Software Design Document

(SDD)

For

Quick Share

A Faster File sharing application

Version 1.1 29/08/2012

Department of Computer Science Engineering
Jodhpur Institute of Engineering and Technology
Session 2012-13

APPROVAL and Compulsory Review LIST		
Role	**Name**	**Date**
Lab Faculty/ Incharge	Mr. Bharat Singh	29/08/2012
Guide	Mr. Ashutosh Vyas	27/08/2012

Version Tracking

Date	Version	Comments	Author
27/08/2012	1.0	Initial version of SDD	Kunal Bohra
29/08/2012	1.1	Changes in section 2.0 and 4.1	Kunal Bohra

Table of Contents

1.0 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 Windows ™ 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.

2.0 Functional Description

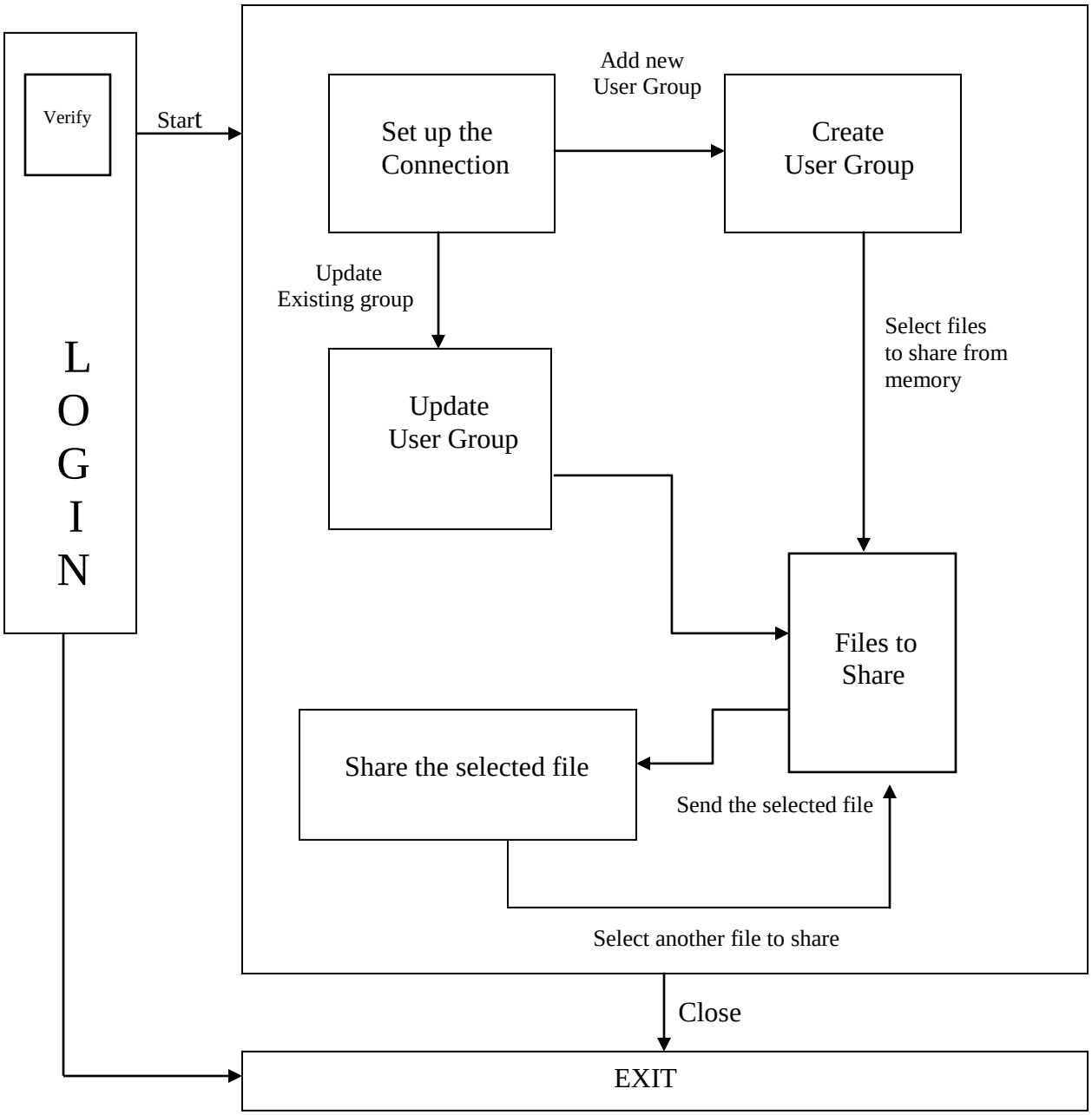
Functional overview

This section describes the basic functionality of the sofware. 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.
- Use of faster Internet bandwidth as available in 3G Networks or dedicated leased lines over a Wi-fi(802.11 a/b/g/n) is done in Quick Share application to provide a faster file access.

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 geagraphical area to connect to other users of application and share files. The connection is TCP/IP based Internet connection such as Broadband network.

Create a group of users

This group is used to create a group of users with whom files are to be shared. Creating a group 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.

Update existing group

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

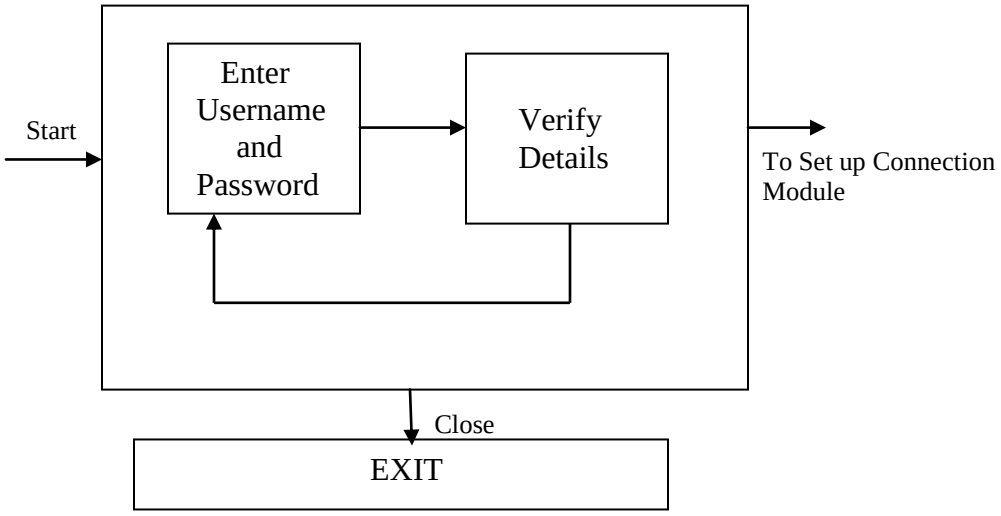
The application can be closed at any instant by the user.

3.0 Functional Partitions

3.1 Module Name: Login

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

3.1.1 Functional Block Diagram



3.1.2 Module Dependencies

3.1.2.1 *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.

3.1.2.2 *Output Dependencies*

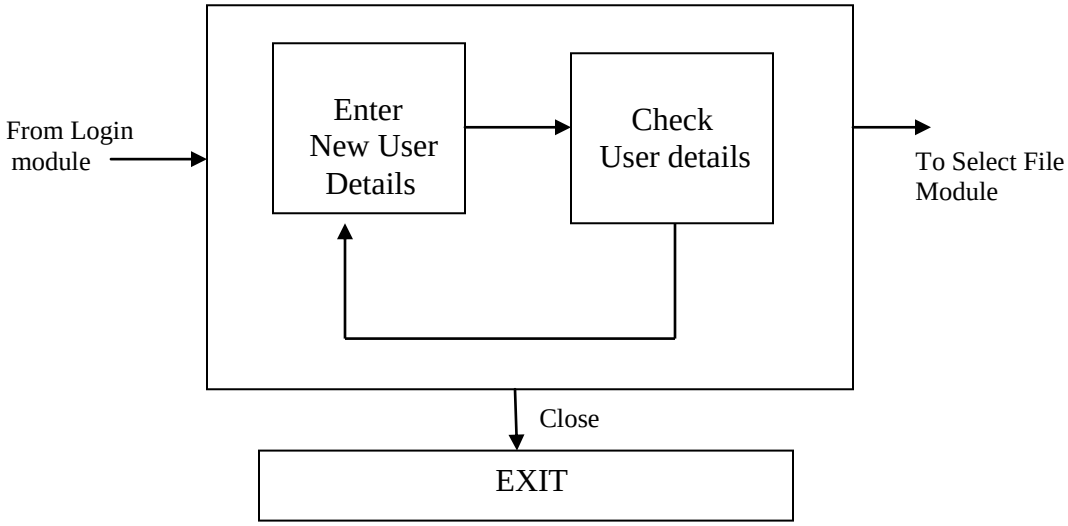
The output dependencies of Login module are described below:

Module	Data	Description
Set up Connection	Username	Correct username should be entered to proceed to next module.
Set up connection	Password	Correct username should be entered along with username to start application.

3.2 Module Name: Create Group

The Create Group 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.

3.2.1 Functional Block Diagram



3.2.2 Module Dependencies

3.2.2.1 *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 existing in the network.

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.

