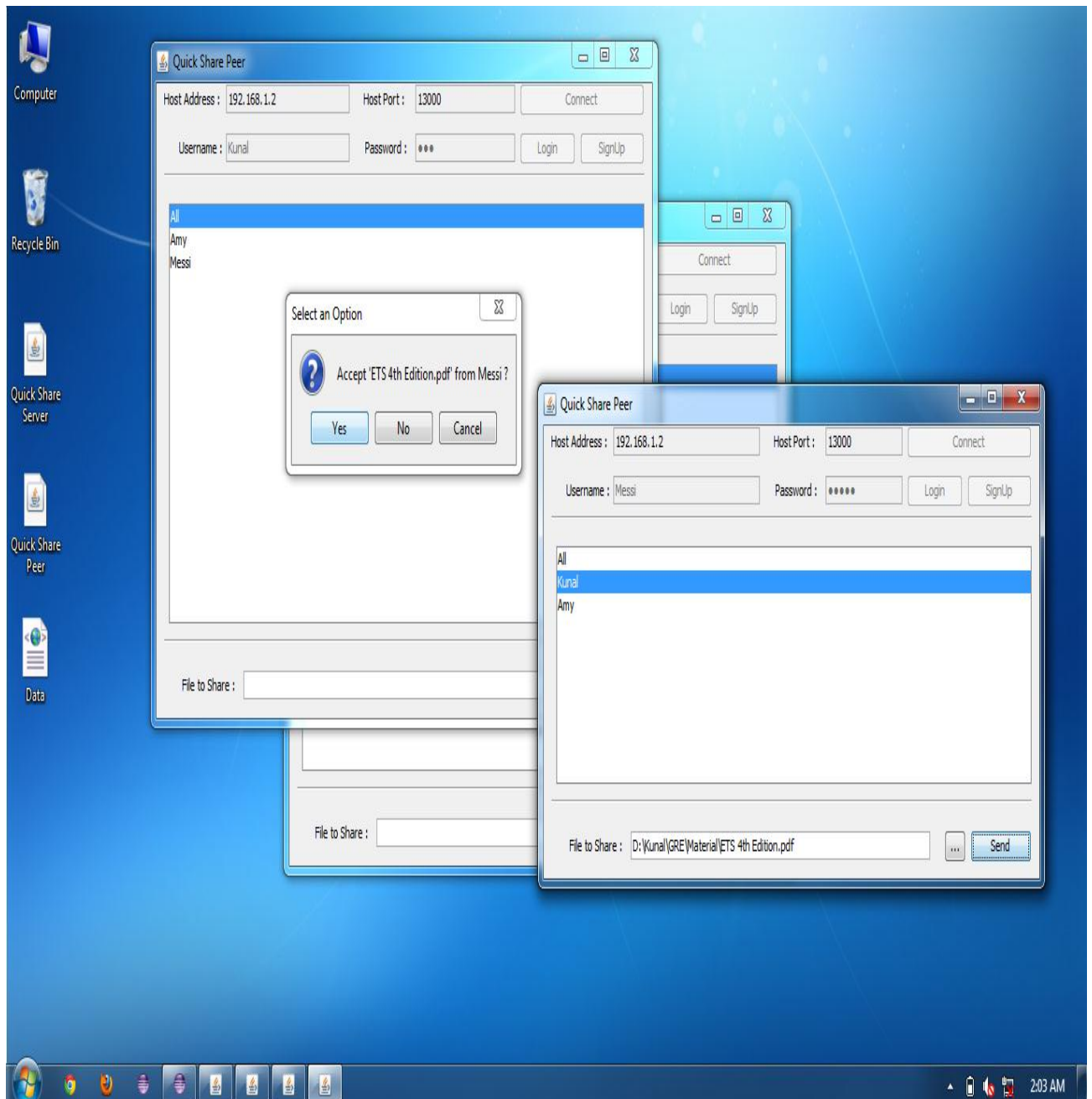
- **Quick Share Peer Select File**

This snapshot shows how a peer can select file from his system using file explorer.



- **Quick Share Peer File Sharing with other peers**

This window shows how finally a peer can act as client as well as server in order to send or receive file.



Test Status Report

S.No.	Module Name	Status (Pass/Fail)	Error Status (Fixed/Open)	Remarks
1	Login	Pass	Fixed	Peers can login in the application any time using credentials satisfying test pass criteria.
2	Sign Up	Pass	Fixed	A new Peer in network can use application by signing up anytime.
3	Set Connection	Fail	Open	Connection between Server and Peers cannot be established with an exception of WAN networks.
4	Select File	Pass	Fixed	Peers can select files using file explorer.
5	Share	Pass	Fixed	Application successfully sends files between peers. Any peer is client and server same time satisfying major criteria for a P2P network. Exception with file sending to multiple clients at a time.

8. Conclusion & Future Work

The application has been implemented successfully in accordance with timeline. There are certain issues that need to be worked upon for passing the test criterion such as connection with a remote server on a WAN network. The application has been developed up to its potential and aims at covering a medium sized geographical area such as classrooms, labs etc. where certain number of users can form a Peer to Peer network and can share file between them in a quick and easy way.

The future work on the project is aimed at:

- There is wide scope enhancing the Quick Share application by porting it to mobile devices which holds a great part of our daily lives today.
- Porting the application on remote networks and using the port forwarding in routers
- Sending multiple files or directories to the single peer or multiple peers as per user's choice.
- Enabling application to send files of large size.

Glossary

Abbreviation	Description
RFC	A Request for Comments (RFC) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.
DFD	Data Flow Diagrams are used to depict the flow of data between various functional modules of the software being developed.
JRE	The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language.
P2P	The main protocol in Internetworking for transferring files in a P2P network.
FTP	A widely used protocol in Internetworking for controlling the transfer of files between two or nodes in a network.
IDE	Interactive Development Environment

References

Web:

- [1] UML Diagrammer- www.pacestar.com/uml/index.html
- [2] Makin Use Case Diagrams- www.andrew.cmu.edu/course/90-754/umlucdfaq.html
- [3] Data Flow Diagram: <http://www.smartdraw.com/resources/tutorials/data-flow-diagrams>
- [4] E-R Diagram: <http://www.smartdraw.com/resources/tutorials/entity-relationship-diagrams>
- [5] Flow Chart: <http://www.breezetreel.com/articles/what-is-a-flow-chart.htm>
- [6] Module Description:
http://wiki.answers.com/Q/What_is_a_module_in_software_engineering
- [7] Function Signature:
<http://www.cs.unm.edu/~storm/C++/ProgrammingTerms/FunctionSignatures.html>

Text:

- [1] Software Engineering: A Practitioner's Approach, 7/e
Author: Roger S Pressman, R. S. Pressman & Associates, Inc.
- [2] Java: The Complete Reference
Author: Herbert Schildt, McGraw Hill Publication

Literature References:

- [1] Miguel Castro, Peter Druschel, Y. Charlie Hu, Antony Rowstron. "Topology aware routing in structured Peer-to-Peer overlay networks"
- [2] Robbert van Renesse, Ken Birman, Adrian Bozdog, Dan Dumitriu, Manpreet Singh, Werner Vogels. "Heterogeneity-Aware Peer-to-Peer Multicast"
- [3] Michael Kleis, Eng Keong Lua, Xiaoming Zhou. "Hierarchical Peer-to-Peer Networks using Lightweight Super Peer Topologies"
- [4] Zhonghong Ou. Structured Peer-To-Peer Networks: "Hierarchical Architecture and Performance Evaluation"