FTP Virtual Drive Explorer

By

Muhammad Arif

A report submitted to Quaid-i-Azam University Islamabad
As a partial fulfillment of the requirements for the
Degree of M.Sc in Computer Science

August 2004

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: 13-09-2004

FINAL APPROVAL

This is to certify that we have read the project report submitted by Mr. Muhammad Arif and it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of Master of Science in Computer Science.

COMMITTEE:

1. External Examiner

   Dr. M. Ayub Alvi
   Dean
   National University of Computer Science &
   Engineering Sciences
   FAST House Rohtas Road
   Sector G-9/4
   Islamabad

2. Supervisor

   Dr. Muhammad Afzal Bhatti
   Professor
   Department of Computer Science
   Quaid-i-Azam University
   Islamabad

3. Chairperson

   Professor Dr. Farhana Shah
   Department of Computer Science
   Quaid-i-Azam University
   Islamabad

Preface:

Chapter 1: It gives brief introduction of FTP Virtual Drive Explorer. It also contains objectives and scope of the system.

Chapter 2: Describes the requirements of the system. It also contains use case diagram and use case description.

Chapter 3: It gives the design of FTP Virtual Drive Explorer including class relationship diagram and classes description. This chapter also includes the sequence diagrams.

Chapter 4: This chapter introduces implementation of the system. It discusses the technology used for the development of the system.

Chapter 5: This chapter named "testing" describes some test cases applied on the FTP Virtual Drive Explorer for testing the working of it.

Dedicated to ....

My Great loving and caring **parents** whose prayers and affection has enabled me to accomplish such a difficult task, to my family members and all my friends, class fellows (*REBELLIONZ*), teachers whose inspirations were the source of encouragement for me.

Acknowledgement

I would like to extend my humble and gratitude to Almighty Allah who endowed me potential and ability to complete this project. During my master program I received the help, support and encouragement from several people. It is really a pleasure for me to acknowledge this cooperation.

First and foremost my *parents*, my *sisters*, *brothers* (**Muhammad Tariq, Naveed Anwar, Amir Ghafoor**) and my lovely *nephew* (**Zaid bin Tariq**) who practically freed me from all responsibilities and who constantly prayed for me through out my academic career.

Greatest thanks to the honorable Chairperson ***Dr. Farhana Shah***, who helped me in every hurdle during study at QAU.

I am greatly thankful to internal supervisor ***Professor Dr. Afzal Bhatti*** for his healthy suggestions and constant encouragement to complete this project report. He was really there to help me whatever he can do. I greatly appreciate his caring attitude to me. Whenever I went to him for some help, He gave me the time even when he was busy.

I would like to pay special thanks to ***Mr. Abdul Jabbar*** my external supervisor from ***Comcept Pvt. Limited.*** for providing me the technical guidance during the development of project. His kind and elder brother's behavior helps me to complete this project in time.

I am greatly thankful to ***Aasim Bhai***, my senior whose guidelines during the final project are really great supplement for my future life.

How can I forget those people ***Aamir Ali Jadoon, Tariq Mehmood Zubairy, Nasir Hussain, Khizer Hayat*** whose absence could disable me to achieve this task.

I will never forget the company of my cute and closet buddies ***REBELLIOZ, Kashif Ali, M.Afzaal, Hassaan Masood, M.Ali*** . Guys I will never forget this association.

Muhammad Arif

4th August 2004.

Table of Contents

Chapter 1
Introduction

Chapter 1

Introduction

1.1 Introduction:

The purpose of this project is to provide a Virtual Drive (folder) on Linux platforms for an Internet client. This folder should enable the user to view and treat remote site's directories like local file systems. The FTP Virtual Drive Explorer will provide some familiar features of Linux Konqueror (File Manager).

1.1.1 What is FTP Virtual Drive Explorer?

The Red Hat Linux user interface gives users access to a wide variety of *objects* necessary for running applications and managing the operating system. The most familiar of these objects are the folders and files that reside on computer disk drives. There are also a number of virtual objects that allow the user to do tasks, such as send files to remote printers or access the Trash Bin.

When the user is connected to any FTP site the contents of the site are shown in the hierarchical structure like main folder containing subfolders and files. Basic idea to develop the FTP Virtual Drive Explorer is to give the user a look and feel as he/she is working on the local system while connecting to Internet.

FTP Virtual Drive Explorer allows a user to connect to server which hosts a site and view its file, folder hierarchy as local folder hierarchy and perform the normal file operations like cut, copy, paste, drag drop etc.

1.1.2 How FTP Virtual Drive Explorer works?

User interacts with the folders through shell, and shell interacts with the folder through standard interfaces. When user interacts with a folder the shell communicates with the folder through one of the interfaces and then folder object does what ever is needed.

manipulation of remote site's files and directories.

3. To enable users to upload and download files and folders using cut, copy, paste, drag and drop commands.

Chapter 2
Requirement Analysis

Chapter 2

Requirement Analysis

2.1 Introduction:

Requirement analysis is a software engineering task that bridges the gap between system-level software allocation and design. Requirements analysis enables the system engineer to specify software function and performance indicate software's interface with other system elements and establish constraints that software must meet. [Pressman, 2001].

2.2 Requirement Engineering

It is the set of activities that leads to the production of requirements definitions and specifications of system.

System requirement include the following sub headings

- Requirements Definitions
    - Functional Requirements
    - Non Functional Requirements
- Requirements Specifications.

2.3 Requirements Definition:

The process of establishing the services the system should provide and constraints under which it must operate is called requirements engineering. A software requirement definition is an abstract description of the services, which the system should provide and the constraints under which the system must operate. It should only specify the external behavior of the system. A software requirement specification is the detail description of what the system should do.

[Somerville 1995].

For example when user double clicks a folder icon in treeview or iconview the folder object enumerates its subfolder and files in the treeview or iconview. Linux Konqueror type browser is the browser for FTP Virtual Drive Explorer.

1.1.3 Why FTP Virtual Drive Explorer?

As mentioned that the Graphical User Interface for FTP Virtual Drive Explorer is Linux Konqueror type browser. Almost all Linux users are familiar with Konqueror and know how to perform file operations (cut, copy, paste, delete, drag drop, etc.) on file folder objects. Since FTP sites also deals with file and folder objects so it makes sense to be able to deal with remote site's directories like local directories and perform the normal operations. And if this functionality is embedded into the explorer then it will be most convenient way to interact with remote site's directories. This is the reason of implementing FTP Virtual Drive Explorer. With this explorer a user can connect to any number of servers and upload and download files and directories using common copy, paste or drag drop operations.

1.2 Scope of the system:

The scope of this system (FTP Virtual Drive Explorer) includes the following.

1. The software should be an explorer and work on Linux platform.
2. It will support both uploading and downloading of files and folders from FTP servers.
3. FTP Virtual Drive Explorer should also support multiple selection.
4. FTP Virtual Drive Explorer should also implement shell context menus.
5. The user should be able to connect with multiple FTP Servers.

1.3 Objectives:

The objectives to be achieved are:

1. To enable the users to view and treat remote site's directories like local file systems.
2. To provides users with familiar interface (like Linux Konqueror) for the

The purpose of this project is to provide a Virtual Drive (folder) on Linux platform for an Internet client. This FTP Virtual Drive Explorer should enable the user to view and treat Site's directories like local file system. The FTP Virtual Drive Explorer should provide important features of Linux Konqueror (explorer).

Since the system is about the implementation of an explorer therefore it must meet the minimum requirements of an explorer. The Linux Konqueror should be able to navigate the FTP Virtual Drive Explorer. It should also implement the folder view in which the user should be able to see files and folders and perform file operations.

Requirements of the system are as follows

2.3.1 Functional Requirements

1. FTP Virtual Drive Explorer should support both uploading and downloading of files and folders.
2. FTP Virtual Drive Explorer should support multiple selections.
3. FTP Virtual Drive Explorer should support the cut, copy and paste operations.
4. FTP Virtual Drive Explorer should support the delete operation.
5. FTP Virtual Drive should support drag and drop facility.
6. FTP Virtual Drive Explorer should implement shell property pages to view properties of the selected items.
7. User should be able to connect to multiple sites.
8. User should be able to create new folders in the FTP Virtual Drive Explorer.
9. User should be able to rename a file or folder in the FTP Virtual Drive Explorer.
10. User should also be able to disconnect any site.

