P2P WORKSPACE

SHEIK AZHAR BUX B. SHEIK AZMI BUX

WEK010256

SUPERVISOR: MR. LIEW CHEE SUN
MODERATOR: MR. PHANG KEAT KEONG

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY MALAYA
KUALA LUMPUR
2004

ABSTRACT

P2P Workspace is a workspace where users can communicate with each other while doing several task in the workspace with a Peer-to-Peer (P2P) technology or communication.

P2P Workspace is designed mainly to fulfill the requirements on the user needs of a workspace application. The workspace is designed to curb certain situation like to gather all the information for the users in a workspace, to ease people task, to allow ad-hoc connectivity as well as because of the increasing of decentralized individuals or group.

The workspace will be for the use of all users who need the services such as file sharing, chatting, shared whiteboard, shared calendar and auto discovery. The implementation of P2P technology is due to the fact that P2P has given a lot of advantages to users to control and manage the use of files and data across networks.

ACKNOWLEDGEMENTS

In the name of ALLAH, I would like to take this opportunity to thank my supervisor Mr. Liew Chee Sun for his willingness to share his knowledge and guide me throughout all aspects in developing this project.

This appreciation is also dedicated to the moderator of this project, Mr. Phang Keat Keong for all his comments which has motivated me in producing a better project.

Grateful appreciation is also extended to my family and friends who have contributed directly or not in producing this project.

Finally I would like to thank my friend Raja Ummi Kalthum and a special appreciation to Mohd Hafiz b. Mohd Yusof for sharing all the knowledge, support and cooperation during all this torrid time.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION

Distributed computing has played a vital role in lives of millions. Especially with the advent of the Internet in the 90's the involvement has taken a new height with millions adopting this technology. Distributed computing helped more people to get involved in computing especially for business purposes, and Internet is for the masses.

Peer-to-Peer (P2P) computing has given a lot of advantages to users to control and manage the use of files across networks. Peer-to-Peer (P2P) computing has gained much attention in recent times, especially with the popularity of Napster, Gnutella, Morpheus and many more.

A Peer-to-Peer (P2P) application needs a workspace to operate. A workspace is a space or screen that presents or shared all the information needed by a user or group.

Peer-to-Peer (P2P) is seen as a revolution after technologies such as the Internet and the World Wide Web (WWW). The potential of Peer-to-peer (P2P) computing can get much more exciting than what it is today, the possibilities are still to be explored.

1.1 PROJECT MOTIVATION

As P2P workspace becomes more mature, its future infrastructures will improve. There will be increased interoperability, more connections to the Internet world, and more robust software and hardware. Nevertheless, some inherent problems will remain. A P2P workspace will remain an important approach for the following reasons:

- Scalability will always be a problem at certain levels (network, system, and application), especially with global connectivity. It will be hard to predict and guarantee all service level agreements. P2P can contribute to each area.
- Certain parts of the world will not be covered by sufficient connectivity, requiring ad-hoc, decentralized groups to be formed. P2P is a well-suited alternative when there is a lack of infrastructure.
- Certain configurations of systems and applications will inherently be P2P and will lend themselves to P2P solutions.
- To gather all the information and people, which are needed to create new knowledge, into the workspace.
- Bring together the set of documents and people to carry out a task and then supports them with a variety of services in a workspace.

1.2 PROJECT OBJECTIVES

1. To develop a useful and convenient P2P workspace which consists of several features such as file sharing, chatting, and shared whiteboard.
2. Reduces the overhead and maintenance cost for server.
3. To enhance the sharing of information among each user, which will make the information sharing easier and efficient.
4. To let the users interact or do discussion in a virtual environment.
5. To create a user-friendly P2P workspace.
6. To improve the scalability of data transmission over the network.

1.3 PROJECT SCOPE

This project will let the users to have similar capabilities among each other where there is no user level or administrator in the workspace. The project scope is based on P2P workspace that provides features such as file sharing, chatting, whiteboard, and auto discovery.

Below are the project scopes:

- **File Sharing** – A file can be download at other locations by users. Users will be able to download and upload files to another user and others. Users also are capable to do a search based on the type of files, where only the type of files being search will be shown.
- **Chatting** – Users can chat with each other in a chat room or can send messages to a user where other user will not be able to read the message. A nickname will be required for each user to enable the user to chat.
- **Shared Whiteboard** – The shared whiteboard allows two or more people to view and draw on a shared drawing surface even when they're at a distance. Shared whiteboards are designed for informal conversation and more sophisticated drawing tasks, such as collaborative graphic design.
- **Auto Discovery** – Each of the nodes that have the P2P workspace will automatically discover by other user when they login and appear.

1.4 EXPECTED OUTCOME

The P2P workspace will let the users to share their resources and information in an easier way. Besides that, the P2P workspace also will reduce the overhead and dependencies of the server. Lastly, the P2P workspace should improve the scalability of data transmission over the network.

1.5 PROJECT SCHEDULE

This project is developed based on the traditional system development life cycle SDLC. This is to maintain all activities to be properly desigend and well monitored. The main activities conducted throughout the project development are as listed below.

i. Preliminary Investigation

ii. Literature Search and Review

iii. System Analysis

iv. System Design

v. Development

vi. Testing and Maintenance

vii. Implementation

viii. Documentation

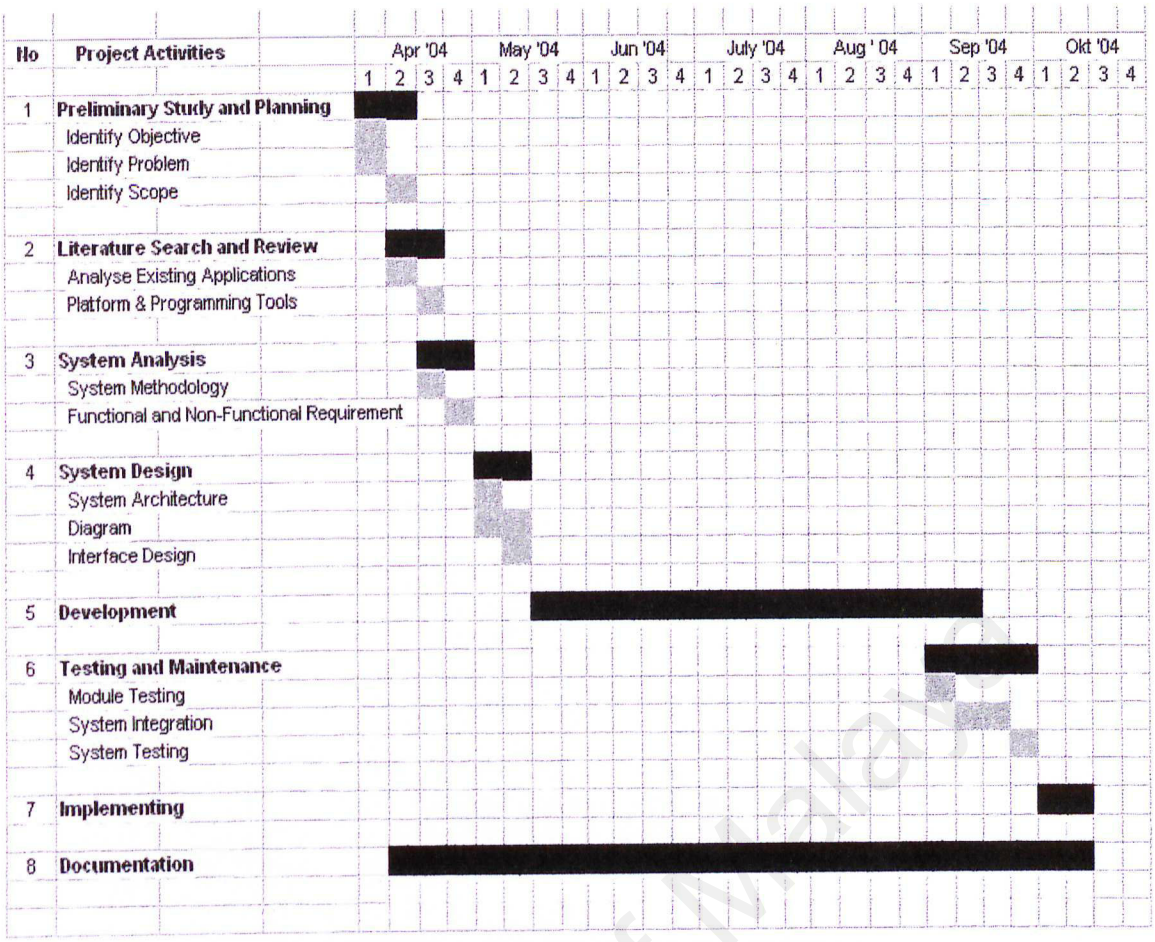


Figure 1.1: The Gantt chart shows the schedule of this project.

CHAPTER 2

LITERATURE REVIEW

2.0 PEER-TO-PEER (P2P)

P2P can be defined as the coordinated use of geographically distributed resources in absence of central control, based on direct exchanges of information. In other words, P2P can be described as a communications model in which each party has the same capabilities and either party can initiate a communication session.P2P communications is implemented by giving each communication node both server and client capabilities.

In a simple way, P2P can be described as the sharing of computer resources by direct exchange between systems. These resources include the exchange of information, processing cycles, cache storage, and disk storage for files. The clients will take their advantage of benefit through the entire enterprise by using a system that uses desktop computing and networking connectivity.

Clients who want to communicate directly with each other have to used the P2P network. They can play the role of the server, hence the load on servers in the traditional way has reduced.

P2P is a technology that involves individuals on a network. Some of the popular examples are for file sharing, popularized by Napster and chatting by MSN.

2.1 GOALS OF P2P

As with any computing system, the goal of P2P systems is to support applications that satisfy the needs of users. Selecting a P2P approach is often driven by one or more of the following goals:

- **Improved reliability** - With the lack of central authority for autonomous peers, improving application reliability is an important goal. As a result, algorithm innovation in the area of resource discovery and search has been a clear area of research, resulting in new algorithms for existing applications.
- **Cost sharing/reduction** - Centralized systems that serve many clients typically bear the majority of the cost of the system. When that main cost becomes too large, a P2P architecture can help spread the cost over all the peers.
- **Increased autonomy** - Users prefer that all data and work on their behalf be performed locally. P2P systems support this level of autonomy simply because they require that the local node do work on behalf of its user.
- **Dynamism** - P2P applications assume that the computing environment is highly dynamic. Resources, such as compute nodes, will be entering and leaving the system continuously.
- **Enabling ad-hoc communication and collaboration** - Related to dynamism is the notion of supporting ad-hoc environments. By ad-hoc, we mean environments where users come and go based perhaps on their current physical location or their current interests.

2.2 CHARACTERISTICS OF P2P

There are some characteristics that make P2P applications different from other applications, which are:

- The location of a file is not known by its retriever, not even after the file is retrieved. So there is no concept of a hooking to a server to get some services done.
- Files move freely among applications, they can show up on some applications and disappear unexpectedly. Large numbers of people request content from a global system rather than from a particular host.
- Have an operational computer of server quality, since each node acts as a client and a server.
- An addressing system which is independent of the DNS.

2.3 MAJOR FORMS OF P2P

2.3.1 Pure P2P

Nodes on this form are peers, for example users can act as clients and servers and have the same capability as its neighbors. It has no central servers. It has every node as a peer and has no central router. Further there are two routing structures, one which is a distributed catalogue and the other direct messaging. There is equality among nodes. Example: Gnutella, Freenet.

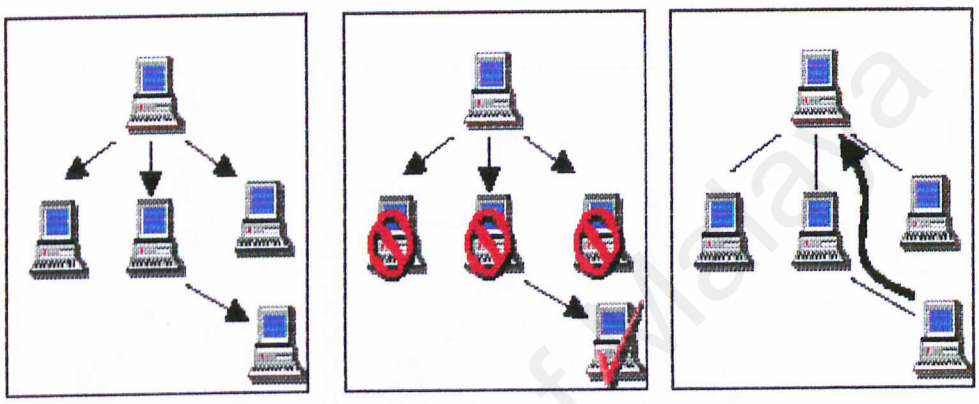


Figure 2.1: The connections of pure P2P. Adapted from http://wiki.cs.uiuc.edu/cs427.