3.2.2.2 *Output Dependencies*

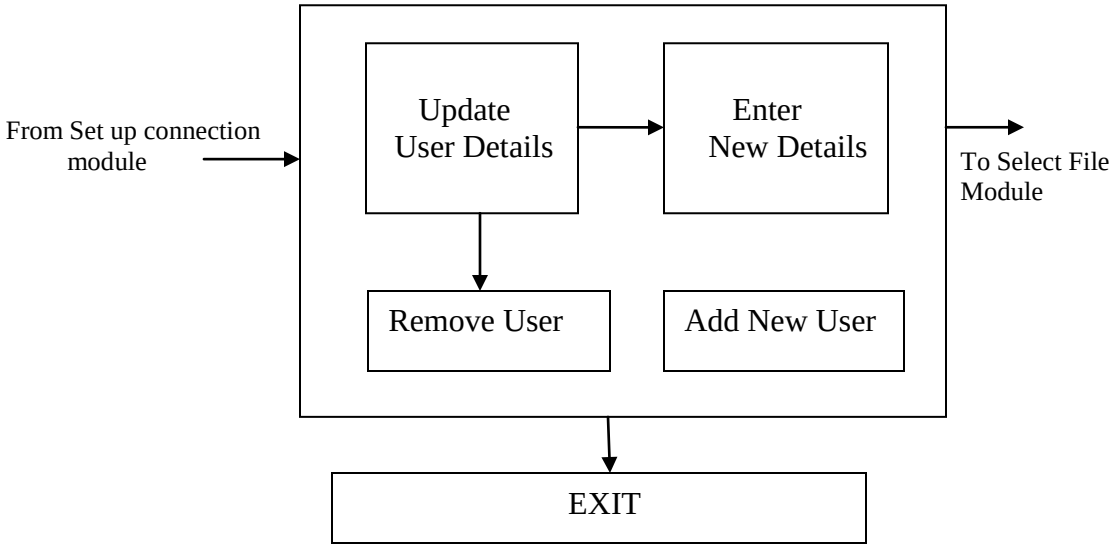
The output dependencies of Create User Group module are described below:

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.

3.3 Module Name: Update User Group

This module is incorporated in the architecture to provide extra flexibility to the user while using the application with users of different networks and geographical location. The update module is necessary so that user can modify an existing group in the application by adding new user or removing an existing user. The basic details can also be modified as per needs of user.

3.3.1 Functional Block Diagram



3.3.2 Module Dependencies

3.3.2.1 *Input Dependencies*

The Update User Group module has following Input dependencies:

User details: In order to update details of any user or remove existing user, the user must already exist in the group.

New user: Any new user can be added in the group provided that the user has valid IP address according to network as well as the maximum number of users defined for the group doesn't get exceeded while adding a new user.

3.3.2.2 *Output Dependencies*

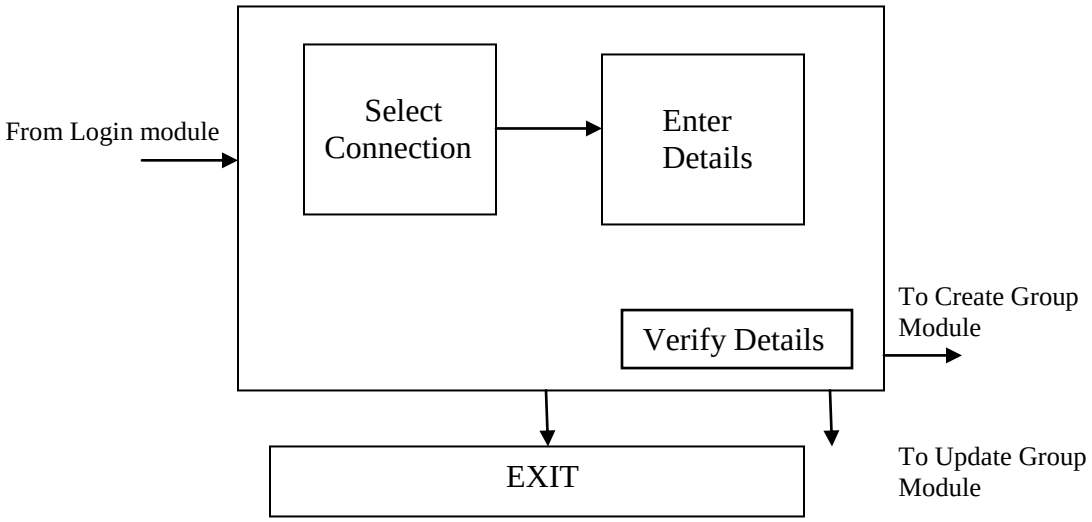
The output dependencies of Update User group module are described below:

Module	Data	Description
Select File	File name	Correct file name should be given according to database.
Select File	Updated details	The updated file details should be consistent i.e. multiple copies should not exist.

3.4 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.

3.4.1 Functional Block Diagram



3.4.2 Module Dependencies

3.4.2.1 *Input Dependencies*

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

Connection: The available connection in the geographical area is to be selected.

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

3.4.2.2 *Output Dependencies*

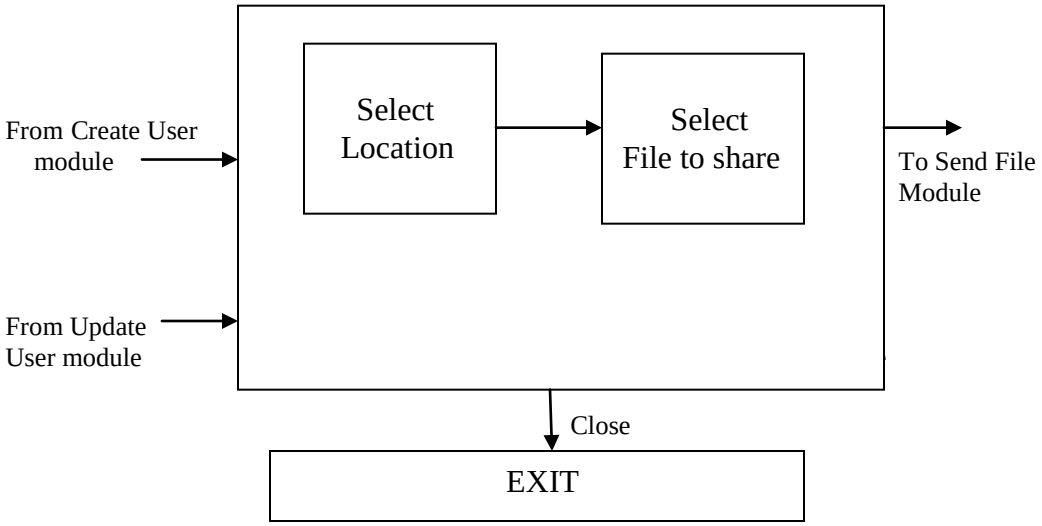
The output dependencies of Set up Connection module are described below:

Module	Data	Description
Create User Group	User details	The user details primarily IP address should be consistent.
Update User Group	Updated details	Details like password for protected connection should be entered correctly.

3.5 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.

3.5.1 Functional Block Diagram



3.5.2 Module Dependencies

3.5.2.1 *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.

3.5.2.2 *Output Dependencies*

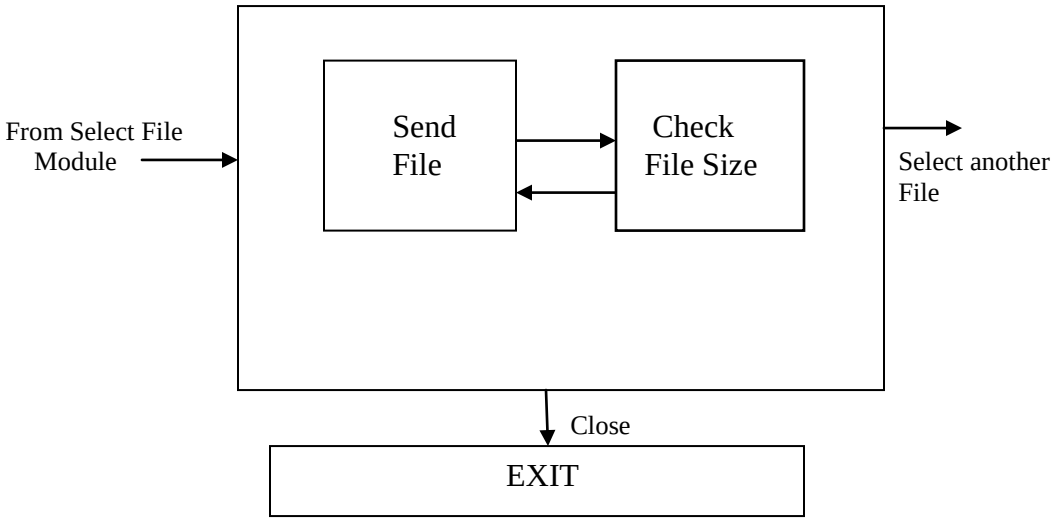
The output dependencies of Select File module are described below:

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.

3.6 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.

3.6.1 Functional Block Diagram



3.6.2 Module Dependencies

3.6.2.1 *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.

3.6.2.2 *Output Dependencies*

The output dependencies of Share module are described below:

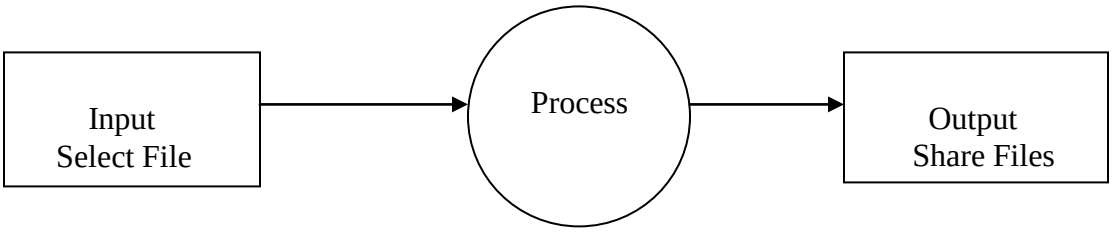
Module	Data	Description
Select File	File	A file with size less than or equal to prescribed size should be there.