2.3.2 Non-Functional Requirements

Nonfunctional requirements are the constraints under which the services of the system should be provided. Following are the nonfunctional requirements of this system.

2.3.2.1 Product Requirements:

1. The software should be reliable and it should conform to its requirements.
2. Downloading and uploading should not take too much time.
3. While performing paste, delete and drag drop operations standard visual feedback should be given.

2.3.2.2 Process requirements:

1. Qt will be used as a development tool.
2. The software will work on Linux platform.
3. System will be delivered before the end of September 2004.

2.4 Requirement Specification

In it I have specified each and every functional requirement defined above in a precise way using form based method.

Form 1	
Function:	FTP Virtual Drive Explorer should support both uploading and downloading of files and folders.
Description:	If user will cut/copy the selected items from the FTP Virtual Drive Explorer and paste these items in the local drives then explorer will download the selected items. If user will paste the clipboard data in the FTP Virtual Drive Explorer then explorer will upload that data.
Inputs:	Files and folders.
Source:	The user will initiate the operation.
Outputs:	Uploading / Downloading will take place.
Destination:	In case of uploading, connected site. In case of downloading, local drive.
Requires:	Site's address and Internet Connect or FTP Server.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	Clipboard data will upload/download.

Form 2	
Function:	FTP Virtual Drive Explorer should support the cut, copy and paste operation.
Description:	User will select cut, copy or paste commands either from popup menu (by right clicking) or from edit menu of menu bar.
Inputs:	Clipboard data.
Source:	The user will initiate the operation.
Outputs:	If target of cut, copy or paste commands is FTP Virtual Drive Explorer than clipboard data will be uploaded and in case of local system clipboard data will be downloaded.
Destination:	Local system or connected site.
Requires:	Clipboard's data.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site and site's contents must be in the FTP Virtual Drive Explorer also clipboard has the data.
Post-conditions:	If FTP Virtual Drive Explorer is target of cut, copy or paste commands then uploading will start otherwise downloading will start.

Form 3	
Function:	FTP Virtual Drive Explorer should support the delete operation.
Description:	User will select one or more items using mouse or select all the items in the treeview or iconview on clicking the select All submenu of the edit menu. User will select delete command either from popup menu (by right clicking) or from edit menu of menu bar. A pop dialog will appear for confirmation of delete operation if user wants to delete then selected item will delete from the original path.
Inputs:	One or more than one selected items that can only be files and folders.
Source:	The user will initiate the operation.
Outputs:	Selected items will be deleted.
Destination:	Nil.
Requires:	It requires minimum one selected item.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	Selected items will delete from the original path.

Form 4	
Function:	FTP Virtual Drive Explorer should implement shell property pages to view properties of the selected items.
Description:	User will optionally select one or more items from the extension. User will right click to see a context menu. User will click on properties command to see the properties of the selected items.
Inputs:	Nil.
Source:	The user will initiate the operation.
Outputs:	Properties dialog will appear.
Destination:	Nil.
Requires:	Connect Site.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	Properties dialog of selected item or connect site will appear.

Form 5	
Function:	FTP Virtual Drive Explorer should support multiple selections.
Description:	User will select one or more items using mouse or select all the items in the treeview on clicking the select All submenu of the edit menu.
Inputs:	Nil.
Source:	The user will initiate the operation.
Outputs:	Items will be selected.
Destination:	Nil.
Requires:	It requires files and folders in the connected site.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	Items will select

Form 6	
Function:	User should be able to connect to multiple sites.
Description:	User will select connect command from context menu to connect to a site.
Inputs:	Address of site, and optionally user name and password.
Source:	The user will initiate the operation.
Outputs:	Site will connect or error message will appear in case of wrong site's address, user name or password.
Destination:	FTP Virtual Drive Explorer.
Requires:	It requires Site's address and optionally username and password.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	For connected site a new folder will appear in the FTP Virtual Drive Explorer.

Form 7	
Function:	User should be able to disconnect any sites.
Description:	User will select a site. User will right click to see an appropriate context menu. User will click the disconnect command.
Inputs:	Nil
Source:	The user will initiate the operation.
Outputs:	Site will disconnect.
Destination:	FTP Virtual Drive Explorer.
Requires:	Nil.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site.
Post-conditions:	Site will disconnect.

Form 8	
Function:	User should be able to create new folders in the FTP Virtual Drive Explorer.
Description:	User will select File->New->Folder command from menu bar or NewFolder from the popup menu to create a new folder.
Inputs:	New folder name.
Source:	The user will initiate the operation.
Outputs:	New folder will create in the FTP Virtual Drive Explorer.
Destination:	FTP Virtual Drive Explorer.
Requires:	New folder name.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	New folder will be created in the FTP Virtual Drive Explorer.

Form 9	
Function:	User should be able to rename a file or folder in the FTP Virtual Drive Explorer.
Description:	User will select rename from the popup menu to rename a file or folder.
Inputs:	New name for file or folder.
Source:	The user will initiate the operation.
Outputs:	File/folder will be renamed.
Destination:	FTP Virtual Drive Explorer.
Requires:	New name for file or folder.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	File/folder will be renamed

Form 10	
Function:	FTP Virtual Drive Explorer should support drag and drop facility.
Description:	FTP Virtual Drive Explorer will be drag source or drag target or both. In target mode user will drop selected items (from local system) into or at some folder in the FTP Virtual Drive Explorer. FTP Virtual Drive Explorer will upload selected items. In source mode user will select one or more items from the FTP Virtual Drive Explorer and drop into the system. FTP Virtual Drive Explorer will download selected items at the target path. When FTP Virtual Drive Explorer will be both a source and target of a drag drop operation. User will select one or more items from FTP Virtual Drive Explorer and will drop at some other folder in the FTP Virtual Drive Explorer (source and target folders may be on the same Site or on different sites). The FTP Virtual Drive Explorer will transfer selected items from site to site.
Inputs:	File/folder.
Source:	The user will initiate the operation.
Outputs:	If FTP Virtual Drive Explorer is drop target then dropped file/folder will be uploaded and in case of local system dropped file/folder will be downloaded.
Destination:	Local system or connected site.
Requires:	Connected site.
Pre-conditions:	FTP Virtual Drive Explorer must be connected to a site, and site contents must be in the FTP Virtual Drive Explorer.
Post-conditions:	If FTP Virtual Drive Explorer is drop target then dropped file/folder's uploading start and in case of local system dropped file/folder's downloading start.

2.5 Use Cases of the system

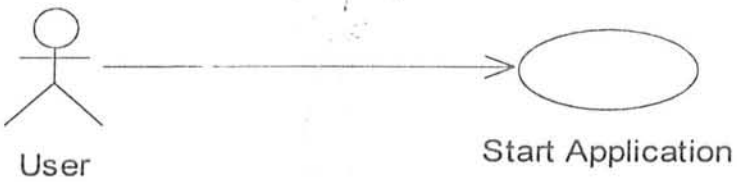
2.5.1 Actors of my system:

User: Only actor of my system is the User of the system.

2.5.2 Use Case Description

Here is the description of the use cases identified above.

2.5.2.1 Start Application



User: User of the system

Purpose: Successfully start the Application.

Overview: After starting the application, the GUI of FTP Virtual Drive Explorer will appear on the screen.

Pre-requisite: FTP Virtual Drive Explorer should be installed.