2.3.2 Hybrid P2P

The central server is responsible for maintaining a registry of shared information and responding to queries for that information. The peers are responsible for hosting the information, communicating what is to be shared to the central server, and downloading it to other peers upon request. This is centralized but not in the conventional client server sense. Route terminals are used to hold catalogues of addresses.

Example: Napster

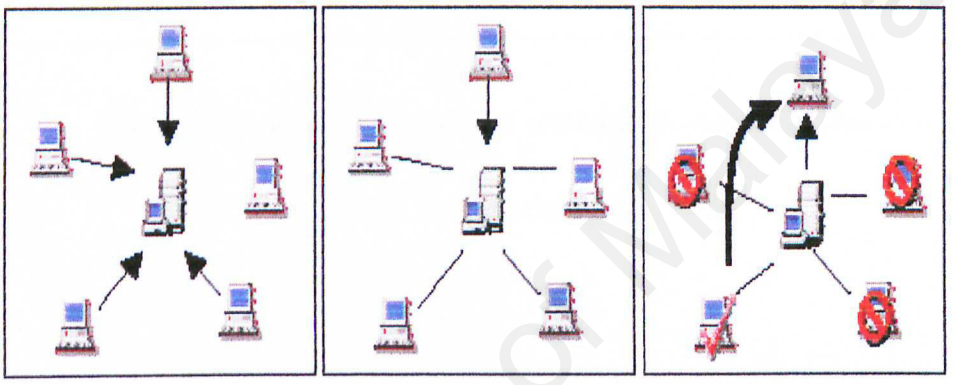


Figure 2.2: The connections of hybrid P2P. Adapted from http://wiki.cs.uiuc.edu/cs427

2.3.3 Mixed P2P

The Pure and Hybrid forms are extremes, the Mixed is a middle ground that involved the server as well as gives the peers adequate independence.

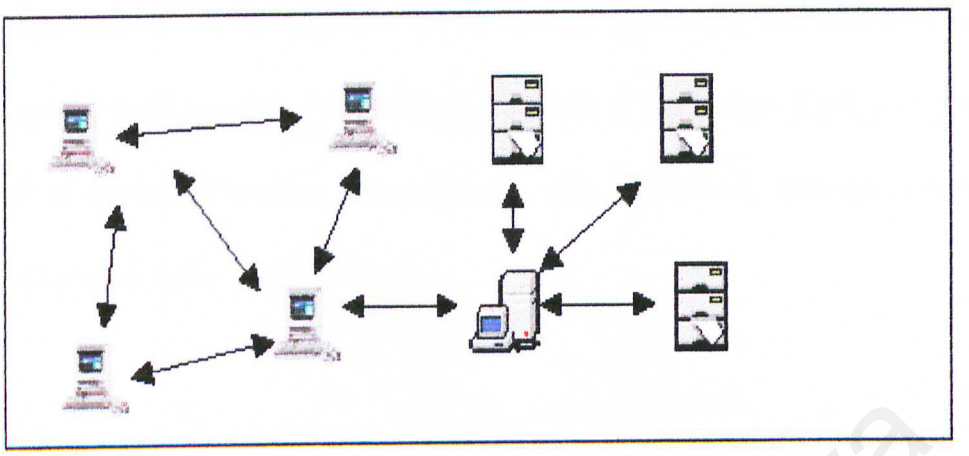


Figure 2.3: The connections of mixed P2P. Adapted from http://wiki.cs.uiuc.edu/cs427

2.4 ARCHITECTURE OF P2P

P2P is a type of network in which each workstation has equivalent capabilities and responsibilities. The development of P2P systems is not easy. The system is built on a set of architectural principles that cater to the needs of the system. Each peer provides a core set of services. It handles communication with the network, manages information about the location, status, and privileges.

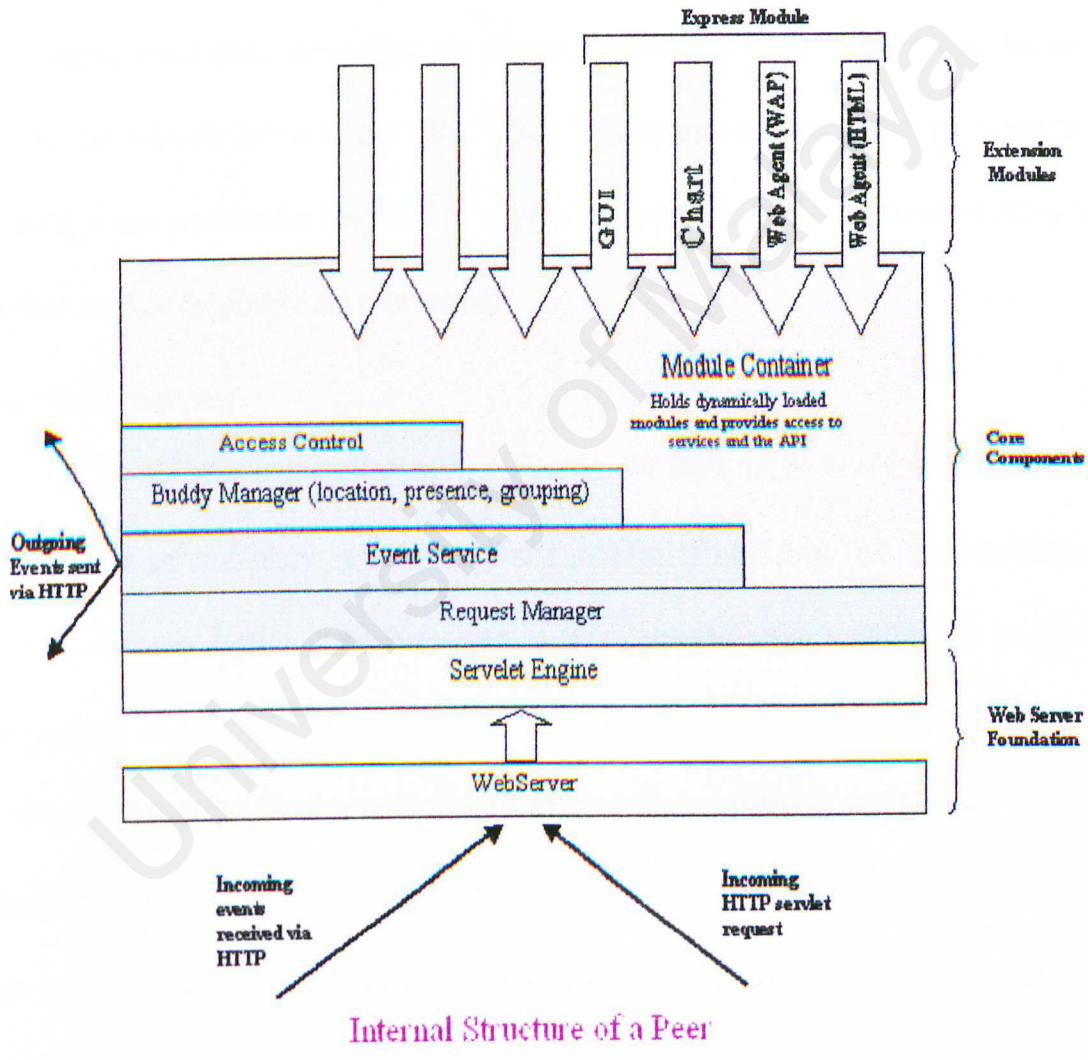


Figure 2.4: The architecture of an individual peer. Adapted from http://wiki.cs.uiuc.edu/cs427.

2.4.1 Building on the network architecture

The system is based on top of a Web server and servlet engine that trigger functionality based on incoming HTTP requests. The network infrastructure may vary by platform and provides the binding to the Web services available on that platform. It provides simple parsing of the HTTP requests, passing the content on to the platform independent request manager.

2.4.2 Request Manager

Request manager receives the HTTP request from the server and, depending on the content, invokes services within the peer to handle the request. Loaded modules may register for requests based on the URL. When the request manager passes a request to a module, it can provide the response (if any) to be returned, or pass the request along to another module for additional processing.

2.4.3 Event Service

Events that are passed over HTTP carry the bulk of communication between peers. Event service receives the events and invokes the services that have registered interest in those events. Also, the event service provides the mechanisms to allow components to send events to other peers. To enhance the efficiency and utility of the event processing, the event service also provides other capabilities. Within each event is information about its creation: the address of the creator, a local timestamp, a count indicating the local ordering of events since that peer was started, and a globally unique identifier string. If the event is in response to another it may also include the identifier of the event to which it is responding.

2.4.4 Buddy Manager

It is a means of identifying individual users and controlling the access relationships between peers. "Buddies" are known to many users from other applications such as instant-messaging tools, and provide an appealing and flexible mechanism for defining such things as who may access resources and about whom the user would like information. Each device has a location on the network that might be consistent or might change frequently. The peer manages this issue by tracking users locations and making them available if modules require them. This is done in cooperation with directory services that help locate connected users.

2.4.5 Access Controller

Buddies, devices and gropus are used as mechanisms to control access to the peers resources. Information in the incoming event or HTTP request serves to identify users as being permitted defined levels of access, thus the user may permit or deny access to services. Further, once another user requesting a service is permitted access by these primary controls, the extension modules may determine, based on information from the controller, a more refined level of access control to their services.

2.4.6 Module Container

Dynamically loaded extension modules that provide new functionality for specific applications extend the peer's core functionality. The modules could be included dynamically to allow custom configuration of peers, in most cases even while the peer is running. These extension modules are provided access to the core APIs, once loaded. They may be mapped to handle incoming HTTP requests, register for events, send events to other peers, and access information about buddies and access controls.

2.5 WORKSPACE

A workspace is a space or environment that presents all the information needed by a team or group. A workspace can be customized to team requirements. The environment should enable the users to access common information, and to share and exchange comments about the information. The workspace or environment also allows users to analyze the information and create new information.

2.5.1 Workspace Components

A workspace should contain several elements:

- Any number of documents and background information. Any number of roles.
- Any number of discussions.
- Any number of people (or participants) assigned to each role.
- Any number of actions (or things to do) assigned to each role.
- Awareness information about the workspace.

2.5.2 What Happens In The Workspace?

The owner of the workspace is the one who created the environment. The owner has the capabilities to make and change the setting of the workspace. The owner can use all the menu commands that are available for a workspace to create new roles and empower them with some of these commands. The owner can compose the workspace by using the commands through the application development interface, which can be used to:

- Add any number of documents to the workspace.
- Add any number of roles to the workspace.
- Assign any number of people (or participants) to each role.
- Assign any number of actions (or things to do) to each role.
- Create any number of services to support discussion between participants.
- Create new workspaces that focus on specific goals.
- Invite people to participate in these workspaces.
- Allow people to gradually release information to others.
- Support discussion people in the workspaces.
- Create new related workspaces to address any new issues raised in an existing workspace.