4.0 Data Description

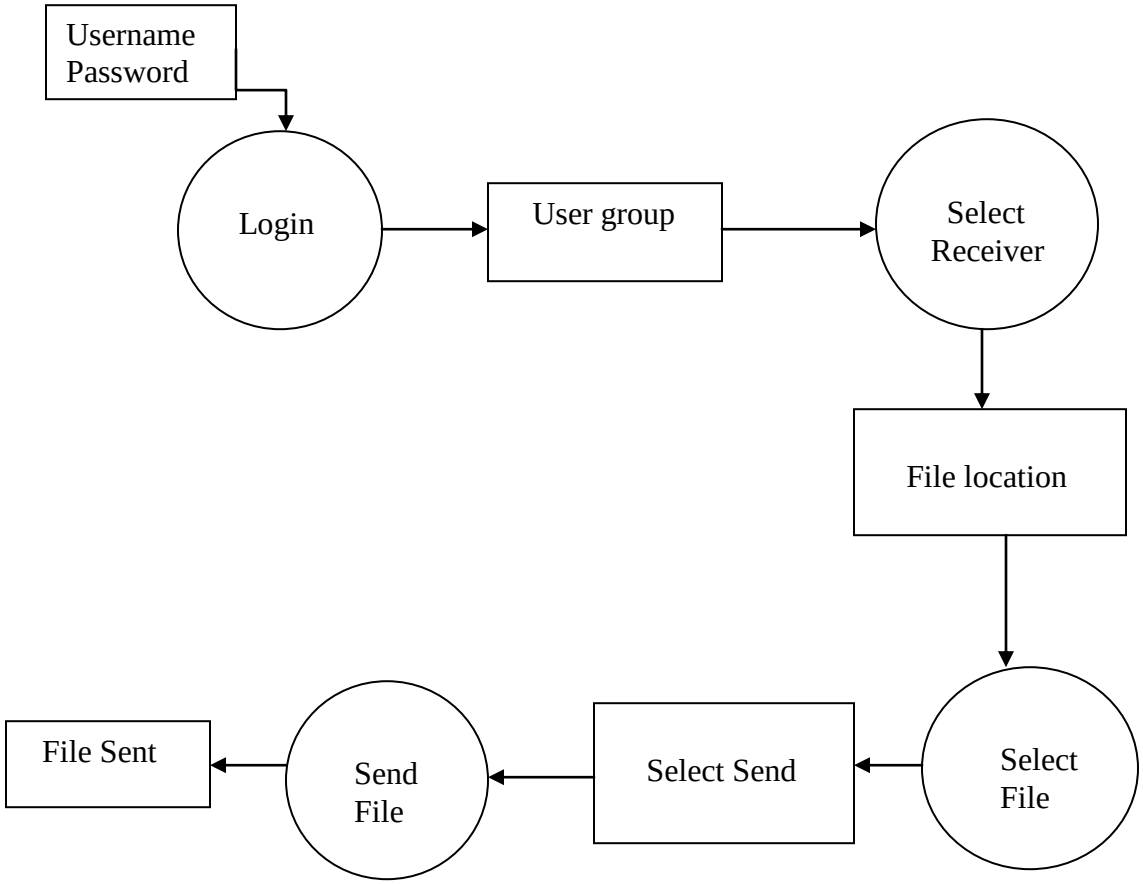
4.1 Data Flow Diagram

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

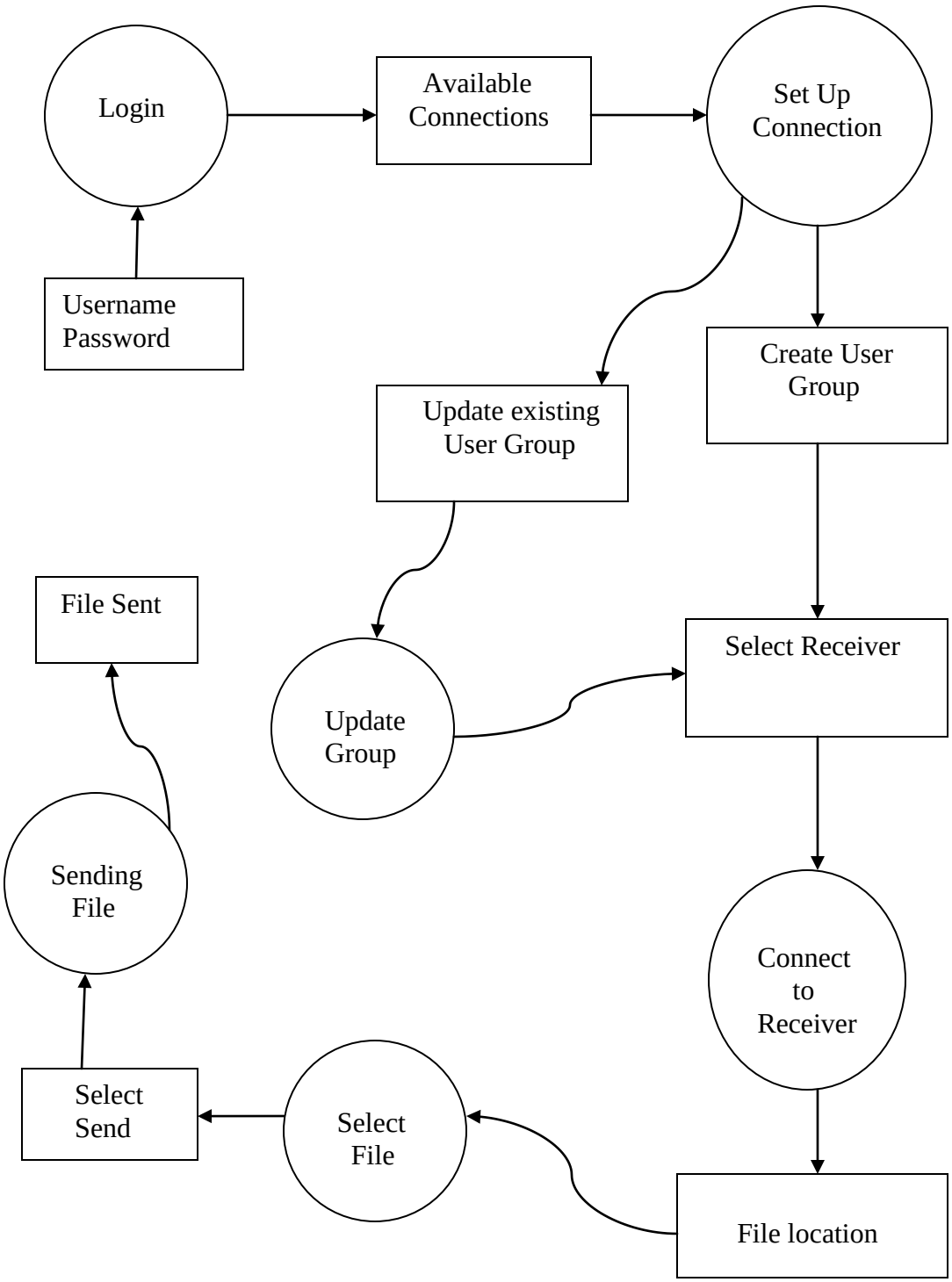
Context Level 0 DFD



Level 0 DFD



Level 1 DFD


Login
Available Connections
Set Up Connection
Username Password
Create User Group
Update existing User Group
File Sent
Select Receiver
Update Group
Sending File
Connect to Receiver
Select Send
Select File
File location

4.2 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
- Vector

4.3 Constant Definition

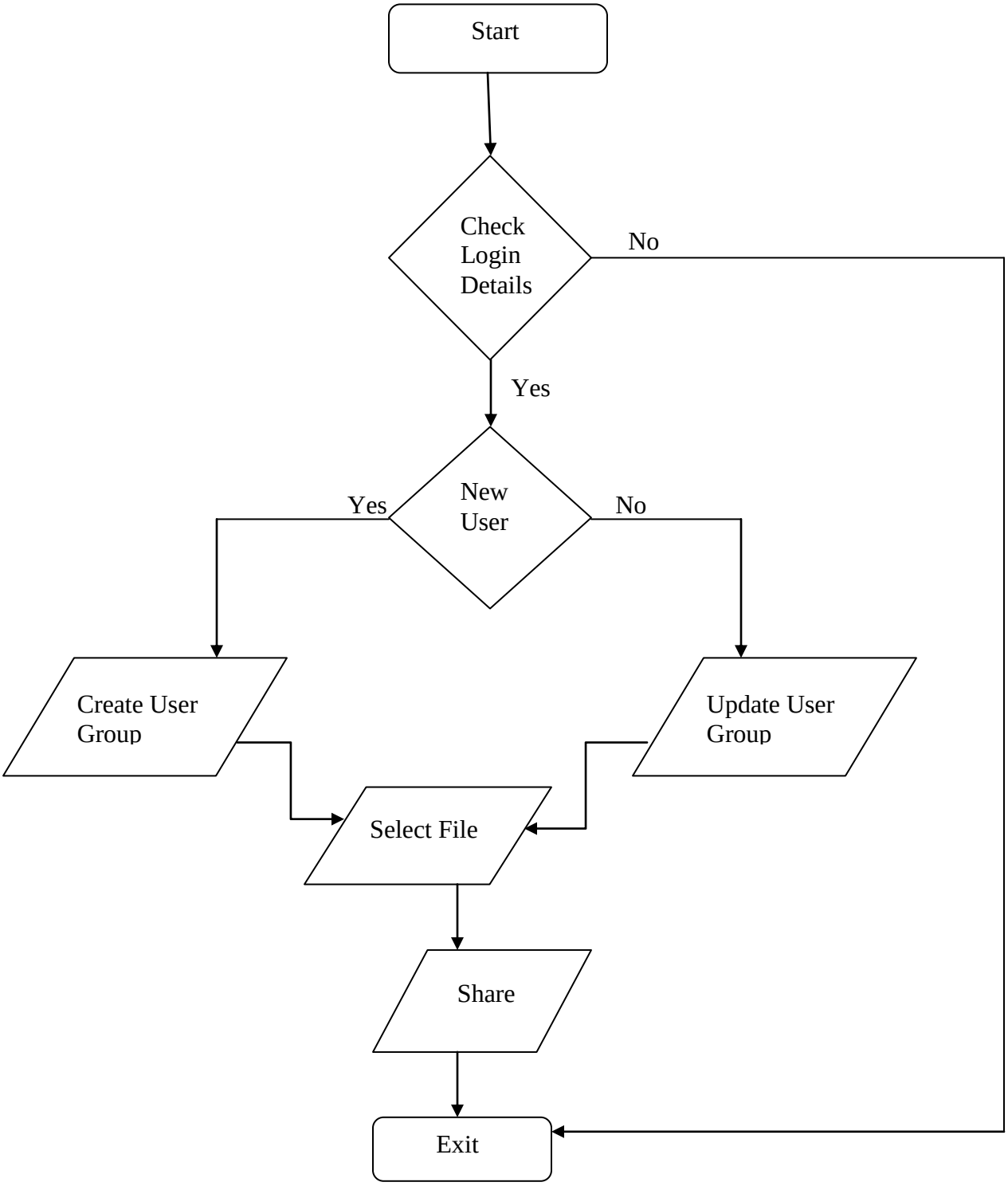
Certain constants are to be defined/declared in code of the application according to need of architecture. They are

- 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 share.
- User_Max: Maximum number of users in a group using the application.