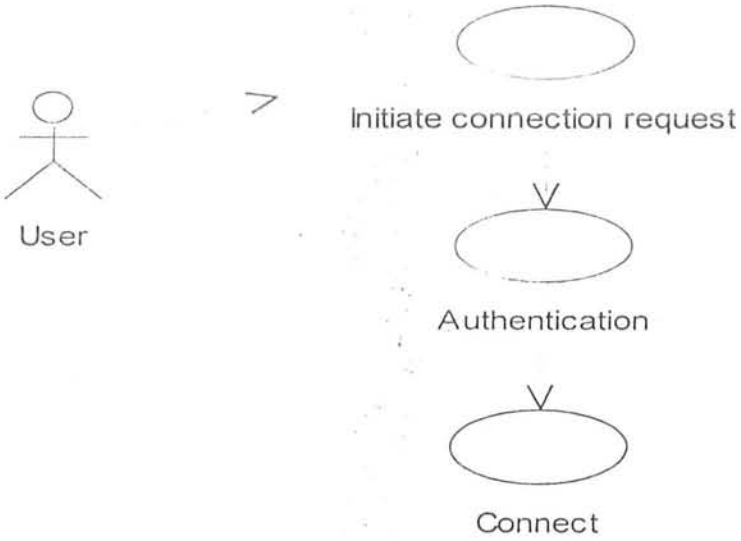
Typical course of action

Sr.#	User's Action	System Response
1	User will click the desktop short cut or executable on the start menu program.	
2		System will display the GUI of the FTP Virtual Drive Explorer and will display the standard icon.
3	User will click the right mouse button	
4		System will display the context menu.

Alternative course of actions

1. None

2.5.2.2 Connect Site



User: User of the system

Purpose: Successfully connect to the multiple sites

Overview: System provides connection dialog and verification process for security measures.

Pre-requisite: Use case: Start Application

Typical course of action

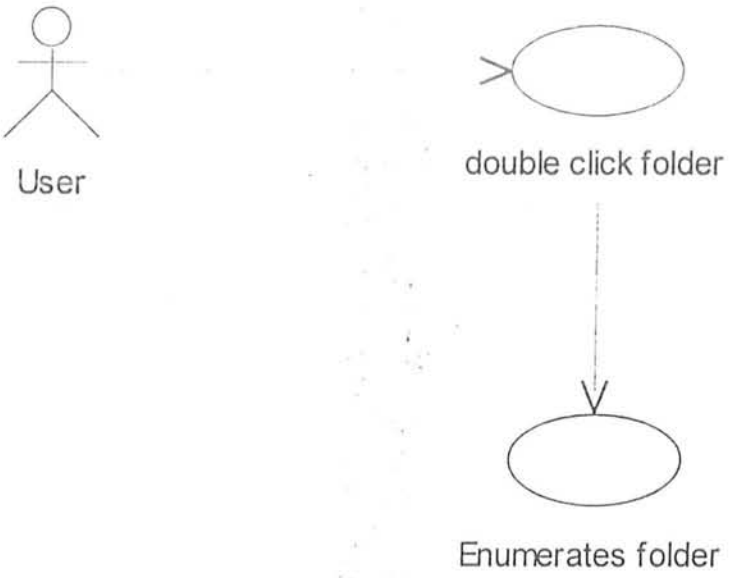
Sr.#	User's Action	System Response
1	User will click the connection button on tool bar or select the connection menu from the menu bar.	
2		The system will open a connection dialog and will require username and IP address of the FTP server.
3	User will enter the username and IP address of the FTP server and will click on the connect button.	
4		System will verify the information

		provided by the user. A new folder named as site address will appear in the FTP Virtual Drive Explorer

Alternative course of actions

1. System will verify the information and will give a message that connection information is incorrect in case of invalid username and IP address of the server.
2. Connection module will be aborted if user presses the cancel button on the connection dialog.

2.5.2.3 Enumerates folders



User: User of the system

Purpose: To enumerate folders.

Overview: Show subfolders and file of any folder.

Pre-requisite: Use case: Start Application.

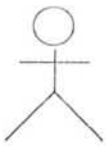
Typical course of action

Sr.#	User's Action	System Response
1	User will double click any folder in system.	
2		The system will enumerate the folder.

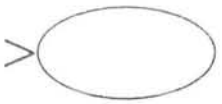
Alternative course of actions

1. System will not enumerate the folder if that is empty.

2.5.2.4 Select item



User



select item

User: User of the system

Purpose: To select one or more than one items at a time.

Overview: Select single or multiple files or subfolders of any folder.

Pre-requisite: Use case: Start Application.

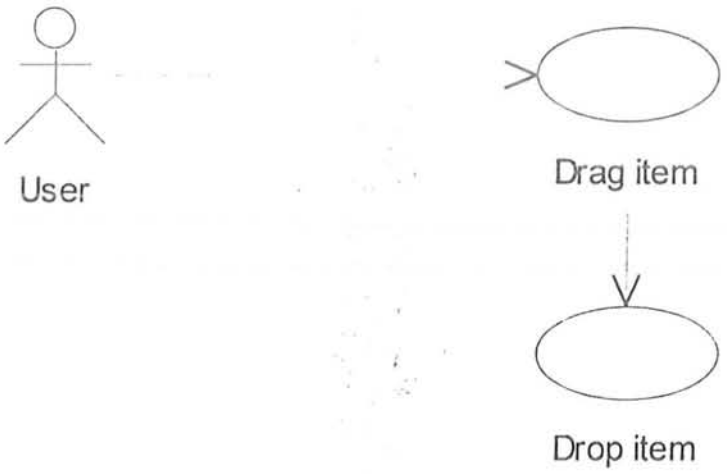
Typical course of action

Sr.#	User's Action	System Response
1	User will click any folder in system.	
2		The system will allow the user to select one or multiple files / folders.

Alternative course of actions

1. System will not allow multiple selection if the folder is empty.

2.5.2.5 Support drag and drop



User: User of the system

Purpose: Provide drag and drop facility.

Overview: Drag any item and drop it any where in the system.

Pre-requisite: Use case: Start Application.

Use case: Select Item.

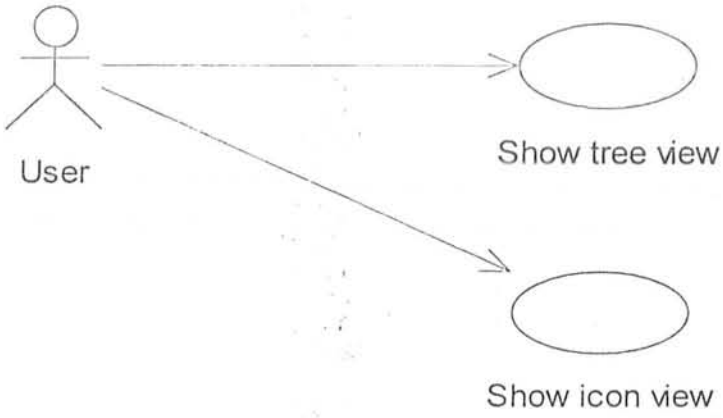
Typical course of action

Sr.#	User's Action	System Response
1	User will drag any file or folder in system and drop it at any location.	
2		The system will cut that item and paste it at the destination.

Alternative course of actions

1. None

2.5.2.6 Show different views



User: User of the system

Purpose: Provide different views.

Overview: User can select iconview or treeview.

Pre-requisite: Use case: Start Application.

Use case: Select Item.

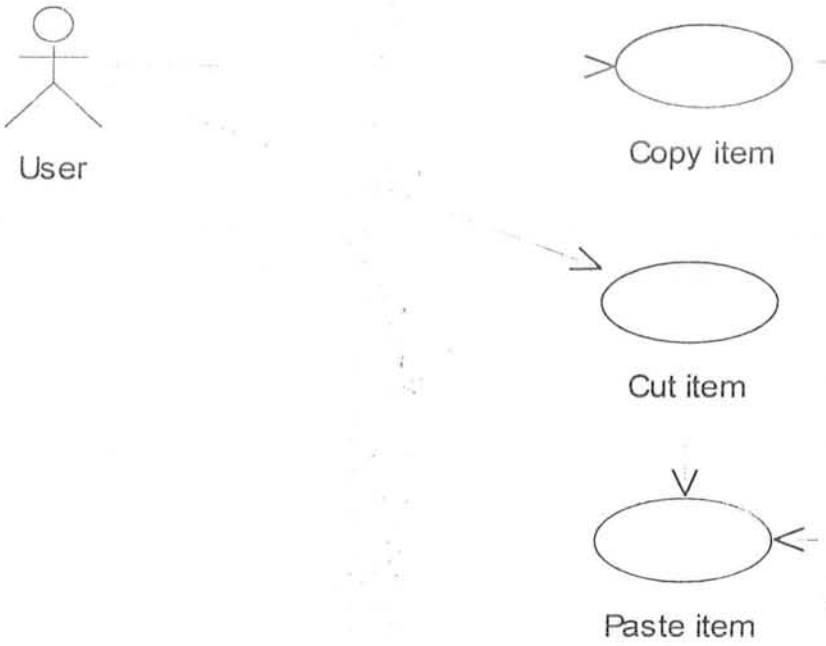
Typical course of action

Sr.#	User's Action	System Response
1	User will select iconview or treeview from the menu bar or user will click on the iconview or treeview buttons on the tool bar.	
2		The system will switch to the required view.