2.6 OVERVIEW OF EXISTING APPLICATIONS

2.6.1 Groove

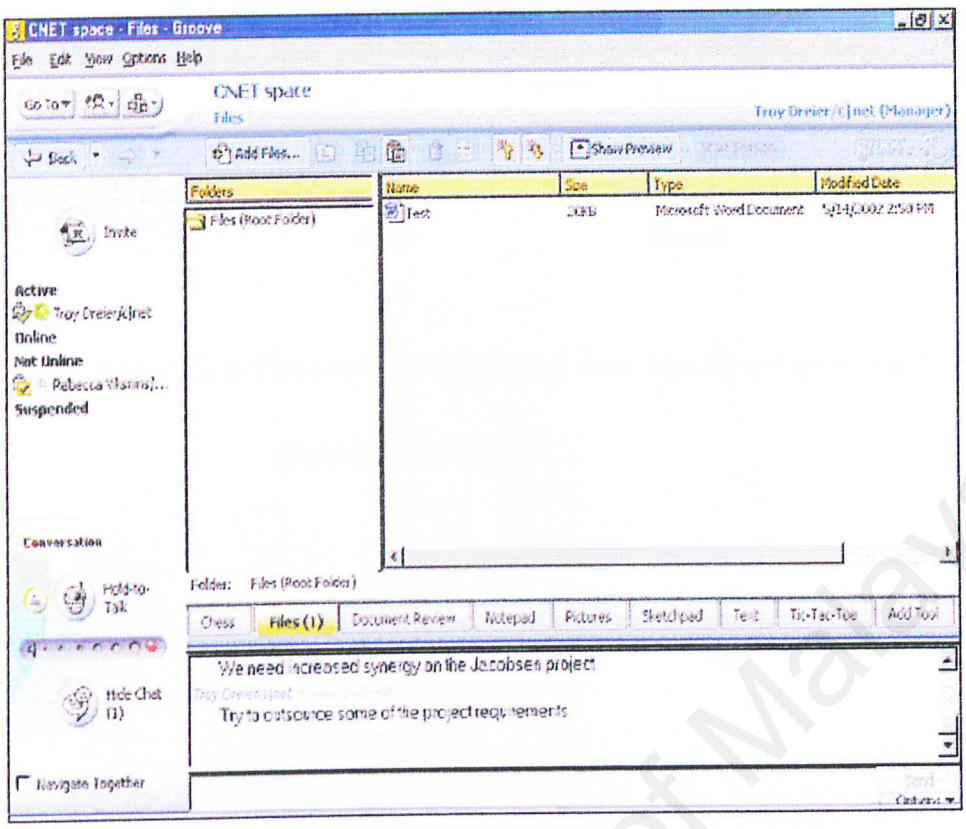


Figure 2.5: Groove workspace interface.

Application Description:

Groove is a collaborative P2P system. It is mainly targeted to Internet and intranet users, although it can also be used on mobile devices, such as PDAs, mobile phones, and tablets. It is intended to enable communication, content sharing, and tools for joint activities. The Groove's main goal, which was P2P by design, was to allow users to communicate directly with other users without relying on a server. Other important goals include security, privacy and flexibility.

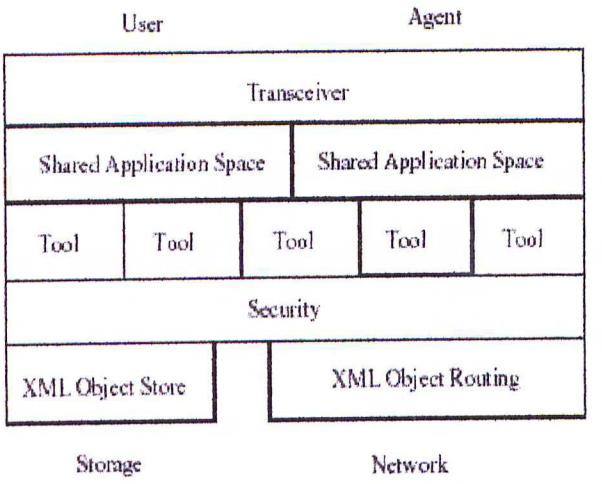


Figure 2.6: Groove layers. Adapted from http://www.groove.net/pdf/back grounder-product.pdf.

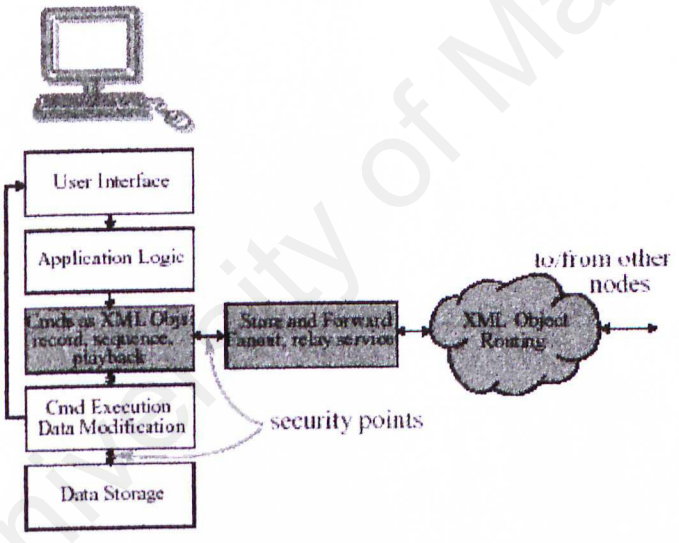


Figure 2.7: Groove application structure. Adapted from http:// www.groove.net/developers/presentations/architecture.exe.

Advantages:

Includes text and voice based instant messaging plus a wide assortment of tools and toolsets for sharing content of all kinds, working together, and managing projects and meetings. Users choose how they communicate and work, selecting tools that match their interaction style and make the most sense for the tasks at hand.

2.6.2 LimeWire

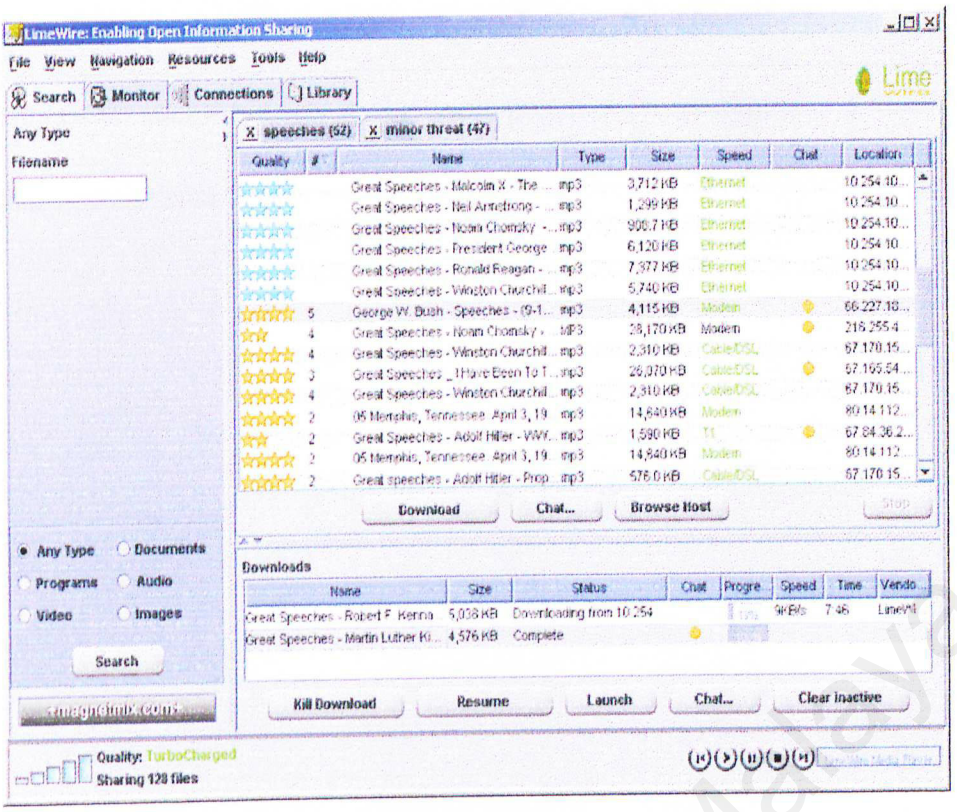


Figure 2.8: LimeWire interface.

Application Description:

LimeWire is a software tool that enables peer-to-peer file-sharing on the Gnutella network. Similar to the once-popular Napster service, it enables the sharing, searching, and downloading of MP3 files. LimeWire is capable of multiple searches, available in several different languages, and is most famous for its ease of use and cross compatibility.

Advantages:

Ease of use, just install, run, and search. Besides that, it has the ability to search by artist, title, genre, or other information. Other features are integrated file library with built-in audio player and even browse host feature and works through firewalls.

Disadvantages:

The security mechanism is inefficient, the Gnutella network has no built-in options for caching. There is also some potential for spamming or a virus by returning a fake search result.

2.6.3 BearShare

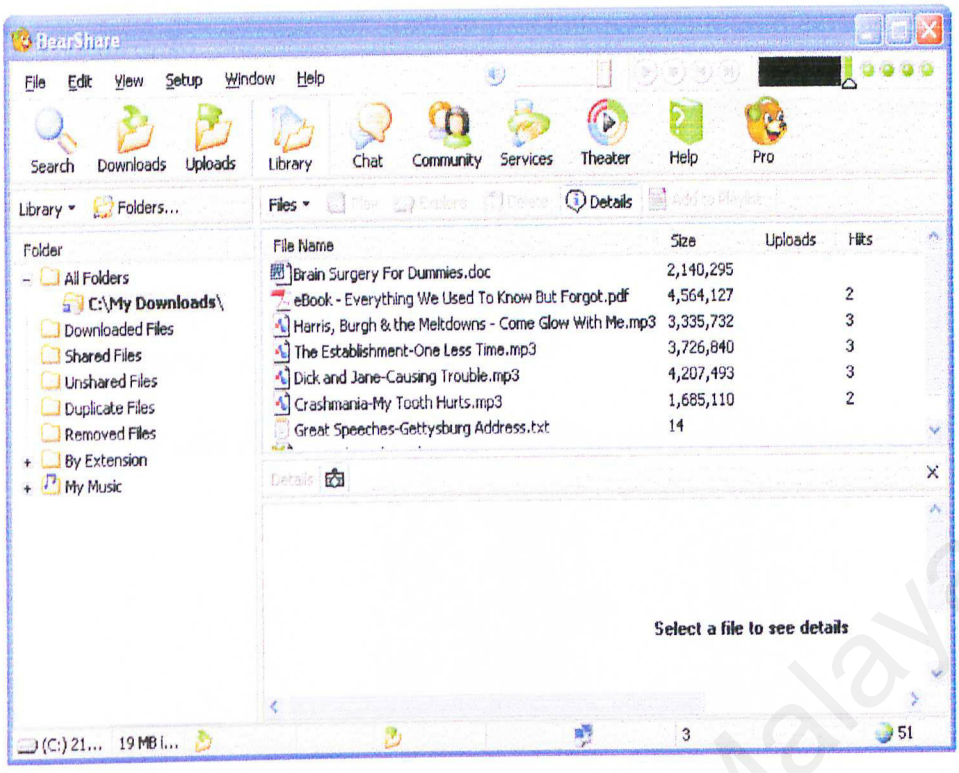


Figure 2.9: BearShare interface.

Application Description:

BearShare is an attractive file sharing application around. Running on the Gnutella network, it returns results quickly, and it now comes with a built-in chat client. Once BearShare is installed, a setup wizard lets user specify where to save downloads and what folders to share with the public. There is automatic resume feature to assures the completion of all requested downloads. User can access and view friend's shared files and even chatting while downloading.

Advantages:

Download requested files from multiple users simultaneously. There is also advanced search for all types of media files. There is also a media player with play lists and full screen preview mode.

Disadvantages:

Unreliable in terms of files downloading and poor download success rate.

2.7 EXISTING PROTOCOLS

2.7.1 Gnutella

Gnutella is a file sharing protocol. Applications that implement the Gnutella protocol allow users to search for and download files from other users connected to the Internet.

History: Gnutella file sharing technology was introduced in March of 2000 by two employees of AOL's Nullsoft division. Touted as an open source program with functionality similar to that of Napster, the Gnutella servant program was taken offline the following day because of a possible threat to Warner Music and EMI. AOL was rumored to be in the midst of merger talks with the record companies at that time. However, the open source programs remained online long enough for eager hackers to discover the Gnutella protocol and produce a series of clones to communicate using the Gnutella communication protocol. Soon after, versions of the original Gnutella servant were communicating with Gnutella clones to search and trade files over the Gnutella Network.