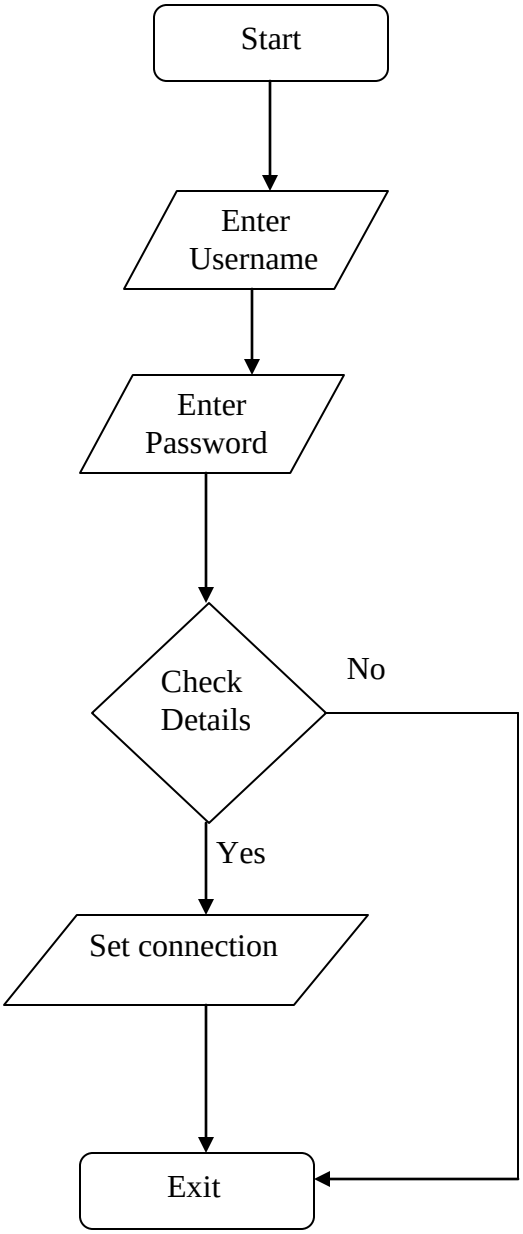
4.4 Flow Charts

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

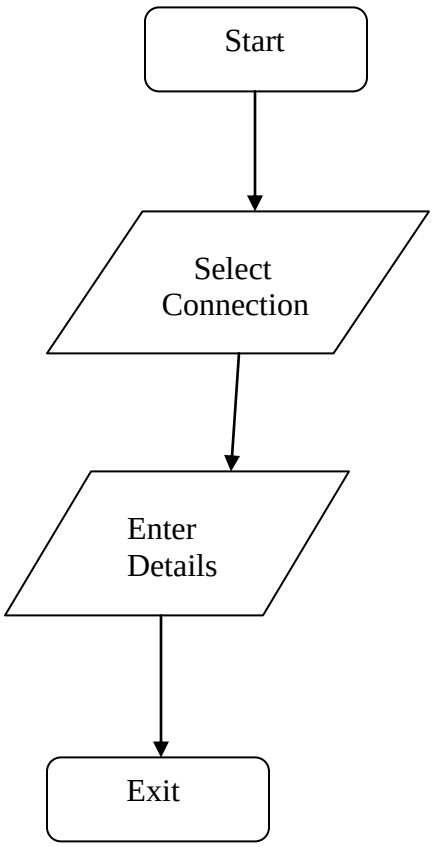
Flow Chart for complete application



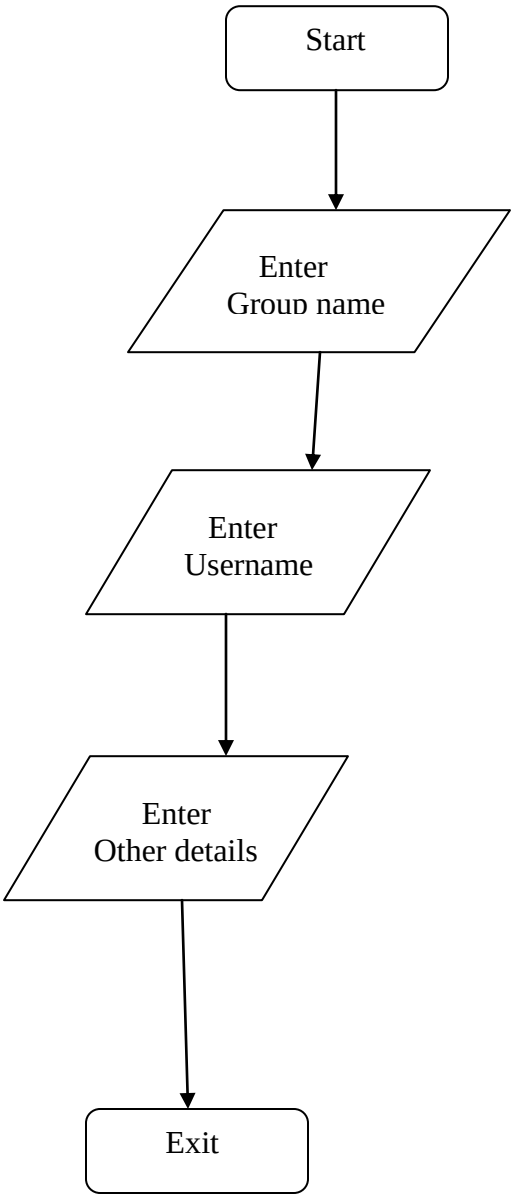
Flow Chart for Login module



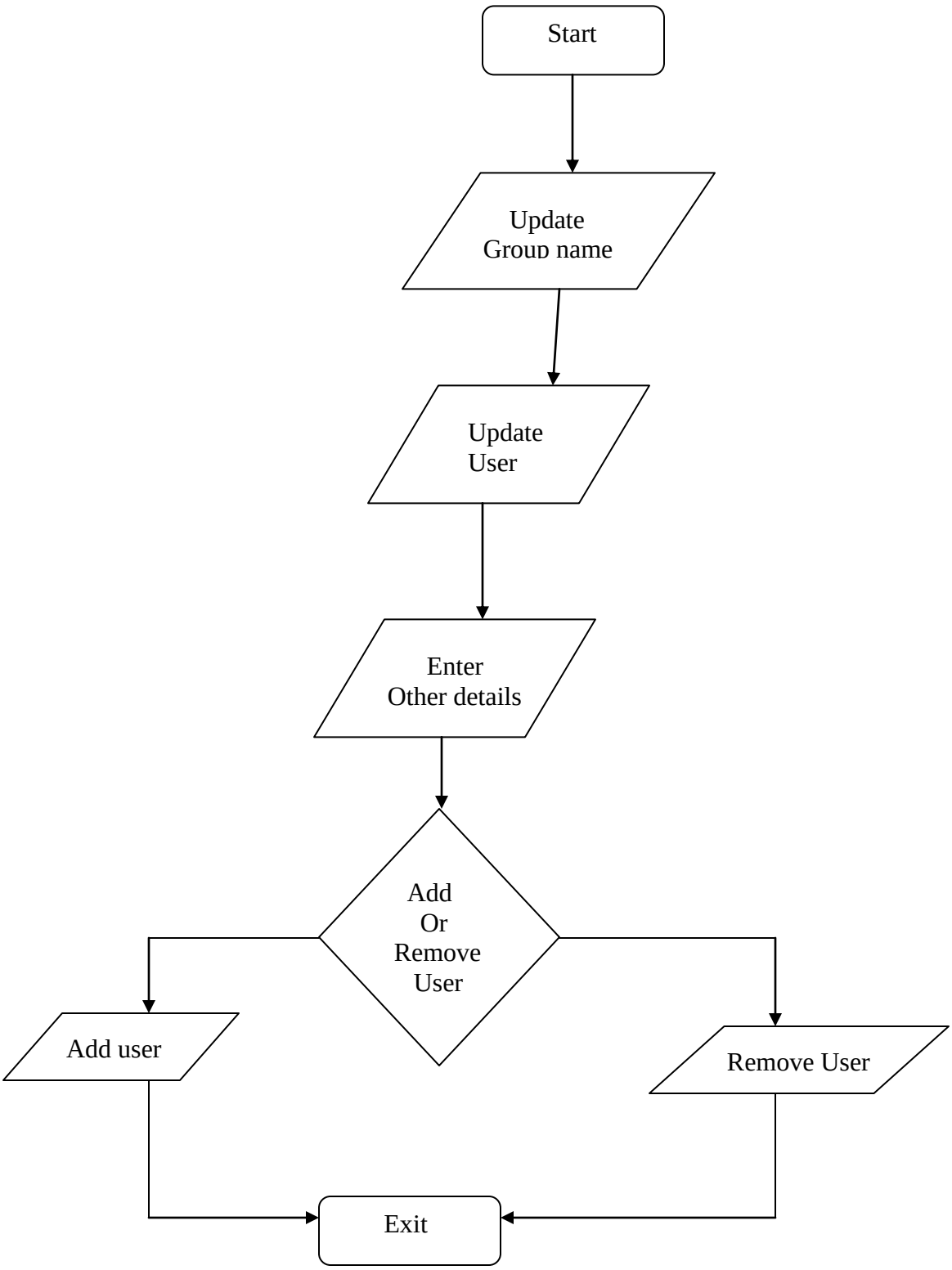
Flow chart of Set up Connection module