Alternative course of actions

1. None

2.5.2.7 Support cut, copy, paste



User: User of the system

Purpose: Provide cut, copy, paste facility.

Overview: User can cut / copy files or folders and paste them any where in system.

Pre-requisite:Use case: Start Application.

Use case: Select Item.

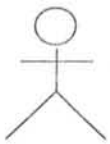
Typical course of action

Sr.#	User's Action	System Response
1	User will select cut, copy, paste options from the menu bar or from popup menu by right clicking the item, or by using the short cut keys after selecting the items.	
2		The system will allow the user to cut, copy, paste items.

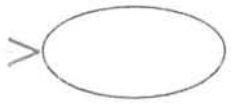
Alternative course of actions

1. System will not allow cut, copy, paste operations if the item does not exists.

2.5.2.8 Support delete



User



Delete item

User: User of the system

Purpose: Provide delete facility.

Overview: User can delete items in the system.

Pre-requisite: Use case: Start Application.

Use case: Select Item.

Typical course of action

Sr.#	User's Action	System Response
1	User will select delete option from the menu bar or from popup menu by right clicking the item, or by using the delete key from keyboard after selecting the items.	
2		The system will delete items.

Alternative course of actions

1. System will not allow delete operation if the item does not exists or not selected.

2.5.2.9 Support rename

User: User of the system

Purpose: Provide rename facility.

Overview: User can rename items any where in system.

Pre-requisite: Use case: Start Application.

Use case: Select Item.

Typical course of action

Sr.#	User's Action	System Response
1	User will select rename option from the menu bar or from popup menu by right clicking the item after selecting the items.	
2		The system will allow the user to rename the folder.

Alternative course of actions

1. None

2.5.2.10 Create new folder

User: User of the system

Purpose: Provide facility to create new folder.

Overview: User can create new folder any where in system.

Pre-requisite: Use case: Start Application.

Typical course of action

Sr.#	User's Action	System Response
1	User will select create new folder option from the menu bar or from popup menu by right clicking in the icon view or treeview.	
2		The system will allow the user to create new folder.

Alternative course of actions

1. System will not allow create new folder if disk is full.

2.5.2.11 Disconnect sites

User: User of the system

Purpose: Disconnect any site.

Overview: User can disconnect any site from system.

Pre-requisite: Use case: Start Application.

Use case: Connect Site.

Typical course of action

Sr.#	User's Action	System Response
1	User will select disconnect option from the menu bar or by clicking the disconnect button from icon bar.	
2		The system will allow the user to disconnect any site.

Alternative course of actions

1. System will not allow disconnect operation if the user not connected to that site.

Chapter 3
Design

Chapter 3

Design

3.1 Introduction

This chapter will give an overview of different classes that were identified during the system analysis and design phase. This chapter also explains some sequence diagrams and interface diagram.

3.2 Design

Design is a meaningful engineering representation of something that is to be built. Design of an object-oriented software is the definition of multilayered software architecture, the specification of subsystem that perform required functions and provide infrastructure support a description of objects that form the building blocks of the system.[Pressman 2001]

Object Oriented Design is divided in two major activities

a) System Design

Creates the product architecture, defining a series of "layers" that accomplish specific system functions and identifying the classes that are encapsulated and identifying the classes that are encapsulated by subsystems that reside at each layer.

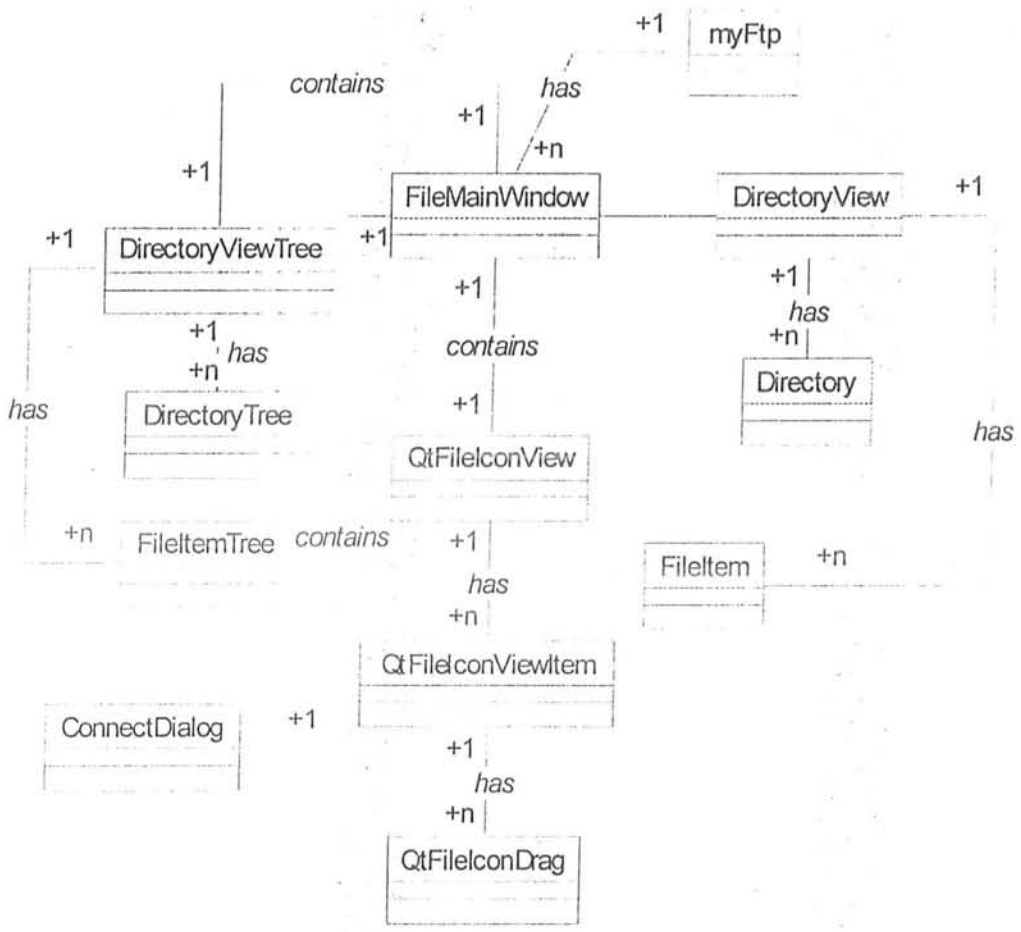
b) Object Design

Focuses on the internal detail of individual classes, defining attributes, operations and message details.

Now object oriented design of the FTP virtual drive explorer is presented in the next section.