Goals: The goal of Gnutella is to provide a purely distributed file sharing solution. Users can run software that implements the Gnutella protocol to share files and search for new files. The decentralized nature of Gnutella provides a level of anonymity for users, but also introduces a degree of uncertainty.

Design: Gnutella is not a system or a piece of software. Gnutella is the communication protocol used to search for and share files among users. A user must first know the IP address of another Gnutella node in the network. This can be discovered by going to a well known Web site where a number of Gnutella users are posted. When a user wishes to find a file, the user issues a query for the file to the Gnutella users about

which it knows. Those users may or may not respond with results, and will forward the query request to any other Gnutella nodes they know about. A query contains a Time-To-Live (TTL) field and will be forwarded until the TTL has been reached.

2.7.2 JXTA

The vision of the JXTA project is to provide an open, innovative collaboration platform that supports a wide range of distributed computing applications and enables them to run on any device with a digital heartbeat. JXTA provides core functionality in multiple layers, including basic mechanisms and concepts, higher level services that expand the capabilities of the core, and a wide range of applications that demonstrate the broad applicability of the platform.

History: The JXTA project was unveiled by Sun on April 25, 2001 and was intended to be a platform on which to develop a wide range of distributed computing applications. Despite its recent introduction, JXTA has been quite popular. Statistics show that on the week of November 3rd, 2001, JXTA had 122 posts and was used by 6,809 users.

Goals: The goal of JXTA is to provide a "general purpose" network programming and computing infrastructure. JXTA goals are:

- *Interoperability*: by enabling interconnected peers to easily locate each other, participate in community based activities and offer services to each other seamlessly across different P2P systems and different communities.
- *Platform independence*: JXTA is designed to be independent from programming languages (such as C or Java), system platforms (such as Microsoft Windows and UNIX operating systems), and networking platforms (such as TCP/IP or Bluetooth).
- *Ubiquity:* JXTA is designed to be implementable on every device with a digital heartbeat, including appliances, desktop computers, and storage systems.

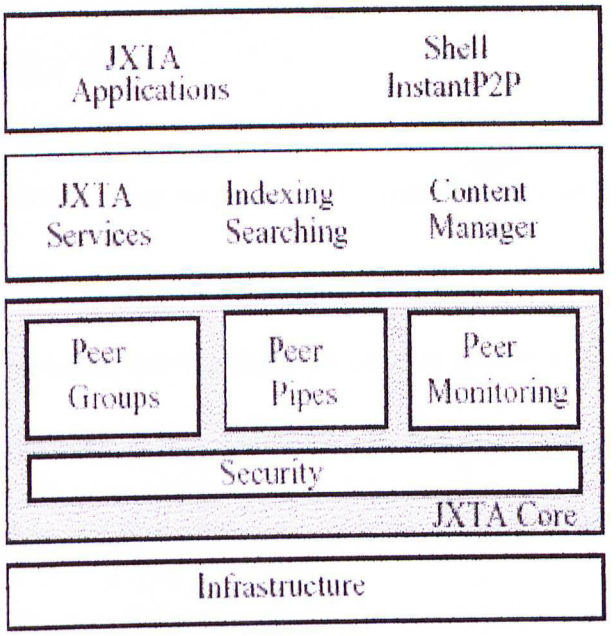


Figure 2.10: JXTA architecture.

Design: It is important to note that the JXTA project is approaching the P2P space from the lower level as they are proposing an entirely new infrastructure with no direct relation to other, existing P2P systems (e.g., Gnutella and Napster). For example, they have built their own distributed search service, called JXTA search.

2.8 THREATS TO SECURITY

Nowadays, security issues regarding P2P computing are of the greatest concern. P2P technologies allow users to retrieve files from other computer users, publish and disseminate data that is stored locally, share processing power, and execute applications across the network. Inherent in these features are threats to personal privacy, system security and the integrity of information.

The first issue is authentication, which is confirming and verifying the identity of persons with whom you interact. The majority of business oriented P2P applications provide mutual authentication in the exchange of public keys. To mutually authenticate two parties using the public/private key system, the first party establishes a connection to the second party and sends their digital certificate from a trusted certificate authority (CA). The second party, after confirming that the digital certificate is valid, responds with a challenge message to be encrypted using the first party's private key. If the second party can decrypt the message using the public key, the user knows that the first party is who they say they are. This process must then happen in reverse. Without such measures, you may not be able to verify with whom you are interacting in a P2P network.

The second issue is the importance of having authorization which is controlling access and security at the local or node level. Because most P2P applications are open systems that require you to leave your computer on and connected to the network, there is the potential for unauthorized access and all the attendant issues that that may introduce, such as hacking and virus attacks. The threat is real considering that most P2P

applications provide some method for tunneling through firewalls and bypassing the perimeter defenses. Once a system is open to direct communication, there is the risk of exposing local content and confidential or private information, such as cookies, to external parties.

The last issue is users of P2P technologies must be confident that their data is not being modified intentionally or accidentally when being sent across the network. Data integrity is especially critical in business transactions. For example, instant messaging applications are rarely protected and messages can be easily intercepted.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.0 INTRODUCTION

A methodology includes technology, management, strategic and cost. It is necessary to investigate the needs of a development environment as to provide a better structure of managing resources, a better defined requirement and a well structured of development phase.

The system development methodology is a method to create a system with a series of steps or can be defined as system development life cycle (SDLC). Every system that uses the SDLC must go through the same phase in their life cycle. The phases are:

- Feasibility study
- Analysis and Requirement Specification
- Design
- Implementation
- Maintenance

3.1 PROJECT DEVELOPMENT METHODOLOGY

In developing a system or application, there are a wide variety of process models to represent a particular process. For example, the models that are widely used are the *Waterfall Model, Matrix Model, V-Model, Spiral Model* and much more. All these models vary and its usage is based on a certain requirement in system development.

The choice to choose the most suitable model is influenced by the nature of the problem:

- The degree of the system structure.
- Familiarity with the technology.
- Project size.

The proposed project will apply the Waterfall with Prototyping Model which considered the process as progressing through a series of stages from requirements analysis through specification, design, coding, testing to documentation and maintenance.

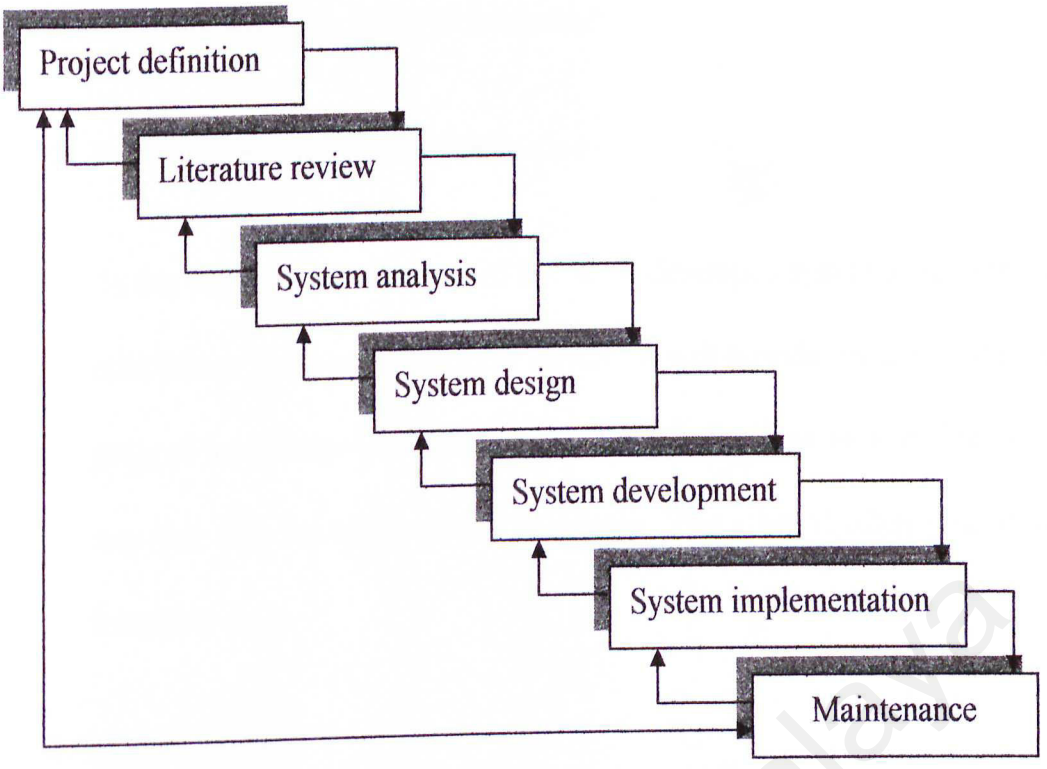


Figure 3.1: Waterfall with Prototyping Model.

3.1.1 Why Waterfall with Prototyping Model?

In Waterfall model, each stage was completed before the next was started. It consists of different stages, which are processed in a linear fashion. Compared to other system development models, it is more rigid and better manageable. The waterfall model is an important model which is the basis of many other models. Prototypes are excellent tools for improving the communications between software developers and end users as well as for providing managers with information needed to conduct cost, schedule, and functionality tradeoff analyses. Prototypes also are very useful when the developers are unfamiliar with the development environment.

There are usually five basis stages in this model:

- **Requirements analysis and definition**.

  In this stage the requirements of the "to be developed system or application" are established. These are usually the services it will provide, its constraints and the goals of the system. Once these are established they have to be defined in such a way that they are usable in the next stage. This stage is often precluded by a feasibility study.

- **System design.**

  In this stage the established requirements, flowing from the first stage, are identified as software or hardware requirements. The software requirements are then translated in such a way that they can be readily transformed into computer programs or applications.

- **Implementation and module testing.**

  This is the stage where the computer programs are created. Each program is called a module, and module testing is the verification that every module meets its specification.

- **System testing.**

  All the modules are combined and now the whole is tested. When the combined modules are successfully tested the system is finished.

❖ **Operation and maintenance.**

Most system or applications include this stage of the development. It involves correcting errors that have gone undetected before, improvement and other forms of support.

3.2 INFORMATION GATHERING METHODS

There are several methods that can be applied upon gathering the desired information. The methods implied may be conducted in a formal or informal manner. In order to gain the appropriate information for the application proposed, the following methods were applied.

3.2.1 Observations

Through the observations, the needs, requirements and problems of the project are being analyze and understand. The observation has also lead to experimenting the existing system such as the groove workspace, bearshare and other applications.

3.2.2 Internet Surfing

Another method to gain more knowledge with unlimited time restrictions and very flexible is by surfing the internet. The internet offers a very huge variety of information on workspace and P2P technologies. The knowledge gained from surfing the internet is not only limited to local information but the information available is worldwide. There are no geographic boundaries in gaining knowledge using the internet either knowledge about P2P, workspace or software and hardware needed to develop this project.

3.2.3 Other Sources

The information gathered are also from various sources apart from what is mentioned above. These sources are such as journals, newspaper articles and the traditional method which is from books. Although the information gained from these

sources are very time consuming but the knowledge offered are tremendously useful upon the development phase of this project.

3.3 THE LIMITATIONS IN DEVELOPING THE PROJECT

- **Time** – Much time is needed to read, study and analyze papers in developing the P2P workspace which based on P2P technology. Time consuming is important in the need to finish this project on time.
- **Resources** – Resources such as the information of the latest technology and source code (coding) are hard to obtain. The reference and text book are expensive, so the last choice of resources is the internet.
- **Inexperienced Skill** – P2P technology is a quite new technology. Therefore, to build one P2P workspace needs a personal skill to develop a new system.

CHAPTER 4

SYSTEM ANALYSIS

4.0 INTRODUCTION

System analysis plays an important role in order to meet the requirements of the system. The system definitely can't be accomplished without the existence of these needed requirements. The requirements might have to change along the development process depending on the unexpected situations that were not considered earlier. The requirements are divided into several categories which are functional and non-functional requirements. Besides that, the software and hardware requirements have been identified to ease the development process.

4.1 FUNCTIONAL REQUIREMENTS

Functional requirement describes an interaction between the system and its environment. The system can be partitioned into related pieces, each of which describes some way in which the system will function.