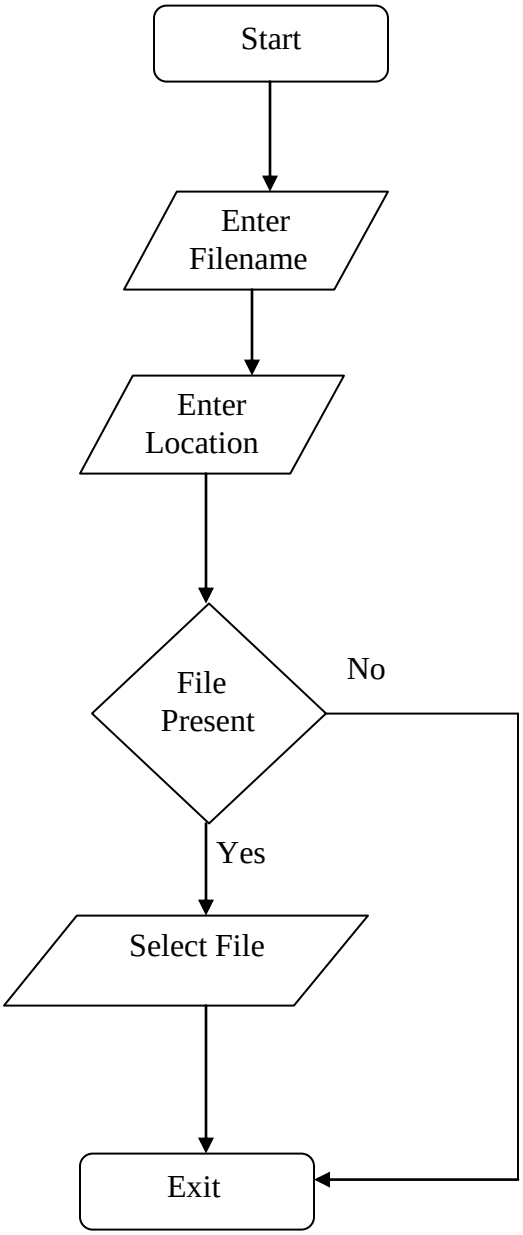
Flow Chart for Create User Group module



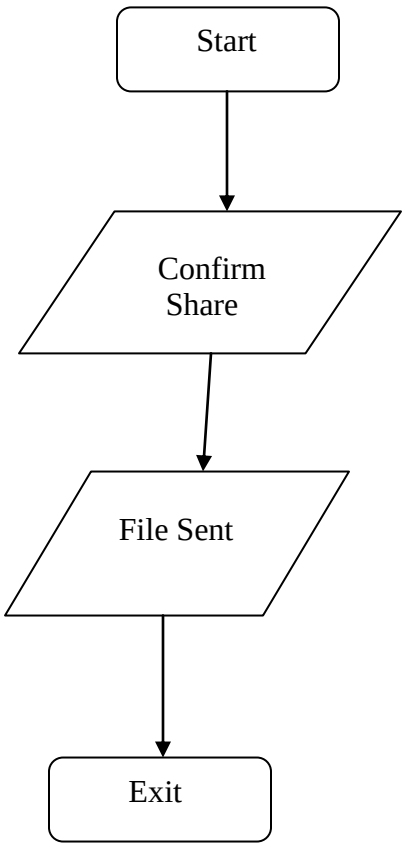
Flow Chart for Upgrade Group module



Flow Chart for Select File module

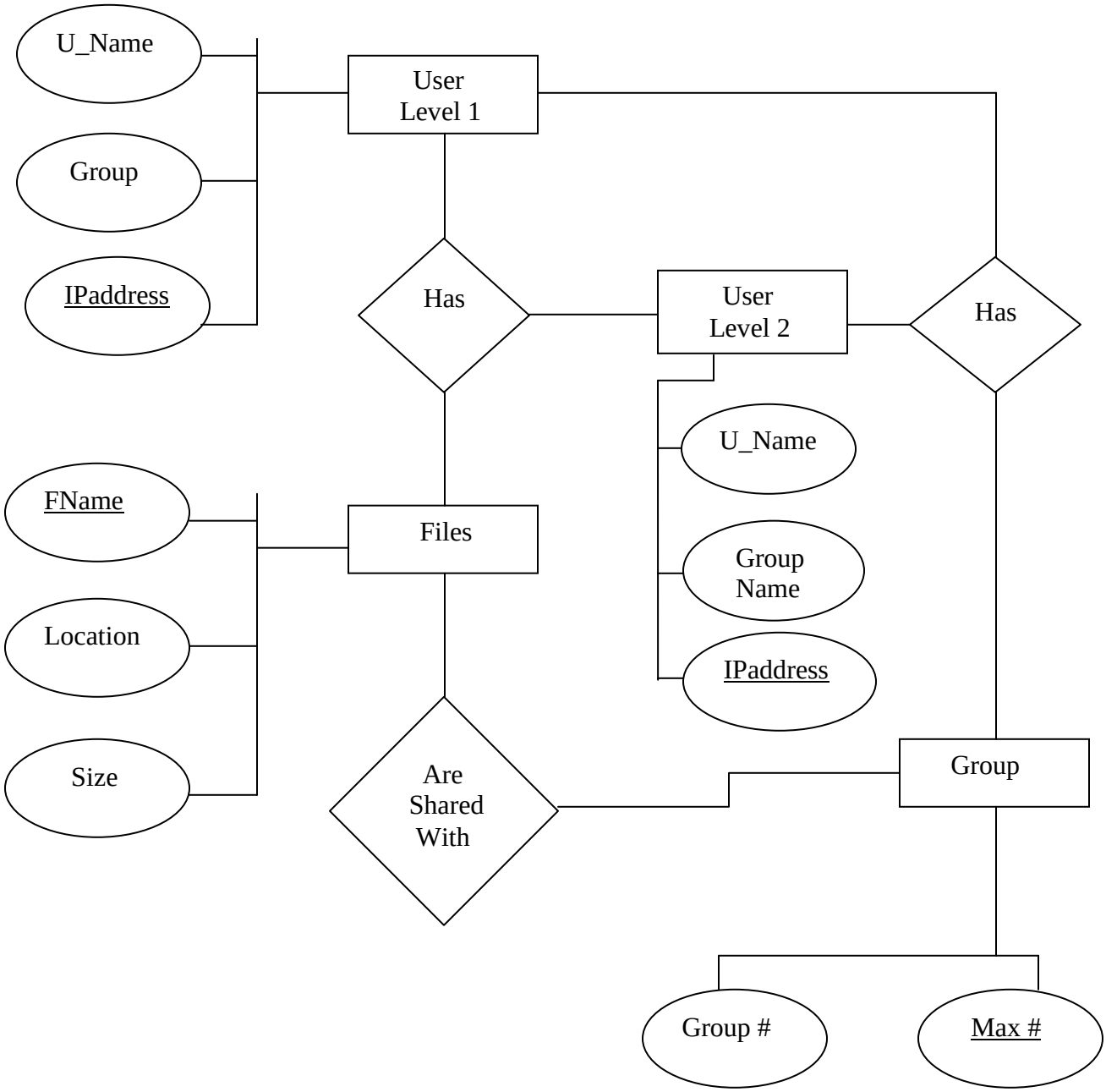


Flow Chart for Share module

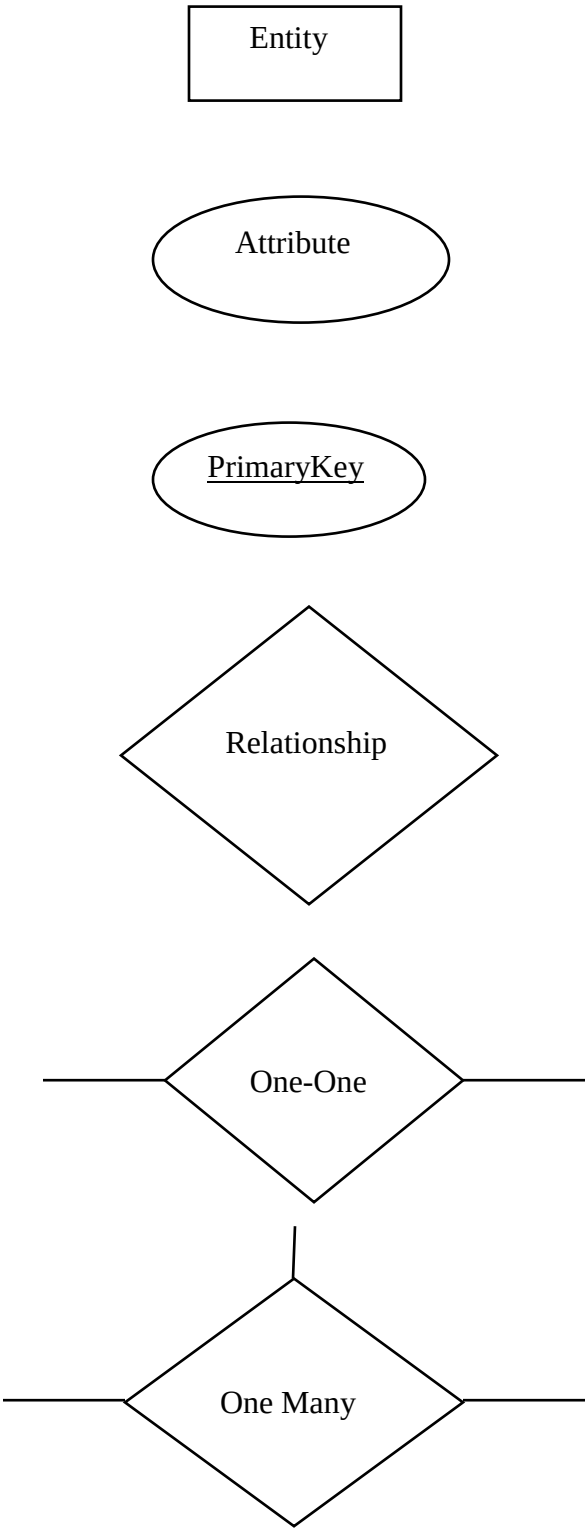


4.5 E-R Diagram

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.