3.3 Class Relationship Diagram

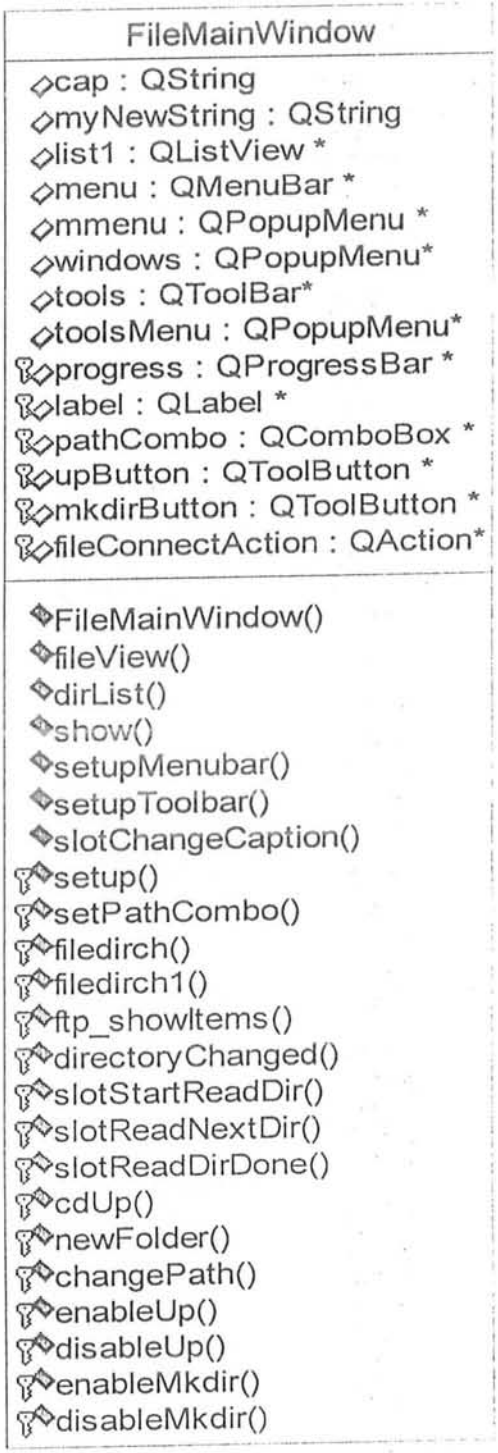
Objects of classes communicate with each other to perform any functions. The type of relationship at class level defines this communication. The interaction relationships are defined in class relationship diagram. Class relationship diagram is a UML (unified modeling language) standard for the describing class interaction.



3.4 Class Description

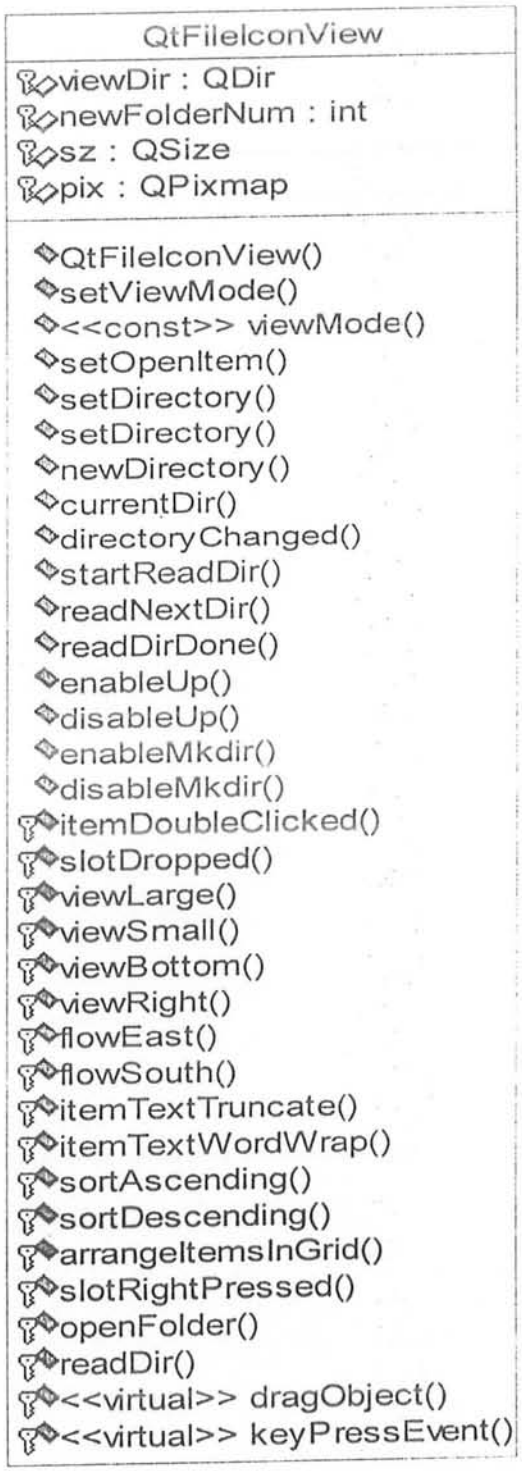
FileMainWindow :

This is the main class of the whole project. User can view different views like icon view or tree view by selecting option from menu bar or from tool bar and menu bar. This class allow user to connect to the FTP server by providing a connection dialog from the menu bar or from tool bar.



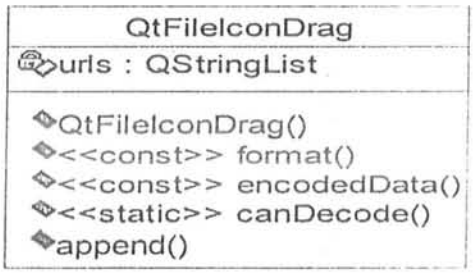
QtFileIconView :

This class is used to create an icon view to the user in the main window of the project.



QtFileIconDrag :

This class allow the drag and drop of items in the icon view.

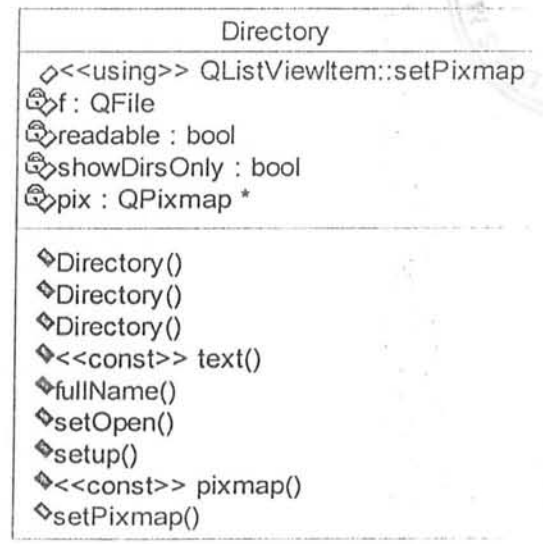


DirectoryView :

This class is used to create a directory view on the left side of the splitter in both icon and tree views.

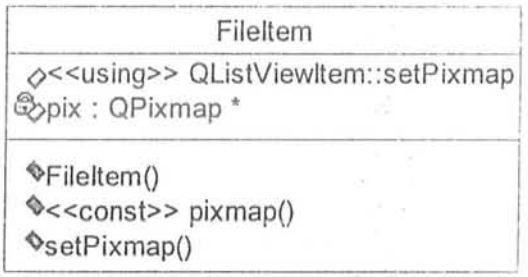
Directory :

This class is used to create or display the directories in the directory view for the user to proceed further at run time. All of the directories are of the system / directory.



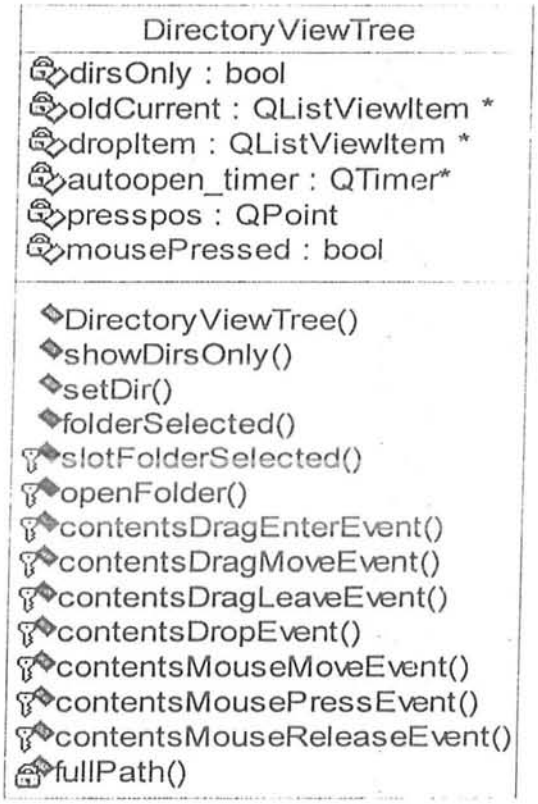
FileItem :

This class is used to display the files in the directory view of the system directory /.