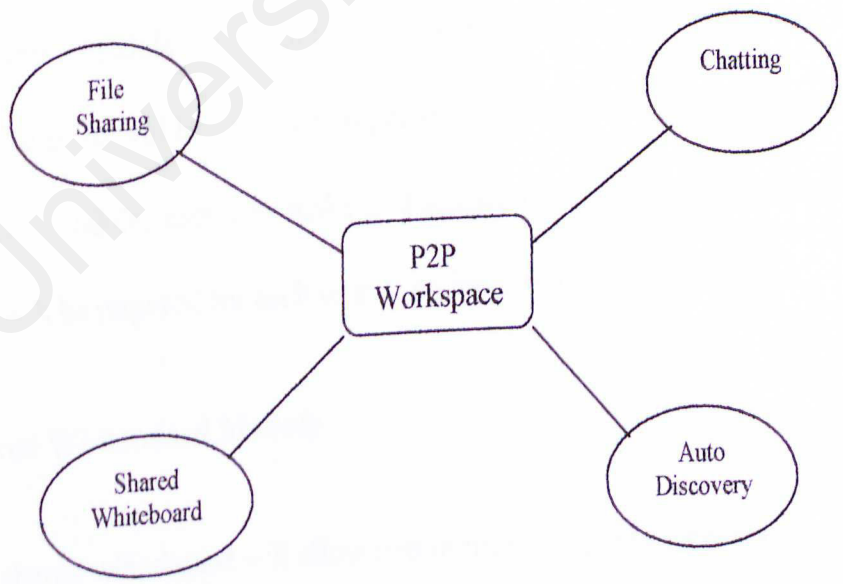


Figure 4.1: The main modules of P2P workspace.

4.1.1 File Sharing Module

File sharing is a module which consists of files searching and files downloading. File searching main function is to provide user with easy ways of locating files in an environment (example: LAN). File searching will search for the file in the entire P2P nodes file sharing folder. Only the files that are located in the file sharing folder of each user will be return as reply. If more than one node has the same file that being requested, it will be listed out also with the details of the file such as files name, files size and download time.

File downloading is important because information sharing can be slow without an efficient downloads. The resources of the network will be wasted if information sharing is not optimize for the utilizing the network bandwidth.

4.1.2 Chatting Module

Chatting module consist only one aspect, a chat room. A chat room is a space or screen where there will be at least two person chatting or sending messages at the same time. In this module, user can make a discussion or exchange ideas and opinion. A nickname will be required for each user as an identifier.

4.1.3 Shared Whiteboard Module

A shared whiteboard will allow two or more people to view and draw on a shared drawing surface even when they're at a distance. Shared whiteboards may indicate where each person is drawing or pointing by showing pointers, which are color-coded or labeled to identify each person. The whiteboard should be able to draw line, freehand,

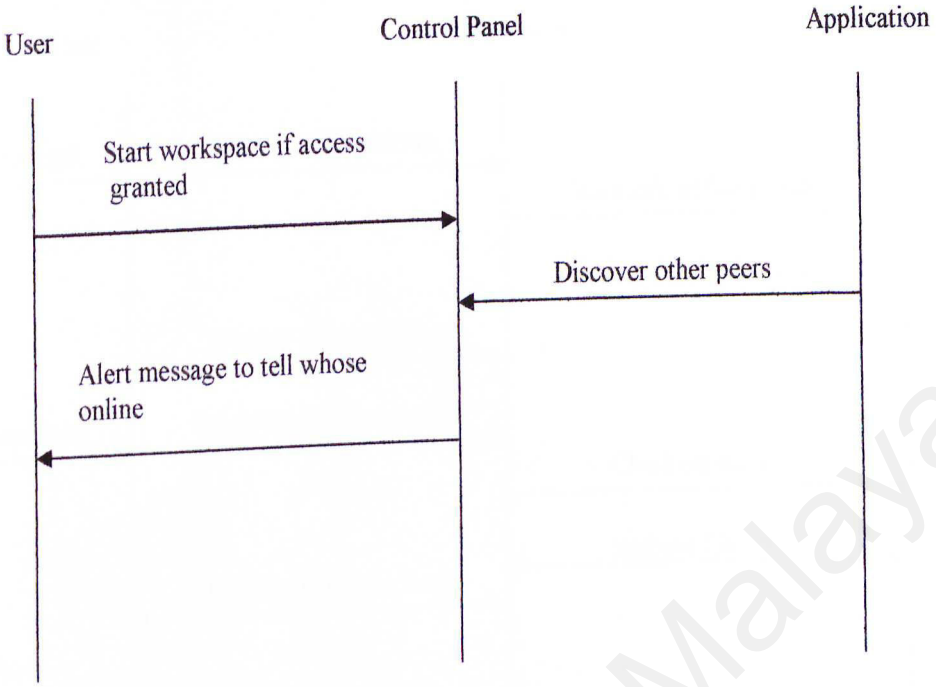
oval, rectangle and other shapes with un-do and re-do process. The background and the foreground color can be set up.

4.1.4 Auto Discovery Module

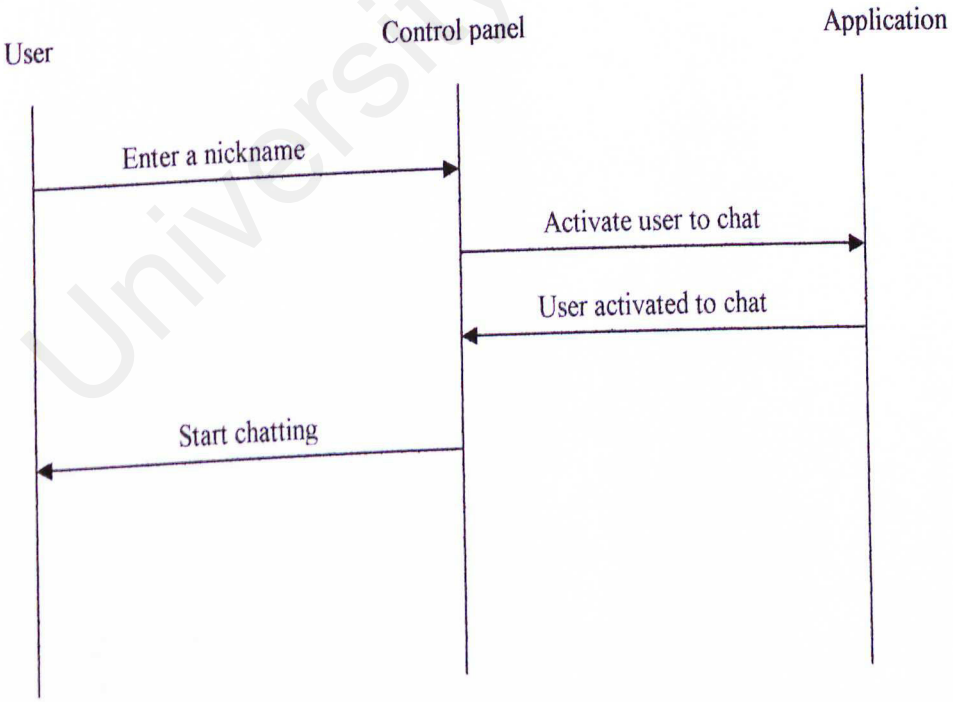
Auto discovery is where a user discovers other users when they login into the workspace. The workspace should alert the user whenever their friends or other users in their list login.

4.2 STATE REPRESENTATION

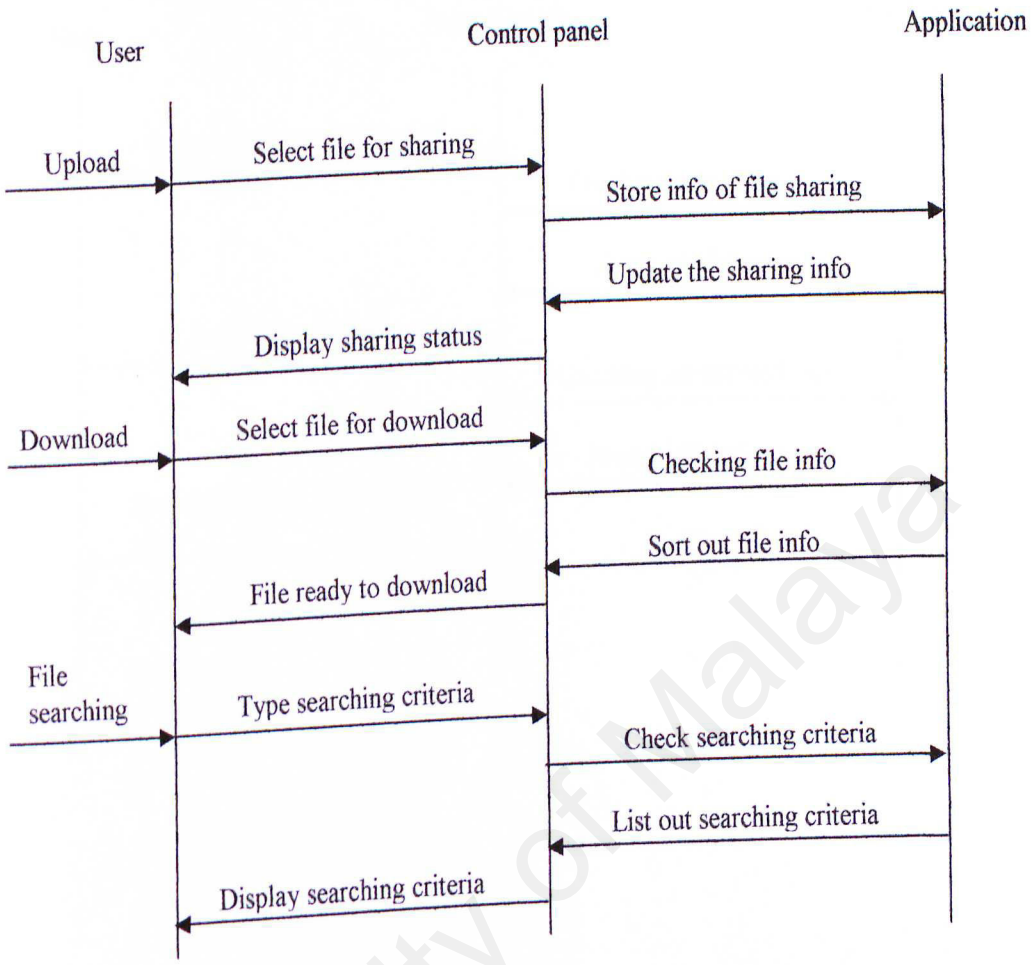
Auto Discovery



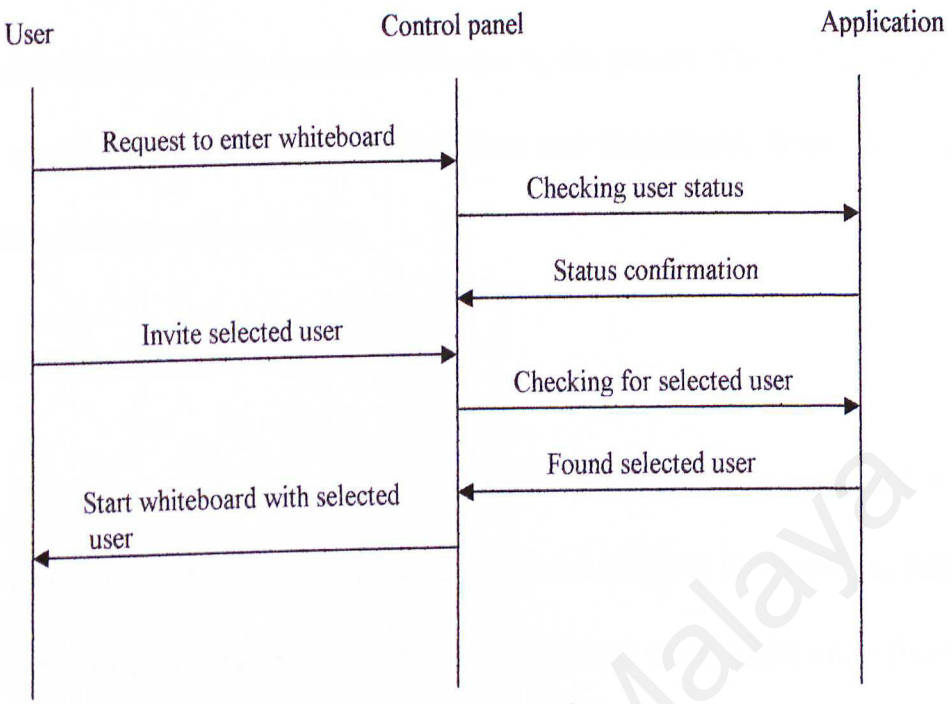
Chatting



File Sharing



Whiteboard



4.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are those requirements which are defined indirectly concerned with the specific functions delivered by the project. The P2P workspace must ensure certain P2P application qualities such as user friendliness, flexibility, reliability, robustness, security and modularity.