Entity-Relationship Diagram Notations



4.6 Database Description

4.6.1 Table Name: File

The table named File is used to store all the necessary information related to Files which are to be shared using application. The File table has following structure:

Field Name	Data Type	Synonyms	Description
FName	Varchar	File-name	This field stores the name of the file being shared.
Location	Varchar	Address	This field stores the current address of file being shared.
Size	Long	File-size	This field is used to store size of file being shared.

4.6.2 Table Name: User 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.

Field Name	Data Type	Synonyms	Description
Group #	Varchar	Group- number	Stores the group number assigned to particular group of users.
Max #	Integer	Maximum-number	Used to store maximum number of users in a group.

4.6.3 Table Name: User 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.

Field Name	Data Type	Synonyms	Description
UName	Varchar	User-name	Stores the name given by the user using the application.
Group Name	Varchar	Group details	Used to store group name/detail.
IPaddress	Varchar	IP address / Network address	Used to store IP address of machine in order to connect to that particular user.

4.6.4 Table Name: Group

The table named Group is used to store all the necessary information related to group of users using application. The Group table has following structure:

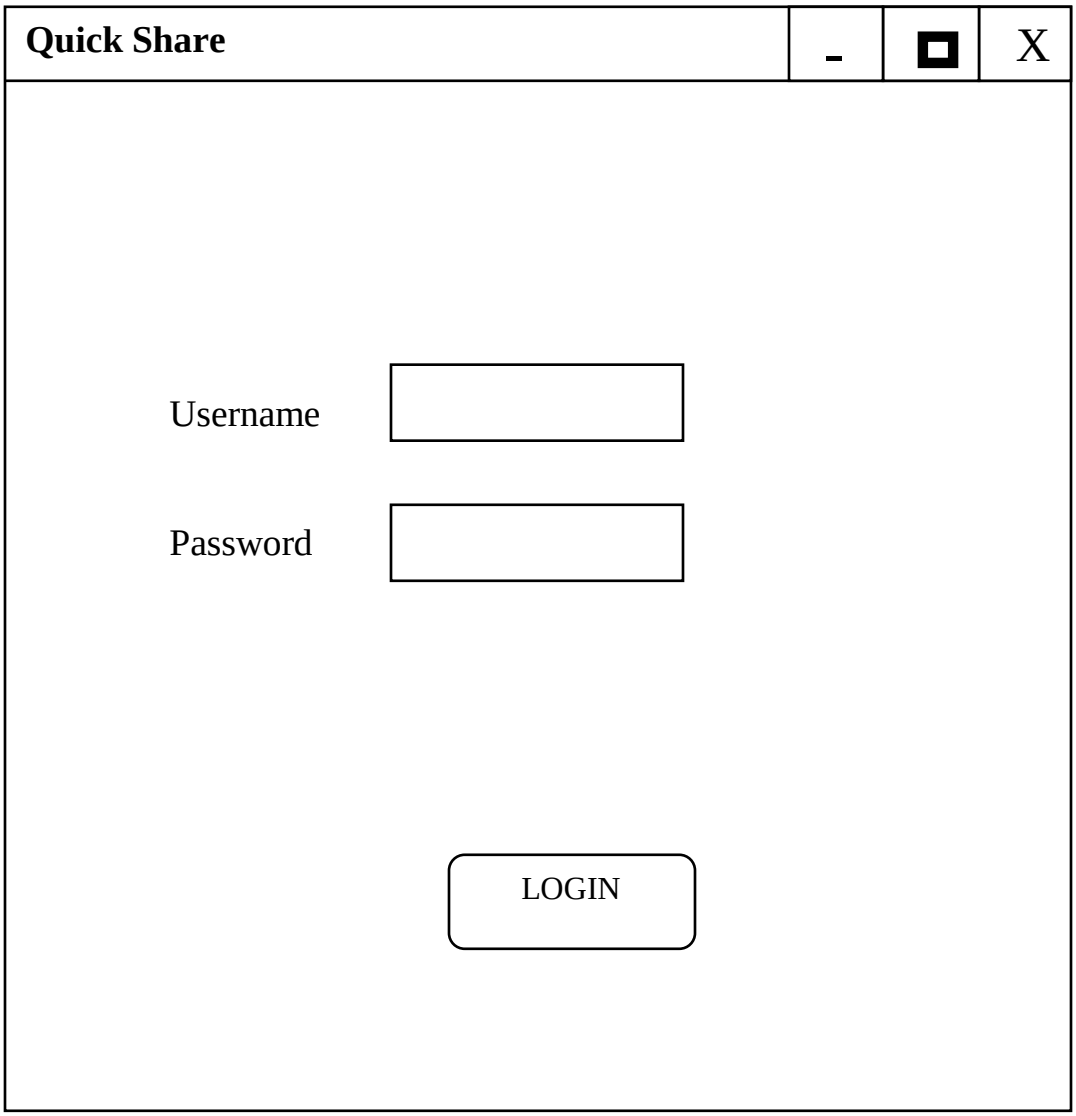
Field Name	Data Type	Synonyms	Description
FName	Varchar	File-name	This field stores the name of the file being shared.
Location	Varchar	Address	This field stores the current address of file being shared.
Size	Long	File-size	This field is used to store size of file being shared.

5.0 User Interface Design

5.1 Form Name: Login

The Login form is the first form/screen the user would see when the application has been started. This form consists of fields for Username and Password created by the user himself for securing the application related data and it's functioning. Apart from above mentioned two text fields there can be options for Recovering passwords, FAQs, Help etc. which are proposed for future work related to the application.

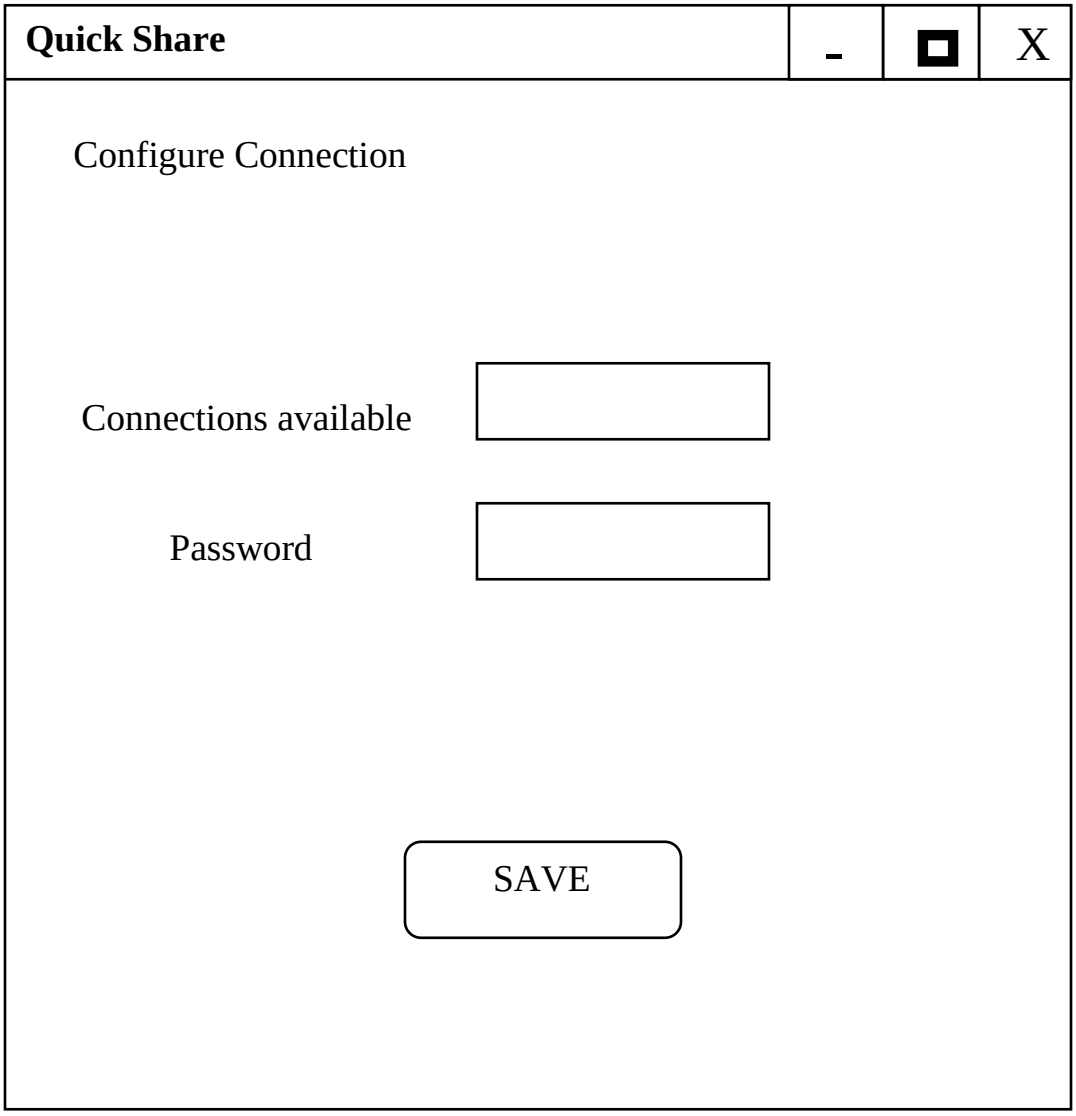
Layout of Login Form



5.2 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.