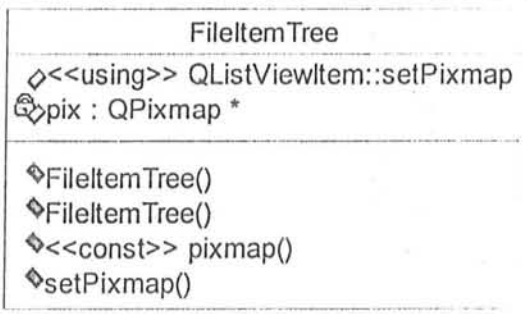
DirectoryviewTree :

This class is used to create a tree view for the user in the main window.



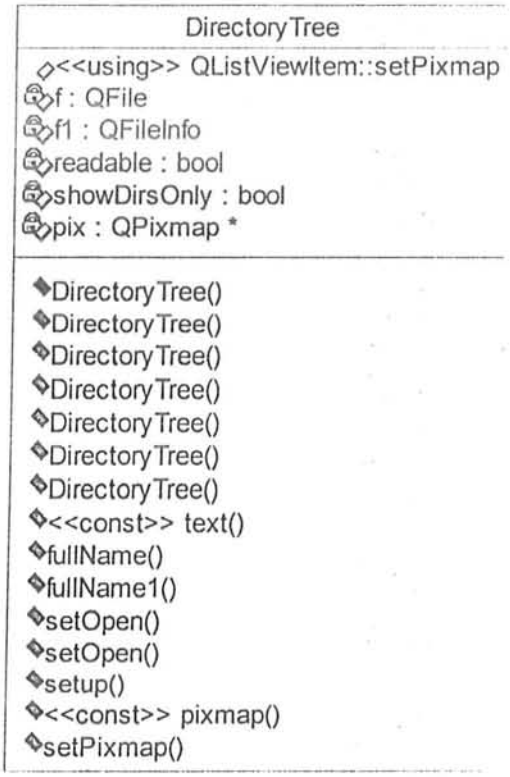
FileItemTree :

This class is used to display or create new files in current directory or in the subdirectories for the user in tree view.



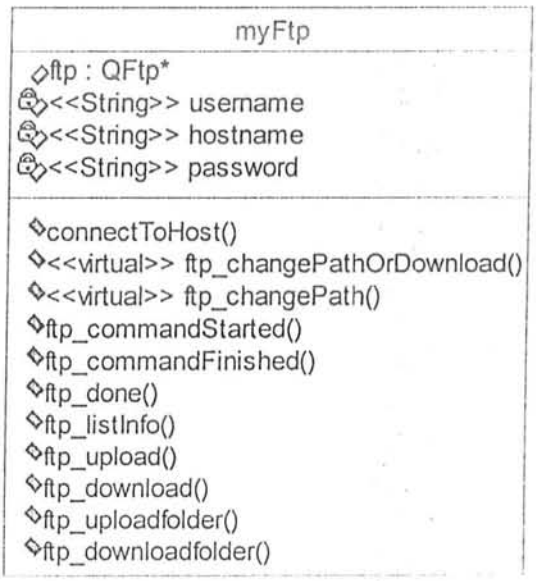
DirectoryTree :

This class is used to display or create new directory for the user in the tree view.



myFtp :

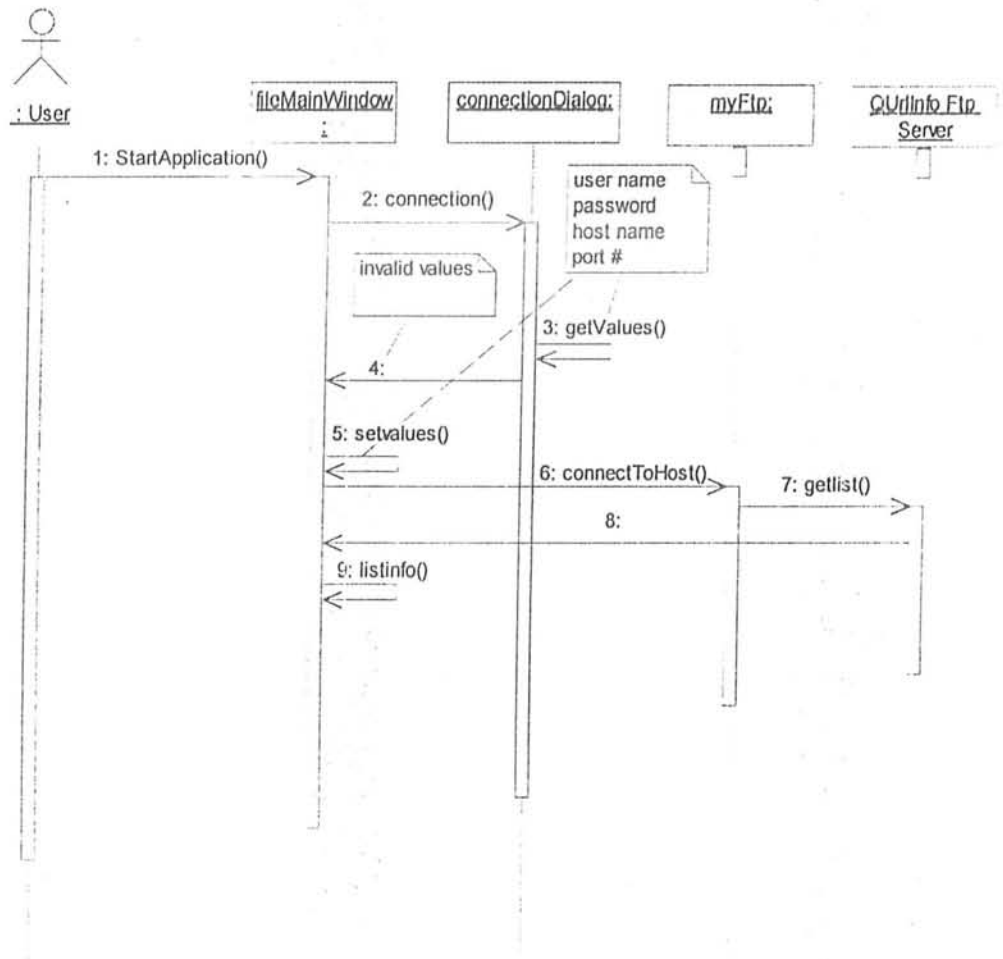
This class allow user to perform different functionalities of the FTP client.



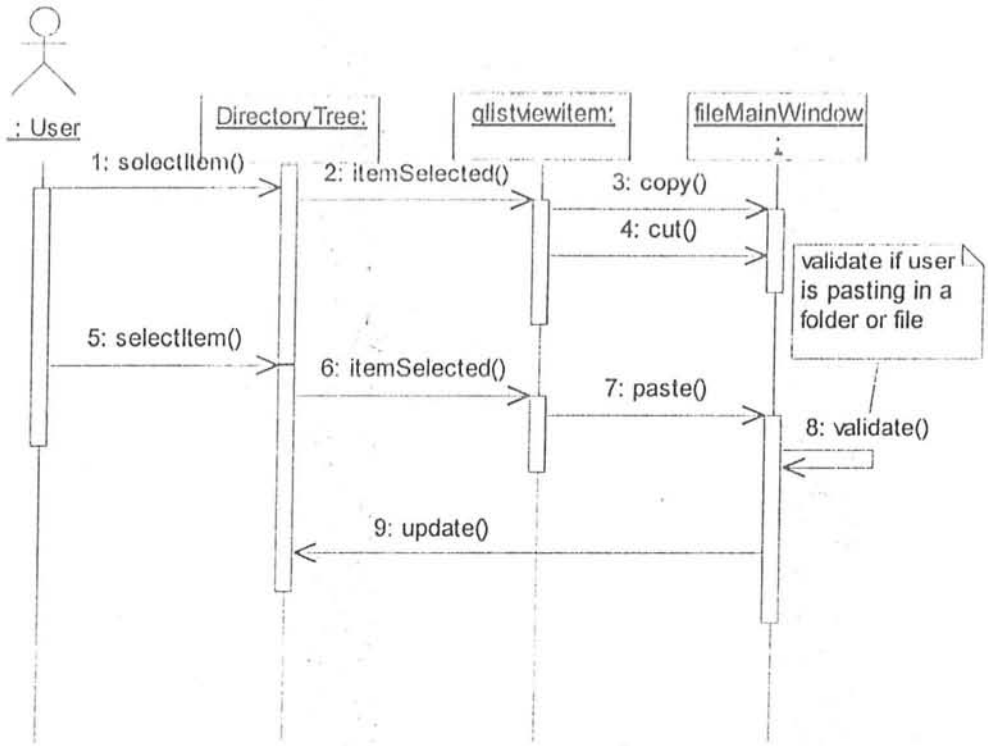
3.5 Sequence diagrams:

Objects interact with each other to get the required functionality by passing message between each other. The messages between objects have to be in a sequence to have meaningful functionality. Sequence diagram defines the sequence of messages. Sequence diagram show the dynamic collaboration between a numbers of object. In other words, it shows how objects communicate with each other.
Different sequence of activities represents different sequence diagrams in the system. Here is a list of some of the scenarios of the system.