4.3.1 User Friendliness

User interface design creates an effective communication medium between a human and a computer. It is very important to make sure that the interfaces fulfill user friendliness aspects so that it would not cause trouble to users. There are a few rules in developing user friendly interface:

- Place the user in control.
- Reduce the user's memory load.
- Make the interface consistent.

4.3.2 Flexibility

The workspace must posses the capability to take advantages of new technology and resources and can be implemented in changing business environments. The workspace must be able to cater for all types of information.

4.3.3 Reliability

Reliability is the extent to which a system or application can be expected to perform its intended function with required precision. It is related to correct link processing, error recovery and user input validation and recovery. This quality is essential as it indicates how far users will be confident in the implementation of the new computerized system in getting processing done.

4.3.4 Robustness

The modules for P2P workspace will be wholly tested to ensure each module achieve its expectation. The modules will be integrated and system testing will be started after the integration. The workspace should be robust enough to handle expected or unexpected failure. Any errors detected will either be corrected or eliminated. The robustness of the system is achieved after thorough testing.

4.3.5 Security

The workspace should have a security mechanism such as authorization. The user of the workspace should be authenticated before entering the workspace. Authorization is important because it controls the access and security at node level, thus preventing unauthorized access and the entire attendant that may introduce, such as hacking and virus attacks.

4.3.6 Modularity

Modularity is a key factor in designing a system or application. The system should be divided into modules of functions so that it will be easier for testing and maintenance.

4.4 SYSTEM DEVELOPMENT REQUIREMENTS

The system requirements to develop the P2P workspace are divided into two, hardware requirements and software requirements.

Hardware requirements:

- Intel Pentium III 500MHz and above.
- 128MB RAM and above.
- 2GB hard disk with a minimum of 650MB of free space.
- Network interface card

Software requirements:

- Windows ME, 2000 or XP.
- Microsoft Visual Studio.NET
- Microsoft .NET Framework.

4.5 USER REQUIREMENTS

The user requirements to execute or to be able to use the P2P workspace are divided into two, hardware requirements and software requirements.

Hardware requirements:

- Intel Pentium III 500 MHz and above.
- 128MB RAM and above.
- 2GB hard disk with a minimum of 650MB free space.
- Network interface card.

Software requirements:

- Windows ME, 2000 or XP.
- Microsoft .NET Framework.

4.6 DEVELOPMENT PLATFORM

4.6.1 Microsoft .NET Framework

The Microsoft® .NET Framework is an important new component of the Microsoft Windows® family of operating systems. It is the foundation of the next generation of Windows based applications that are easier to build and integrate with other networked systems. The .NET Framework helps software developers and systems administrators (users) more easily build and maintain systems with improvements toward performance, security and reliability.

4.6.2 .NET Framework Benefits

- Assists with the deployment of software to users.
- Facilitates the development of software with improved reliability, scalability, performance and security.
- Helps developers be more productive by:
    1. Making it easier for them to reuse existing code.
    2. Enabling them to more easily integrate components written in any of the more than 20 supported programming languages.
    3. Helping them more easily build software for a wide range of devices using same skills and tools.

4.6.3 Basic Component Of The .NET Framework

The .NET Framework is divided into two parts:

Common Language Runtime: Provides the common services for .NET Framework applications. Programs can be written for the common language runtime in just about every language, including C, C++, C#, J# and Microsoft Visual Basic®, as well as some older languages such as Fortran. The runtime simplifies programming by assisting with many mundane tasks of writing code. These tasks include memory management which can be a big generator of bugs, security management and error handling.

.NET Framework Class Library: The library includes sets of functionality that developers can use to more rapidly extend the capabilities of their own software. The library includes three key components:

- ASP.NET to help build Web applications and services.
- Windows Forms to facilitate smart client user interface development.
- ADO.NET to help connect applications to databases.

4.6.4 Reasons To Use .NET Framework

These are the reasons for developers to start building applications using the .NET Framework:

- **Developer Productivity:** The intuitiveness of the programming model, the amount of code already provided in the class libraries and the amount of work that the .NET Framework handles behind the scenes in areas such as memory management have enabled .NET Framework developers to reap huge productivity gains.
- **Increased Performance:** Compilation and caching techniques have never been faster than with the .NET Framework and its ASP.NET technology. .NET Framework increases the speed on the order of 300 to 500 percent improvements.
- **Improved Reliability:** With advanced ways of monitoring the health of running applications, as well as isolating applications from each other, applications built using the .NET Framework running longer than before.
- **Ease of Deployment:** The .NET Framework makes it easy to deploy, run and manage applications. Application isolation and automatic version control of components can help prevent versioning conflicts. Applications built using the .NET Framework can be deployed to a client (user) simply by copying the application directory to the target machine with no registration is required.

- **Support for More Than 20 Programming Languages:** The .NET Framework supports the integration of over 20 programming languages, enabling developers to choose the right programming language to build the application. All programming languages target a single, extensive and extensible set of class libraries.

4.7 DEVELOPMENT TOOLS

4.7.1 Microsoft Visual Studio.NET

Microsoft Visual Studio.NET helps developer to develop a system quickly and integrate them with other applications. Most developers can leverage existing skills, because the .NET Framework's common language runtime allows developers to develop a system or application using any modern programming language.

4.7.2 Reasons to use Microsoft Visual Studio.NET

These are the reasons to use Microsoft Visual Studio.NET as a development tools:

- Develop for devices – Developers can construct applications for variety of mobile devices, including pocket PCs, Tablet PCs and more.
- Enterprise Instrumentation Framework (EIF) – Visual Studio.NET enables integration of the EIF to easily add run time monitoring capabilities to any application.
- Improved debugger – An enhanced debugger displays more readable and intuitive views of data, helping developers to detect and fix errors in the code before deployment.
- Integrated Development Environment (IDE) – IDE provides improved start up time, performance and reliability.

4.8 PROGRAMMING LANGUAGES

Programming languages are an inherently personal choice, one based on a number of factors. Microsoft provides a platform on which a number of languages can each flourish and enjoy full access to the power and flexibility of the .NET Framework.

4.8.1 Visual C# .NET

C# .NET offers programmers a clean and elegant programming language, familiar to C++ developers, with many of the productivity benefits enjoyed by Visual Basic programmers.

C# Features:

- **Support for all CLR types:** The C# language supports all data types defined in the CLR, enabling programmers to create solutions that take advantage of the .NET managed execution environment.
- **Pass by reference and out parameters:** Using C#, programmers may pass variables by reference to a function and even create out parameters that require initialization prior to the termination of the function in which they are defined.
- **Unsafe code:** C# enables programmers to selectively use pointers and manipulate memory. While "unsafe code" is still managed by the CLR, it does afford the advanced programmer greater control over how their application uses memory.

CHAPTER 5

SYSTEM DESIGN

5.0 INTRODUCTION

This chapter focuses on the project design, architecture and user interface design. Before a complete visual design is created, the knowledge gathered form the early stages in this project is very important and is used to create a full description of the modules and functionality required in order to produce a good product.

In order to conceptualize raw data into a model that is clear and precise, the following descriptions are visualized through diagrams and flow chart according to each module which will be integrated upon producing the end product. The system designed must meet certain standard in order to generate a reliable and precise product.

5.1 ARCHITECTURE DESIGN

Application architecture specifies the technologies to be user implement into system in terms of data, process, interface and how these components interact and communicate across a network.

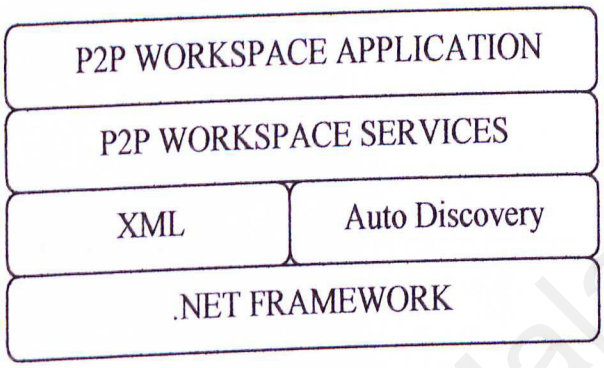


Figure 5.1: P2P Workspace Architecture Layer.

From figure 13, the P2P workspace will be build on a .NET framework as the platform for the workspace. The P2P workspace services is the four modules which is file sharing, chatting, shared calendar and shared whiteboard.

5.2 WORKFLOW DIAGRAM

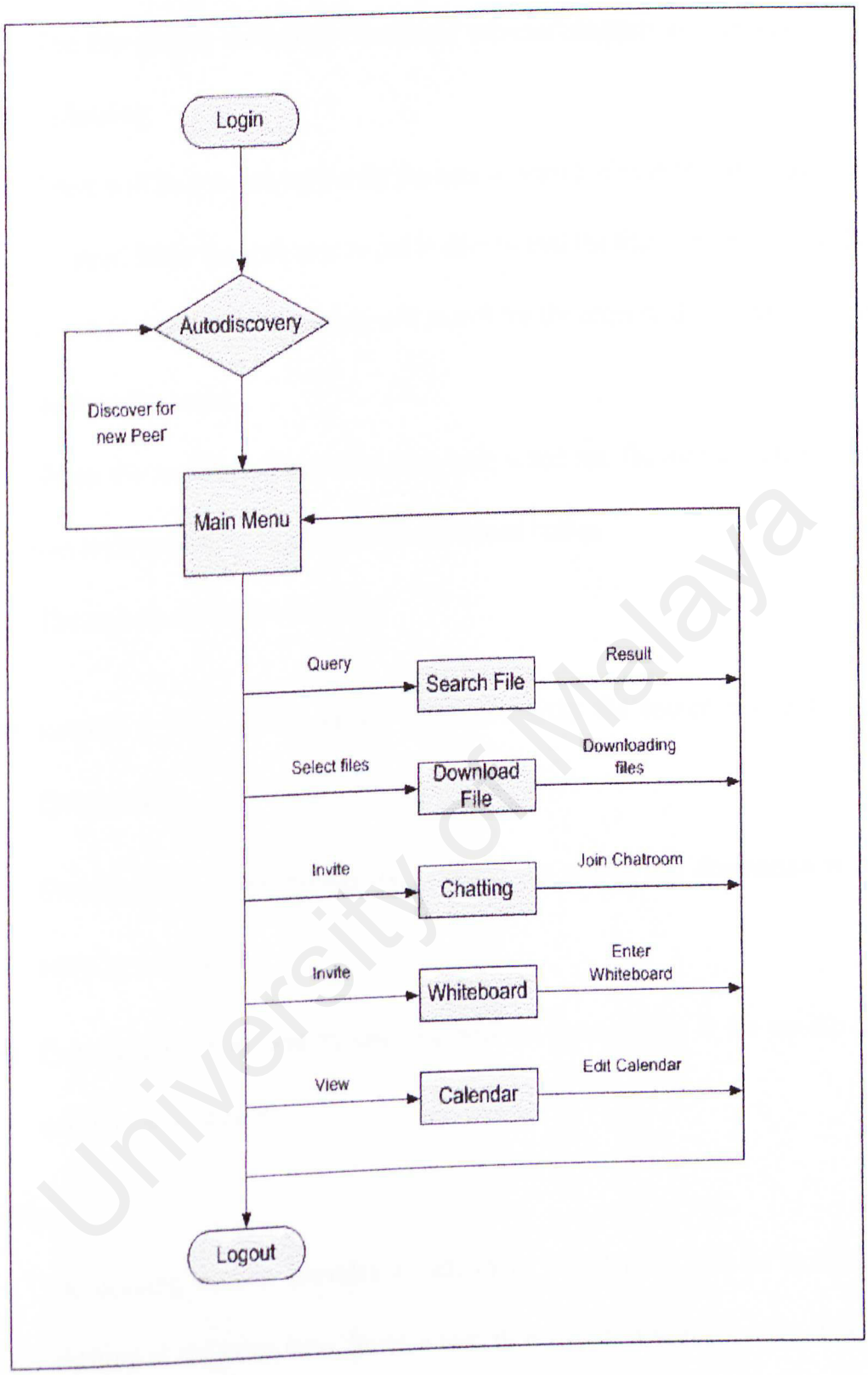


Figure 5.2: P2P Workspace Workflow Diagram.