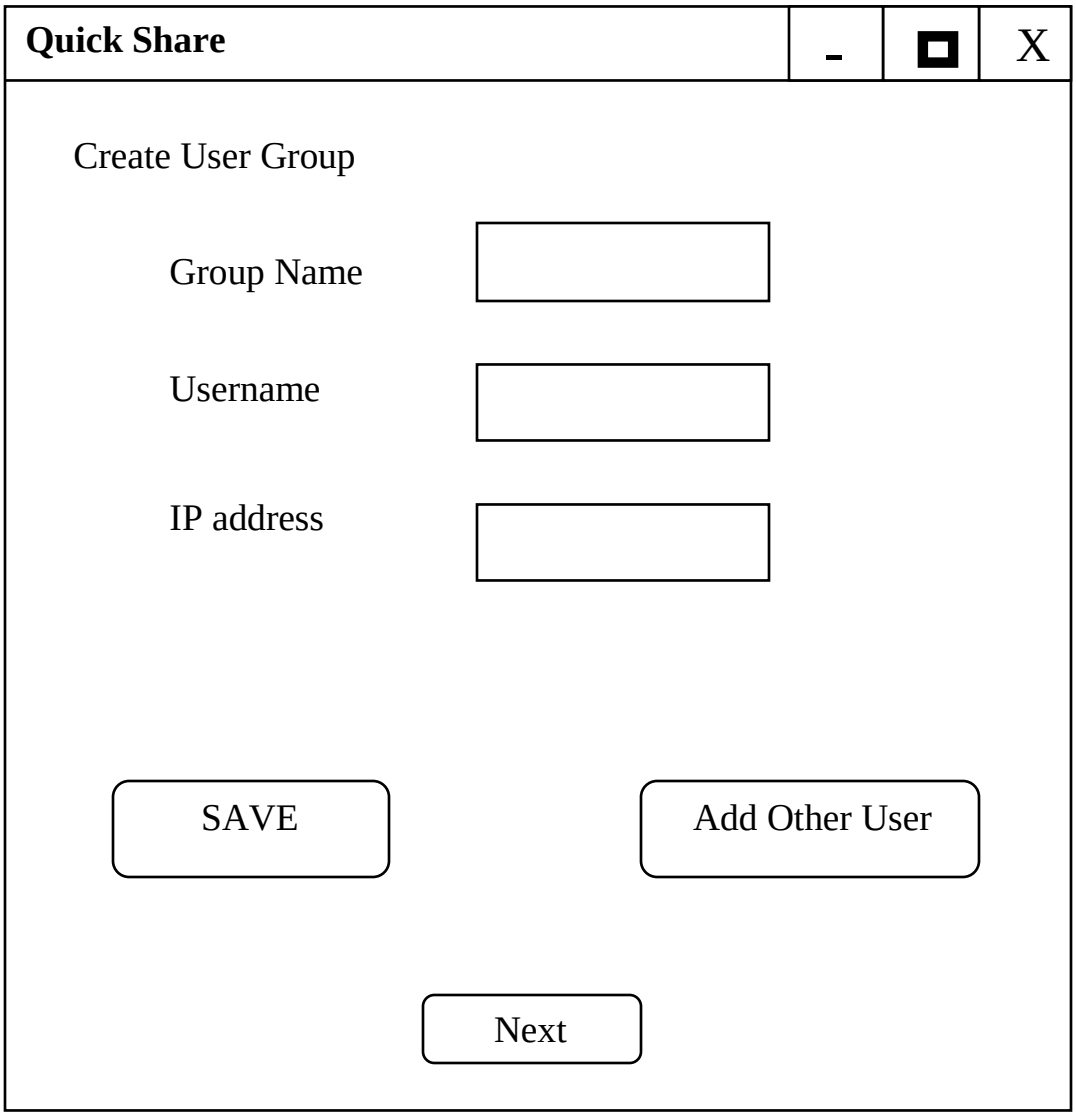
Layout of Set up Connection Form



5.3 Form Name: Create Group

The Create Group form is used to create a new group particularly when application is being used for the first time by the user. Adding new users along with their necessary details etc. all is done using this form.

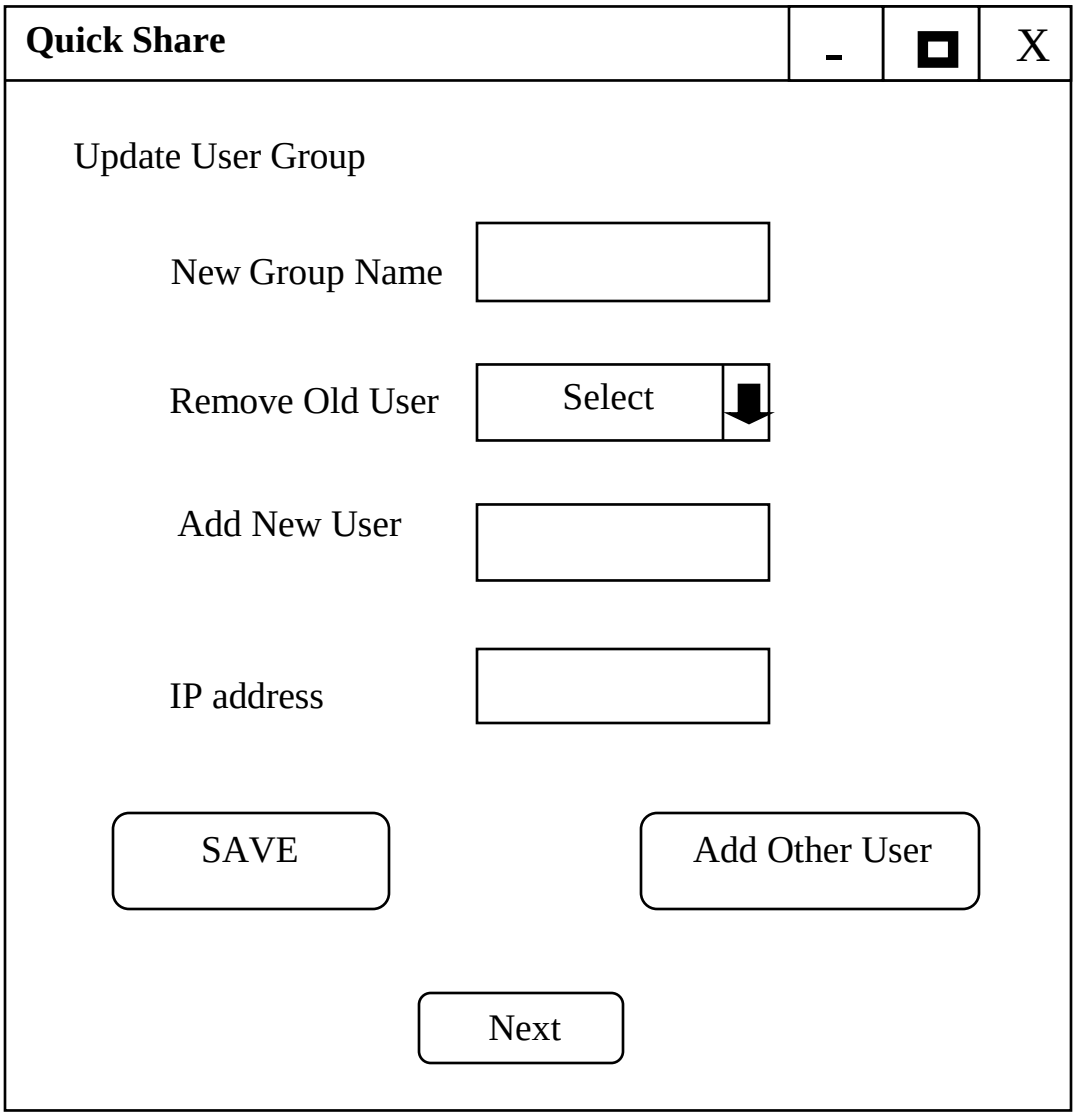
Layout of Create Group Form



5.4 Form Name: Update Group

The Create Group form is used to create a new group particularly when application is being used for the first time by the user. Adding new users along with their necessary details etc. all is done using this form.