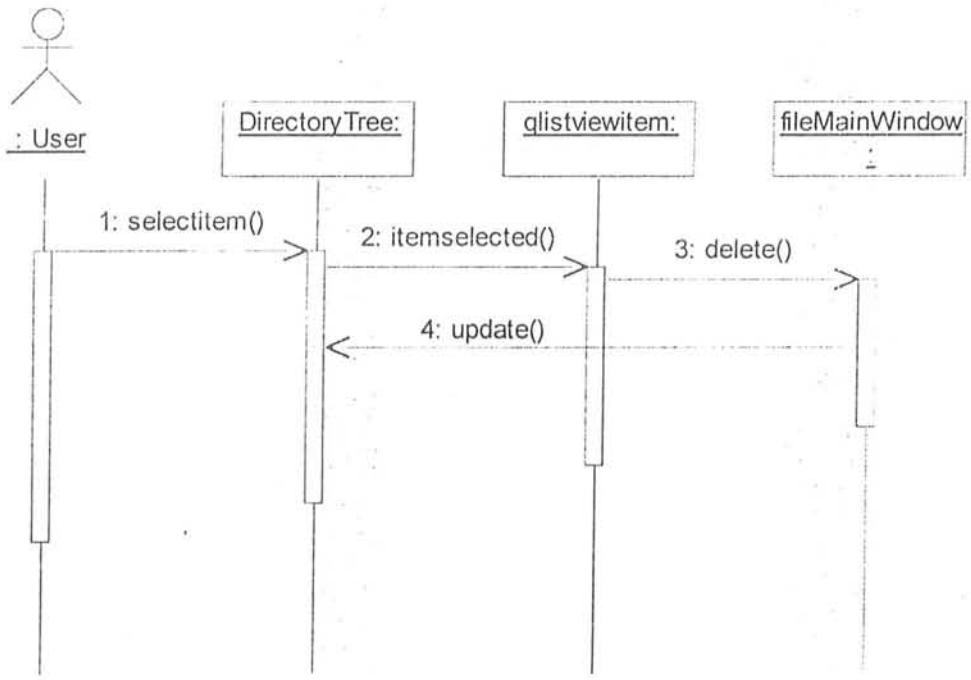
Connect site:



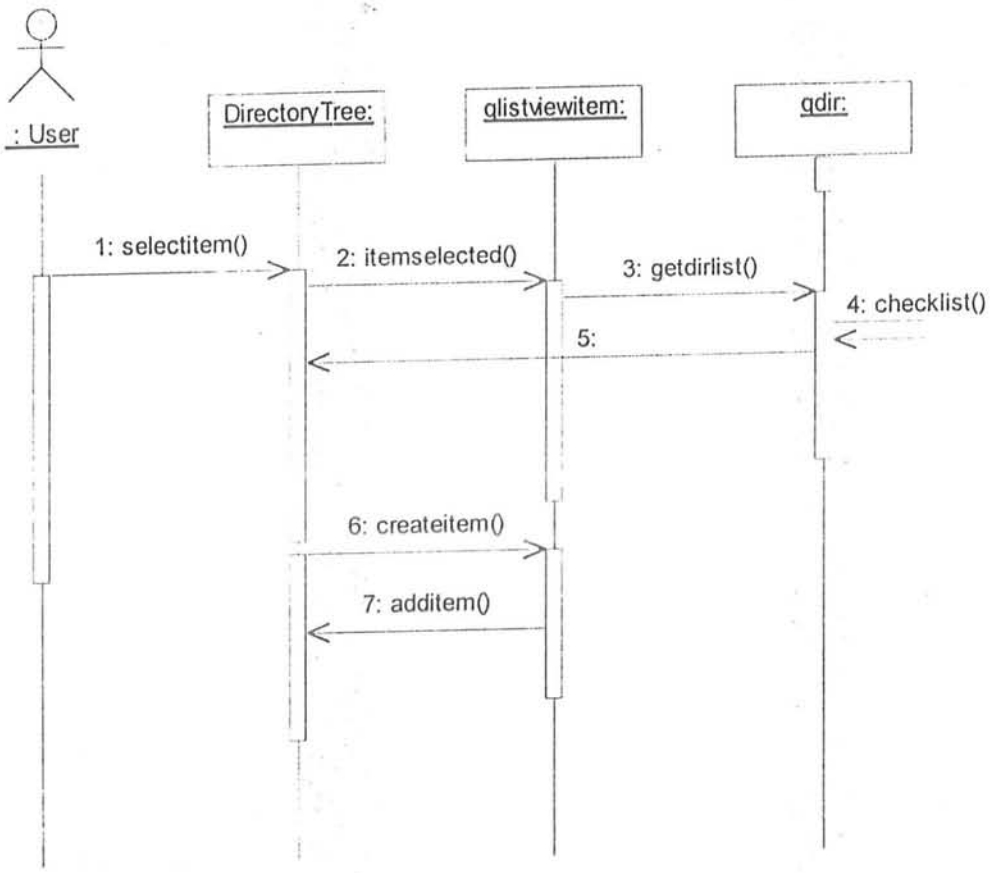
Cut, copy, paste :



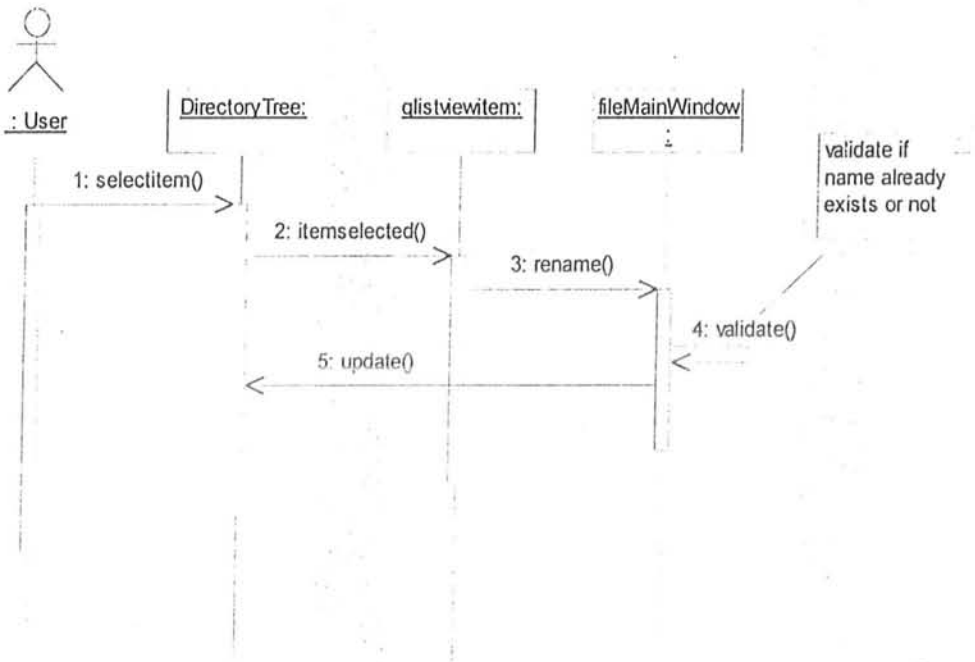
delete:



Enumerates folder:



Rename:


: User
DirectoryTree:
qlistviewitem:
fileMainWindow
validate if name already exists or not
1: selectitem()
2: itemselected()
3: rename()
4: validate()
5: update()

Chapter 4
Implementation

Chapter 4

Implementation

4.1 Introduction

In this chapter different technologies that are used during the development of this project are discussed.