From figure 14, each module integrated is designed. The workflow diagram is clear and simple in order to cater the usage pattern of the P2P workspace where all the modules execute in their own way.

File sharing:

The file sharing module is consists of sub-module such as files searching and files downloading.

There will be a search button for the user to search files in the workspace. There will be a shared folder for each user to put in files so that the files can be downloaded by other users. The file searching module will search for the requested file and will show it in the interface as a result.

When the requested file or files have been listed out, the user can choose which file or files to download by clicking on the download button.

The aspects of file downloading:

- Request a file – A file should be request from the source before it can be downloaded.
- Synchronizing – Synchronization between the source and destination is being setup by a clock.
- Cancellation – User will be able to cancel the downloading in the middle of the downloading process.

Chatting:

The chatting module provides a chat room where there will be at least two persons chatting at the same time. Invite a user or a friend to start chatting. When the other user is invited, they will join the chat room to chat. Before entering the chat room, a username or nickname will be required for each user.

Shared whiteboard:

The shared whiteboard module provides a space or screen where the users can draw or write something between each other. To be able to start or use the whiteboard, a user should invite another user. Then, the invited user can enter the whiteboard and start using it. There will be a tool box or a drawing box on the side of the whiteboard which can be use for discussion.

Auto Discovery:

Auto discovery is a module where it will tell a user when their friends or other users login into the workspace. An event will be raise if a packet from another machine is received. However, it is not clear at this point how to deal with firewalls, for example, how a peer outside a firewall can discover peers inside the firewall or how two peers inside two firewalls can communicate.

5.3 DATAFLOW DIAGRAM

The dataflow diagram is a tool that depicts the flow of data through a system and the work or processing performed by that system.

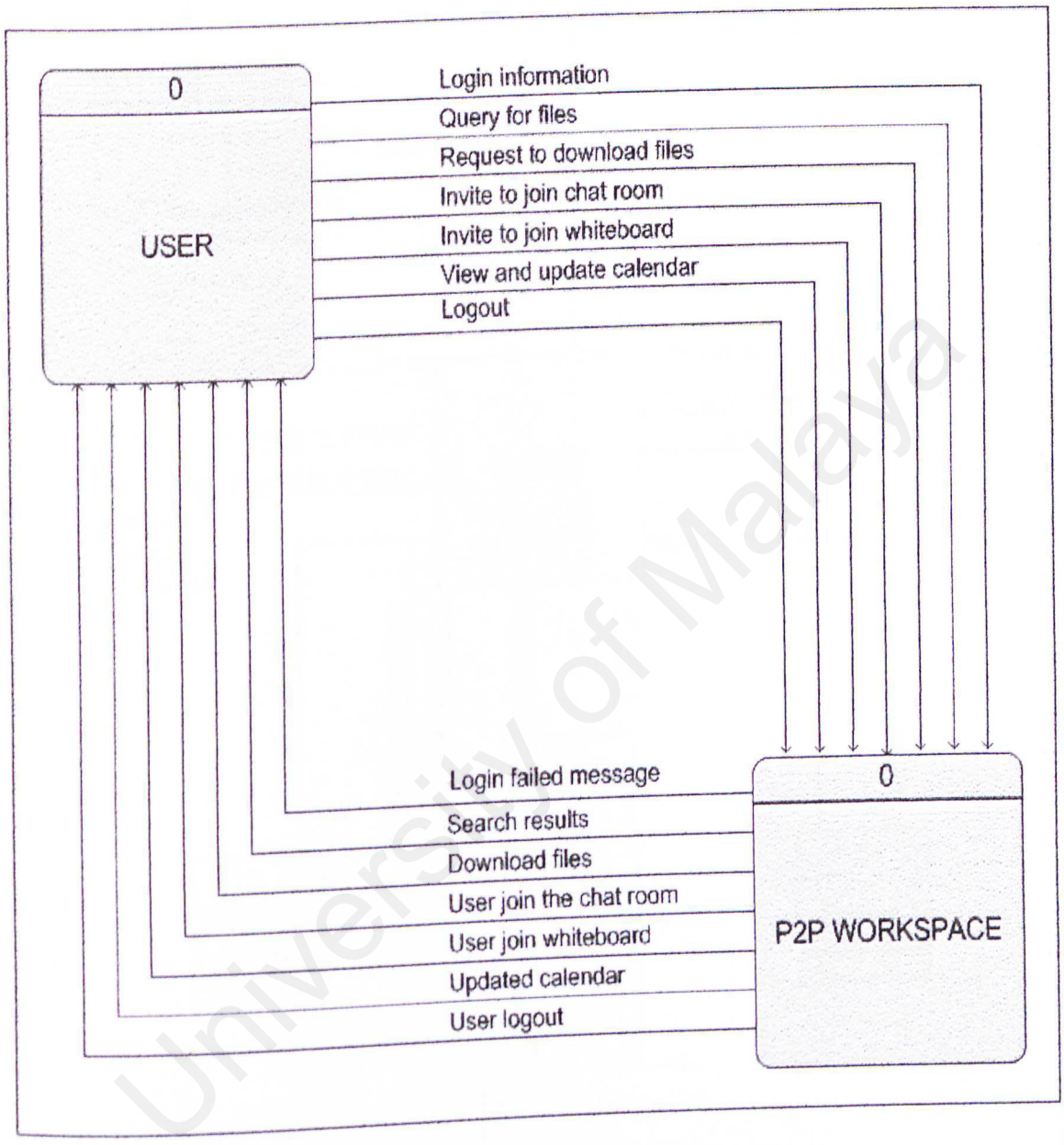


Figure 5.3: P2P Workspace Context Diagram.

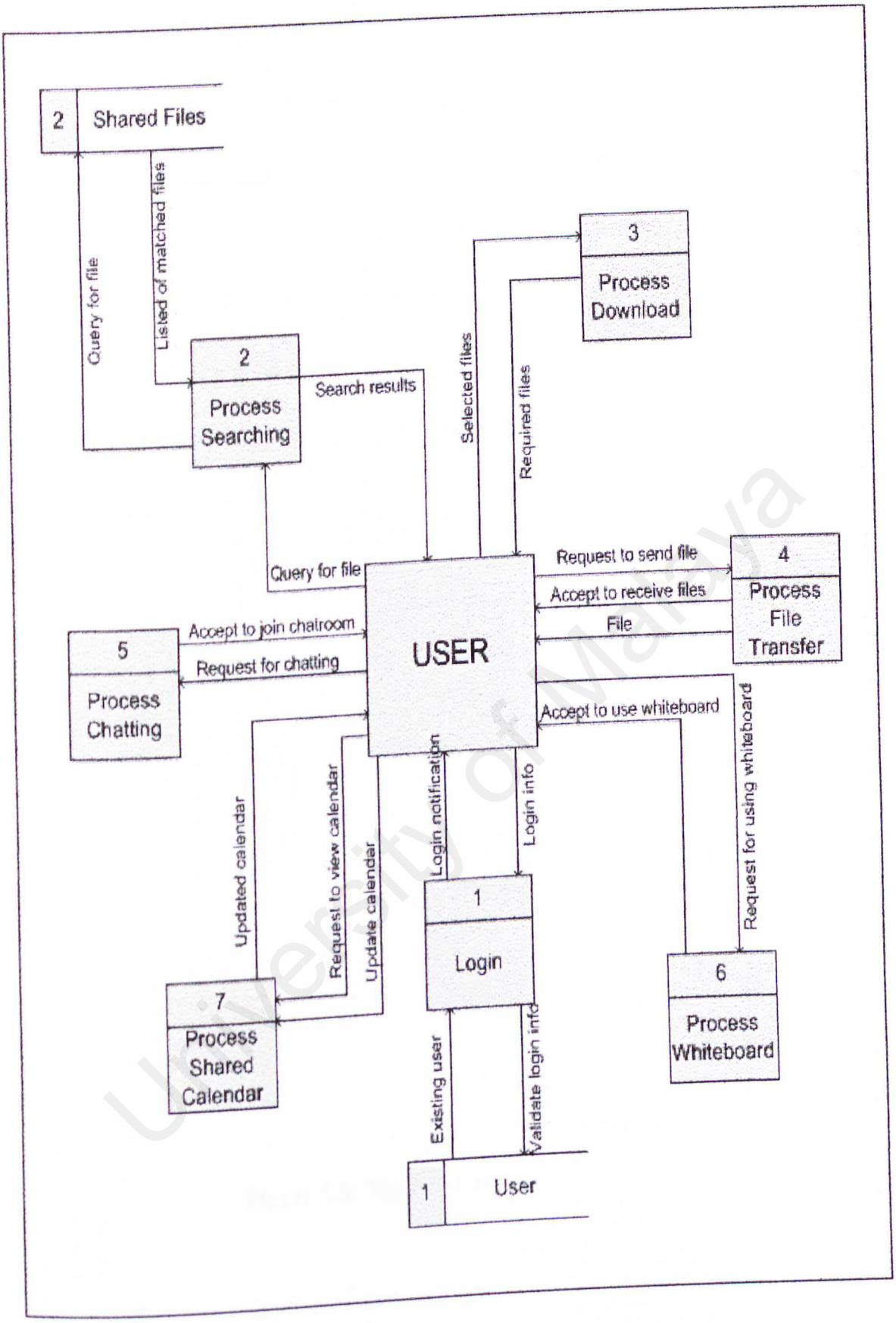


Figure 5.4: P2P Workspace Dataflow Diagram.

5.4 USER INTERFACE DESIGN

User interface design describes how system or application communicates with the human user who uses it. The P2P workspace user interface design focuses on the general interaction between its user and the system. It also takes into account development of complete and easy to understand information display.

The goal of a good interface design is to produce interface that helps users get the information they need in and out of the system. Below are the examples of the user interface for the P2P workspace.

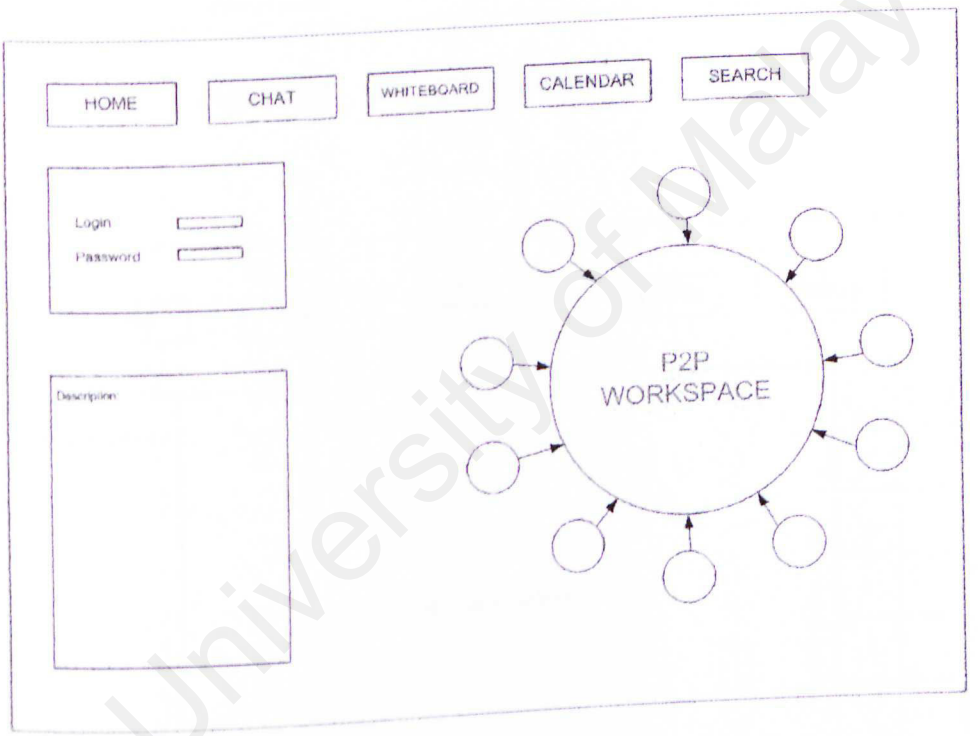


Figure 5.5: The front interface.