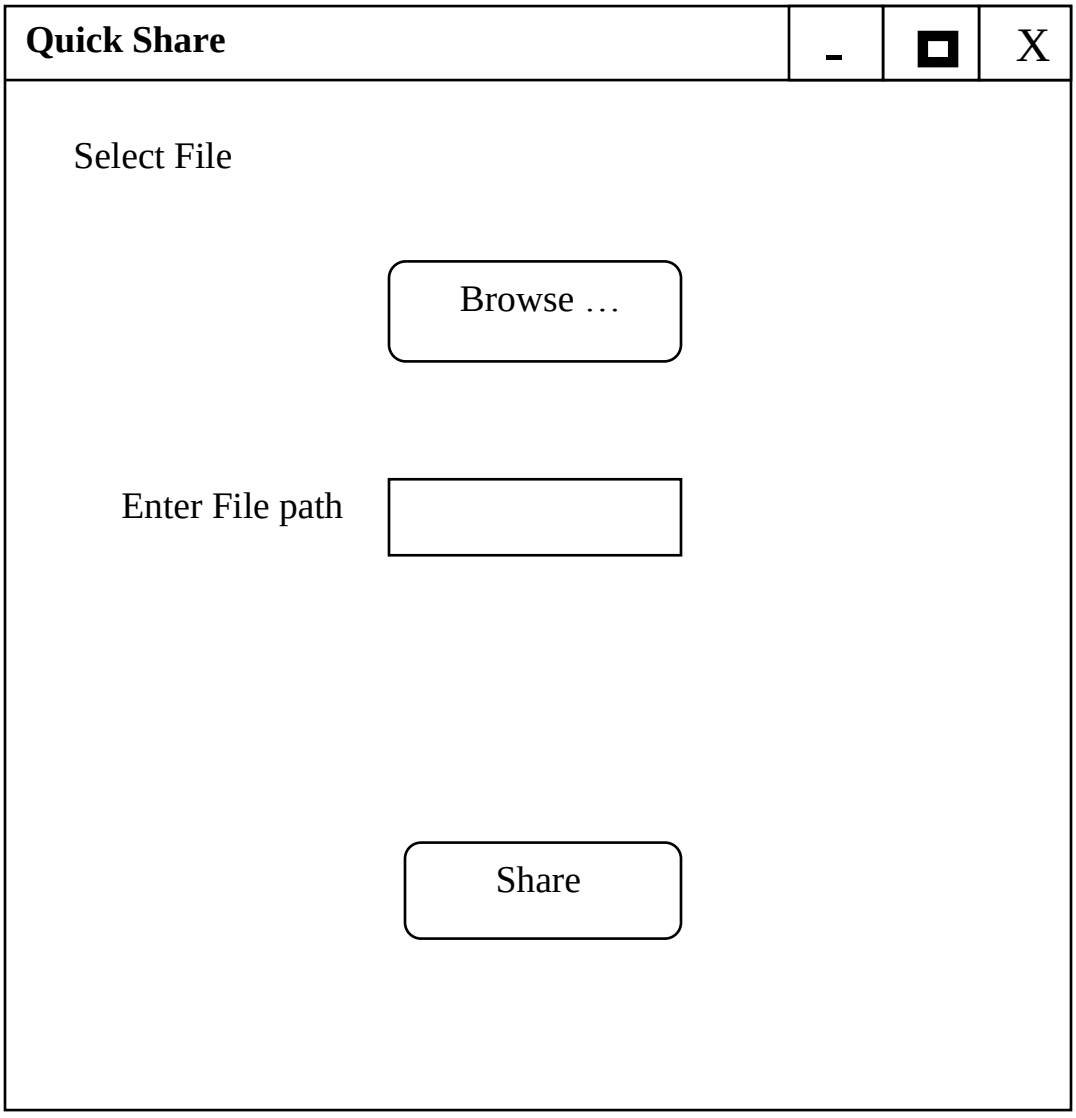
Layout of Create Group Form



5.5 Form Name: Select File

The Select File form is used to select the file to share with other users of group created using Create/Update user group form earlier. This form helps user to locate file in their respective machines' memory using a browse function. After the first configuration of application, the user can share files using single click button on this form for subsequent sharing sessions quickly and with ease.

Layout of Select File Form



6.0 Module Description

6.1 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 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.

6.1.1.1 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.

6.1.1.2 Class Functions

The functions used in this class are:

- Input_Login ()
- ToSetConnection_Page( )

6.1.1.2.1 Function Name: void Input_Login

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

6.1.1.2.1.1 Declaration

void Input_Login (string UName, string Password)

6.1.1.2.1.2 ***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

6.1.1.2.1.3 ***Output Parameters***

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

6.1.1.2.1.4 ***Return values***

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

6.1.1.2.1.5 ***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

6.1.1.2.2 *Function Name:* ToSetConnection_Page ()

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

6.1.1.2.2.1 Declaration

string ToSetConnection_Page(string Status)

6.1.1.2.2.2 ***Input Parameters***

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

6.1.1.2.2.3 ***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.

6.1.1.2.2.4 ***Return values***

Variable Type	Variable Name	Variable Description

Void	N/A	N/A

6.1.1.2.2.5 ***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.

6.2 Module Name: Create Group

The Create Group 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.

6.2.1 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.

6.2.1.1 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.

6.2.1.2 Class Functions

The functions used in this class are:

- void Input_User ()
- ToSendFile_Page( )

6.2.1.2.1 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.

6.2.1.2.1.1 Declaration

void Input_User (string User_Name, string IP_address)

6.2.1.2.1.2 ***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.

6.2.1.2.1.3 ***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.

6.2.1.2.1.4 ***Return values***

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

6.2.1.2.1.5 ***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.

6.2.1.2.2 *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.

6.2.1.2.2.1 Declaration

string ToSendFile_Page(string SendFile_address)

6.2.1.2.2.2 ***Input Parameters***

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

6.2.1.2.2.3 ***Output Parameters***

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

6.2.1.2.2.4 ***Return values***

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

6.2.1.2.2.5 ***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.

6.3 Module Name: Update Group

The Update Group module is used to update set of users created earlier with a predefined size; together they form the end-users of the application. The module has functions to add new user, remove existing user or update details of any user.

6.3.1 Class Name: Update_UserGroup

This class contains all the necessary functions that are needed to update an existing group of users with whom file is to be shared.

6.3.1.1 Class Dependencies

The class Update_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.

6.3.1.2 Class Functions

The functions used in this class are:

- void Add_User ()
- void Remove_User()
- ToSendFile_Page( )
- void Update_User

6.3.1.2.1 Function Name: void Add_User ()

This function takes the input from user regarding the new user to be added to the group.

6.3.1.2.1.1 Declaration

void Add_User (string User_Name, string IP_address)

6.3.1.2.1.2 ***Input Parameters***

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

6.3.1.2.1.3 ***Output Parameters***

Variable Type	Variable Name	Variable Description
String	UName	Adds new user to the database.
String	IP_Addr	Adds IP address of the user in database.

6.3.1.2.1.4 ***Return values***

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

6.3.1.2.1.5 ***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.

6.3.1.2.2 *Function Name:* Remove_User ()

This function any user from the group created earlier by the user of the application.

6.3.1.2.2.1 Declaration

void Remove_User(string UName)

6.3.1.2.2.2 ***Input Parameters***

Variable Type	Variable Name	Variable Description
String	UName	User name of the group member.

6.3.1.2.2.3 ***Output Parameters***

Variable Type	Variable Name	Variable Description
N/A		

6.3.1.2.2.4 ***Return values***

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

6.3.1.2.2.5 ***Pseudo code***

The pseudo code for the function is:

Step 1. Read user name.
Step 2. If user name exists;
Delete User from group and;
Then move to back to form.
Else
Display Error message

6.3.1.2.3 Function Name: **void Update_User ()**

This function takes the input from user regarding updated information of the user to be added to the group.

6.3.1.2.3.1 Declaration

void Add_User (string User_NewName, string IP_NewAddress)

6.3.1.2.3.2 ***Input Parameters***

Variable Type	Variable Name	Variable Description
String	User_NewName	New username of the new user added to the group.
String	IP_NewAddress	New IP address of the new user added to the group.

6.3.1.2.3.3 ***Output Parameters***

Variable Type	Variable Name	Variable Description
String	UName	Adds new user to the database.
String	IP_Addr	Adds IP address of the user in database.

6.3.1.2.3.4 ***Return values***

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

6.3.1.2.3.5 ***Pseudo code***

The pseudo code for the function is:
Step 1. Read new User's name.
Step 2. Read new IP address of added user's computer.
Step 3. Check unique identity of new username.
Step 4. Check new IP address consistency.
Step 5. If User's name and IP address are consistent;
Then move to next form
Else
Request for correct new entries for User's name and IP address.

6.3.1.2.4 Function Name: **void ToSendPage_File ()**

This function takes user to the next form for selecting file that is to be shared.

6.3.1.2.4.1 Declaration

void ToSendPage_File(string User_Status)

6.3.1.2.4.2 ***Input Parameters***

Variable Type	Variable Name	Variable Description
String	User_Status	Current status of the application. If matched against the conditions, then a form for selecting a file is displayed.

6.3.1.2.4.3 ***Output Parameters***

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

6.3.1.2.4.4 ***Return values***

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

6.3.1.2.4.5 ***Pseudo code***

The pseudo code for the function is:

Step 1. Check status signal variable against pre-defined conditions

Step 2. If conditions is matched successfully;

Then move to next form

Else

Display error message.

6.4 Module Name: Set up Connection

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

6.4.1 Class Name: Connect_User

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

6.4.1.1 Class Dependencies

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

6.4.1.2 Class Functions

The functions used in this class are:

- void Check_Connection()
- void Connect_User( )

6.4.1.2.1 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.

6.4.1.2.1.1 Declaration

void Check_Connection (Boolean IP_Status)

6.4.1.2.1.2 ***Input Parameters***

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

6.4.1.2.1.3 ***Output Parameters***

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

6.4.1.2.1.4 ***Return values***

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

6.4.1.2.1.5 ***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.

6.4.1.2.2 *Function Name:* void Connect_User ()

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

6.4.1.2.2.1 Declaration

void Connect_User(string IP_Addr, Boolean IP_Status)

6.4.1.2.2.2 ***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.

6.4.1.2.2.3 ***Output Parameters***

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

6.4.1.2.2.4 ***Return values***

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

6.4.1.2.2.5 ***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.

6.5 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.

6.5.1 Class Name: Select_File

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

6.5.1.1 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.

6.5.1.2 Class Functions

The functions used in this class are:

- Select_File ()

6.5.1.2.1 Function Name: void Select_File

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

6.5.1.2.1.1 Declaration

void Select_File (string File_Name)

6.5.1.2.1.2 ***Input Parameters***

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

6.5.1.2.1.3 ***Output Parameters***

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

6.5.1.2.1.4 ***Return values***

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

6.5.1.2.1.5 ***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.

6.6 Module Name: Share

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

6.6.1 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.

6.6.1.1 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.

6.6.1.2 Class Functions

The functions used in this class are:

- ToSend_Page ()

6.6.1.2.1 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.

6.6.1.2.1.1 Declaration

void ToSend_Page (Boolean Send_Status)

6.6.1.2.1.2 ***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.

6.6.1.2.1.3 ***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.

6.6.1.2.1.4 ***Return values***

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

6.6.1.2.1.5 ***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.

7.0 Definitions and Acronyms

Abbreviation	Description
N/A	Not Applicable
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.

8.0 References

The references used in developing the document are from following sources:

Web

Data Flow Diagram: http://www.smartdraw.com/resources/tutorials/data-flow-diagrams

E-R Diagram: http://www.smartdraw.com/resources/tutorials/entity-relationship-diagrams

Flow Chart: http://www.breezetree.com/articles/what-is-a-flow-chart.htm

Module Description: http://wiki.answers.com/Q/What_is_a_module_in_software_engineering

Function Signature:

http://www.cs.unm.edu/~storm/C++/ProgrammingTerms/FunctionSignatures.html

Text

Software Engineering: A Practitioner's Approach, 7/e
Author: Roger S Pressman, R. S. Pressman & Associates, Inc.