4.2 Technology Used

Following technologies has been used in this software.

- G++
- Qt libraries and designer (version 3.1.1)
- qmake

4.2.1 G++

G++ is a framework of classes for object-oriented programming with the C++ language, a library-based extension of C++ to deal with concurrency, a method for analysis and design of distributed systems, especially Computer Integrated Manufacturing (CIM) systems, as well a Computer Aided Software Engineering (CASE) tool which encompasses the software life cycle through analysis, design, code generation, prototyping through simulation and emulation in distributed environment.

G++ offers a vast library of C++ basic classes, classes which can deal with concurrency and distribution and classes which can interface with X-Window system.

The libraries are mainly divided into three parts:

- OAK (Object Architecture Kernel) is composed of smalltalk-like data structures.
- Rodin is a library which extends the C++ language with classes which deal with concurrency and distribution.

- Mondrian is a library of graphical objects which interface with X- Window.

4.2.2 Qt libraries and designer (3.1.1)

Since we have to develop the software on Linux platform and also it is required to develop designer that must appear as strong GUI. The Linux Konqueror for KDE environment is also developed in Qt so we used Qt to develop FTP Virtual Drive Explorer.

Qt is a multiplatform C++ GUI application framework. It provides application developers with all the functionality needed to build applications with state-of-the-art graphical user interfaces. Qt is fully object-oriented, easily extensible, and allows true component programming.

Qt has formed the basis of many thousands of successful applications worldwide. Qt is also the basis of the popular KDE Linux desktop environment, a standard component of all major Linux distributions.

Qt is supported on the following platforms:

- MS/Windows -- 95, 98, NT 4.0, ME, 2000, and XP
- Unix/X11 -- Linux, Sun Solaris, HP-UX, Compaq Tru64 UNIX, IBM AIX, SGI IRIX and a wide range of others
- **Macintosh** -- Mac OS X
- **Embedded** -- Linux platforms with frame buffer support.

Qt is a product of Trolltech.

4.2.2.1 Qt Designer

Qt designer is a full fledged GUI builder. It includes powerful features such as preview mode, automatic widget layout, support for custom widgets, and an advanced property editor.

Qt designer makes it easy to experiment with user interface design. At any time you can generate the code required to reproduce the user interface from the files Qt designer produces, changing your design as often as you like.

Qt designer helps you build user interfaces with layout tools that move and scale your widgets (controls in Windows terminology) automatically at runtime. The resulting interfaces are both functional and attractive, comfortably suiting your users' operating environments and preferences.

4.2.2.2 In Line C++ code editing

In Qt designer, you can do in-line editing of C++ code. Our editor simplifies the process using syntax highlighting, completion, pattern matching, and incremental search.

4.2.3 qmake

qmake is a tool created by Trolltech to write makefiles for different compilers and platforms.

Writing makefiles by hand can be difficult and error prone, especially if several makefiles are required for different compiler and platform combinations. With qmake, developers create a simple single 'project' file and run qmake to generate the appropriate makefiles. qmake takes care of all the compiler and platform dependencies, freeing developers to focus on their code. Trolltech uses qmake as the primary build tool for the Qt library, and for the tools supplied with Qt.

qmake also takes care of Qt's special requirements, automatically including build rules for moc and uic.

Chapter 5
Testing

Chapter 5

Testing

5.1 Objectives of Testing

Testing of the application is done to achieve following objectives.

1. Execution of the program is done with intent to find the errors in the program.
2. Test cases are designed so that these have high probability of finding an as-yet-undiscovered error.

5.2 Black Box Testing

Black box testing focuses on the functional requirements of the software. Black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. [Pressman 2001]

5.3 Test Cases

Test cases are designed so that these maps with following functional requirements of the system.

1. Connect to a Site.
2. Downloading a file from the FTP Virtual Drive Explorer.
3. Downloading a folder from the FTP Virtual Drive Explorer.
4. Downloading multiple folders from the FTP Virtual Drive Explorer.
5. Uploading a file on the connected server.
6. Uploading a folder on the connected server.
7. Make a new folder in the FTP Virtual Drive Explorer.
8. Disconnect a Site

Req. No.	1
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Connection of FTP Virtual Drive Explorer with the FTP servers is tested by giving server address. The objective of this test is to conformation of connection building of FTP Virtual Drive Explorer with the FTP servers.
Input	Address of Server and optionally Username and password
Expected Output	FTP Virtual Drive Explorer will connect to specified server.
Actual Output	FTP Virtual Drive Explorer has connected to specified server.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being shown.

Req. No.	2
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Downloading a file from the connected server is taken place by copying that file from the FTP Virtual Drive Explorer and pasting it in the local System.
Input	Connected Server's File.
Expected Output	File will download from the connected server at the target path
Actual Output	File has downloaded from the connected server in the targeted folder.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being shown.

Req. No.	3
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Downloading a folder from the connected server is taken place by copying that folder from the FTP Virtual Drive Explorer and pasting it in the local System.
Input	Connected Server's Folder.
Expected Output	Folder will download from the connected server at the target path.
Actual Output	Folder has downloaded from the connected server at the target path.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being shown.

Req. No.	4
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Downloading multiple folders from the connected server is taken place by copying those folders from the FTP Virtual Drive Explorer and pasting it in the local System.
Input	More than one folders of Connected Server.
Expected Output	All Folders will download from the connected server at the target path.
Actual Output	All Folders has downloaded from the connected server at the target path.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being attained

Req. No.	5
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Uploading a file on the connected server is taken place by copying that file from the local system and pasting it in the Connected Server.
Input	A file of local System.
Expected Output	File will Upload on the connected server.
Actual Output	File has Uploaded on the connected server.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being attained

Req. No.	6
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Uploading a folder on the connected server is taken place by copying that folder from the local system and pasting it in the Connected Server.
Input	A folder of local System.
Expected Output	Folder will Upload on the connected server.
Actual Output	Folder has Uploaded on the connected server.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being attained

Req. No.	7
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Creating a new folder in the Connected Server is taken place by selecting the menu item NewFolder from the popup menu or from the File/New/Folder menu. This will show dialog box for getting new folder's name by default new folders name will "New Folder".
Input	Name for new folder.
Expected Output	New Folder with given name will create in the FTP Virtual Drive Explorer.
Actual Output	New Folder with given name has created in the FTP Virtual Drive Explorer.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being attained.

Req. No.	8
Date of Test	30/07/2004
Program	Ftpexplorer.exe
Description	Disconnecting a connected server/site is taken place by selecting "Remove Server" submenu of popup menu. This will disconnect the connected server/site.
Input	Selection of popup menu item "Remove Server" from the FTP Virtual Drive Explorer.
Expected Output	Connected Server will disconnect from the FTP Virtual Drive Explorer.
Actual Output	Connected Server has disconnected from the FTP Virtual Drive Explorer.
Test Conductor	Muhammad Arif

Result of Test: Test showed that desired result is being attained.

References

Web Bibliography:

1. http://www.Trolltech.com
2. http://www.google.com
3. http://www.yahoo.com

Bibliography:

1. **Larman, Craig** "Applying UML and Patterns – An Introduction To Object Oriented Analysis And Design", Prentice Hall PTR.2000.
2. **Somerville, Ian** (Lancaster University) (1998), " Software Engineering", 5th Edition. Addison Wesley Publishing Company.
3. **Pressman R. S.,** "Software Engineering - A Practitioner's Approach", 5th Edition. McGraw Hill Inc.
4. **Oestereich B.**, "Developing Software with UML- Object oriented analysis and design in practice", Addison Wesley Publishing Company Inch.