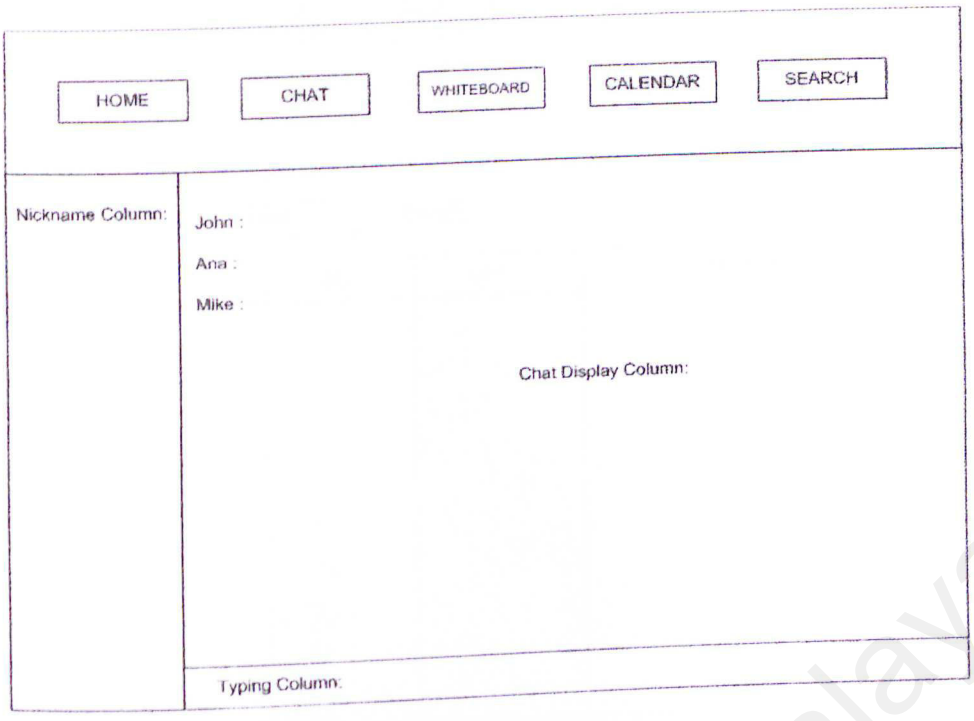


Figure 5.6: The chatting interface.

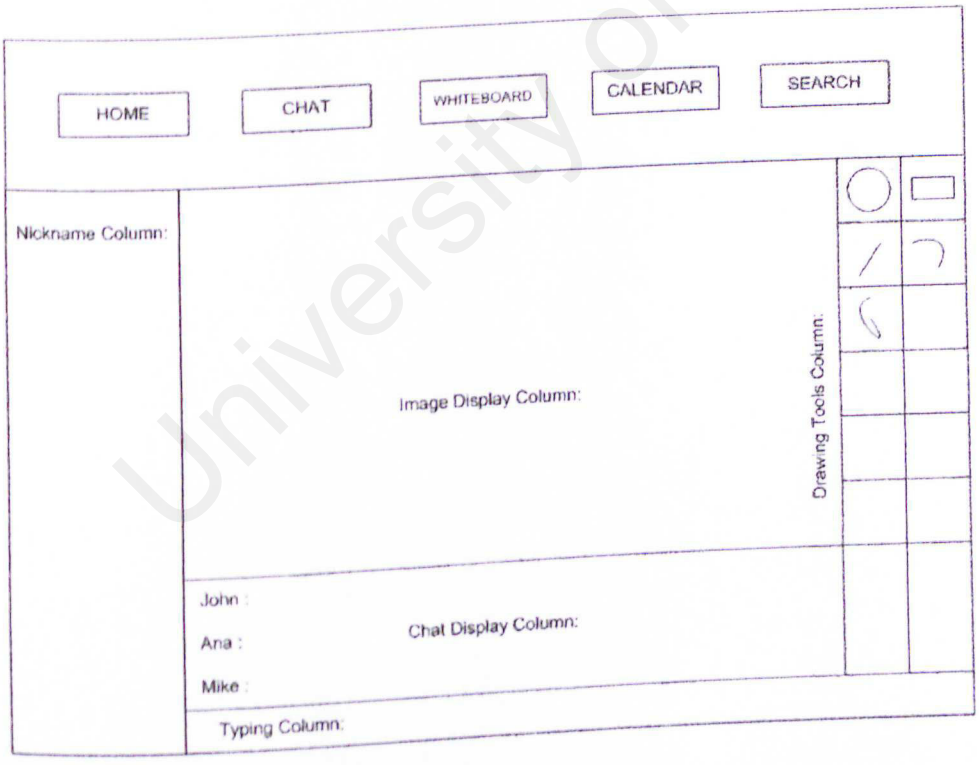


Figure 5.7: The shared whiteboard interface.

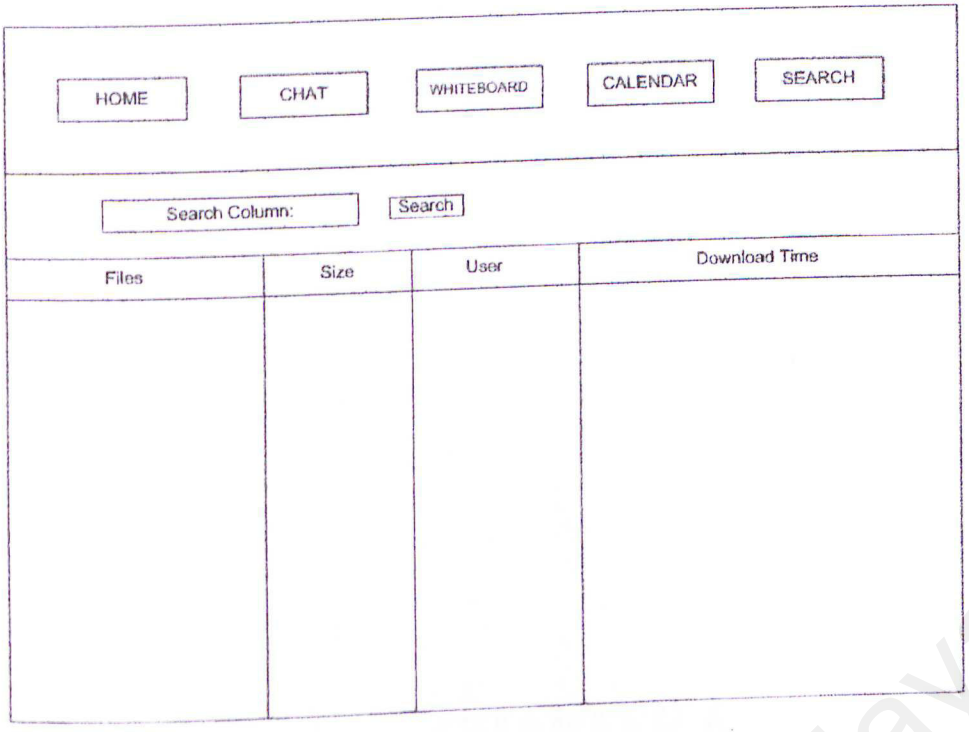


Figure 5.9: The file sharing interface.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.0 INTRODUCTION

The system implementation phase is an important element especially when interaction of system is needed between the subsystems or modules. In this phase, the issues of settings up the development environment which include software and hardware should be considered.

6.1 DEVELOPMENT ENVIRONMENT

Suitable development environment is a major factor in order to optimize the development progress. The software configurations of P2P Workspace are described in the following table.

Develop Tools	Function	Description
1. Microsoft Windows XP	Operating System Platform	Operating System
2. Microsoft Visual Studio.Net	Programming Tool	System Development Tools
3. .Net Framework 1.1	Development Platform	Development Platform

6.2 ALGORITHMS

The program design often specifies a class of algorithms to be used in the coding. A developer uses several ways to organize the codes such as keeping the program simple and tidy to make sure that the codes are readable and always put comments on code because additional comments are useful wherever helpful information can be added to a component.

6.3 CODING

6.3.1 C# Socket Programming

P2P Workspace is a peer-to-peer application which provides the file sharing service, chat and whiteboard. A socket must be create between each computers to communicate in order to transfer data and packets. A socket is a point to point communication link between two programs running on the network. For P2P Workspace, it acts as server and client at the same time.

TCP Socket

To allow file sharing, P2P Workspace acts as a server runs on the computer and has a socket that is bound to a specific port number which will be enter by the user. When the server is connected, it will listen and wait to the socket for a client to make a connection request and request for files.

P2P Workspace also can act as a client to search files in other peers or other computers shared folder. The client just has to know the hostname of the machine on which the server is running and the port number to which the server is connected.

Besides file sharing, P2P Workspace also offers a whiteboard service. This module also acts as a server and client. One user need to listen as a server and wait for a connection to be made with the client. The client needs to know the IP address of the server in order to get connected and start using the whiteboard.

For the chat module, we use a TCP socket with UDP multicast that will function as an auto discovery service at the time. To allow users to start chatting, just click on the chat button and the chatting module is ready to use.

6.4 MODULE IMPLEMENTATION

There are three main modules in the P2P Workspace application which is the file sharing, chat including the auto discovery and whiteboard. Each of these modules has different functionality.

1. Chat Module :

a. User List Box

- This box list the entire user online and entering this chatroom.

b. Chat Rich Text Box

- In this area, all conversation will be display to all users and display the message whenever a user enters the chat module.

c. Message Text Box

- This box provide user to enter the chat conversation and will be display at the Chatting Rich Text Box

2. File Sharing Module :

a. Search Function

- Area for user to download file by entering their search file based on their criteria.

b. Client/Server Shared Files

- This box list down the entire shared file by client or server.

c. Upload/Download

- In this area, all progress of downloading or uploading being show.

d. Options

- This function allow user to adjust the setting of the file sharing such as IP address, and file to be shared.

3. Whiteboard Module :

a. Toolbar

- Provide list of tools for drawing such as rectangle, circle, pen, eraser and options to save the image.

b. Draw Area

- Area for user to sketch their drawing.

c. Connection Status

- Inform the user that connection being made.

d. Connected Peers Box

- List of all users that entering the whiteboard session.

e. Listening/Connecting Box

- The function for establish the connection by listening if user act as server or connecting if user act as client.

CHAPTER 7

SYSTEM TESTING

7.0 INTRODUCTION

System testing is an essential phase to ensure the system performs according to its specifications and meet the user requirements and expectations. Testing is done throughout the system development not just at the end.

To detect the existence of faults in a system, testing is performed so that we could correct it. Therefore, a systematically test procedure is being used to ensure that the system is tested thoroughly and completely.

Three methods of testing have been carried out through the development of P2P Workspace. There are:

- Unit Testing
- Module Testing
- System Testing
- Performance Testing

7.1 UNIT TESTING

Individual components are tested to ensure that stand alone program fixes the bug without side effects. After new component is developed, it is tested independently without other system components. This is to make sure that the component is able to work accurately. All function on each button is examine to ensure it perform the entitles output such as call the right function to execute, link to the right page and many more.

7.2 MODULE TESTING

Module testing will focused on each sub module in P2P Workspace. Each of the sub modules is tested with the specific functions that they perform to see whether they really output the right result or fulfill the design requirement.

By using the P2P Workspace file sharing module as an example, each components or functions in this module has to be examined carefully to discover syntax error. P2P Workspace file sharing module is used to allow user download the files from other peers. As a result, every line of the codes related to these functions has to examine one by one to ensure that these functions fulfilled the user's requirements. If any errors are discovered, correction and debugging has to be carried out immediately to resolve those problems.

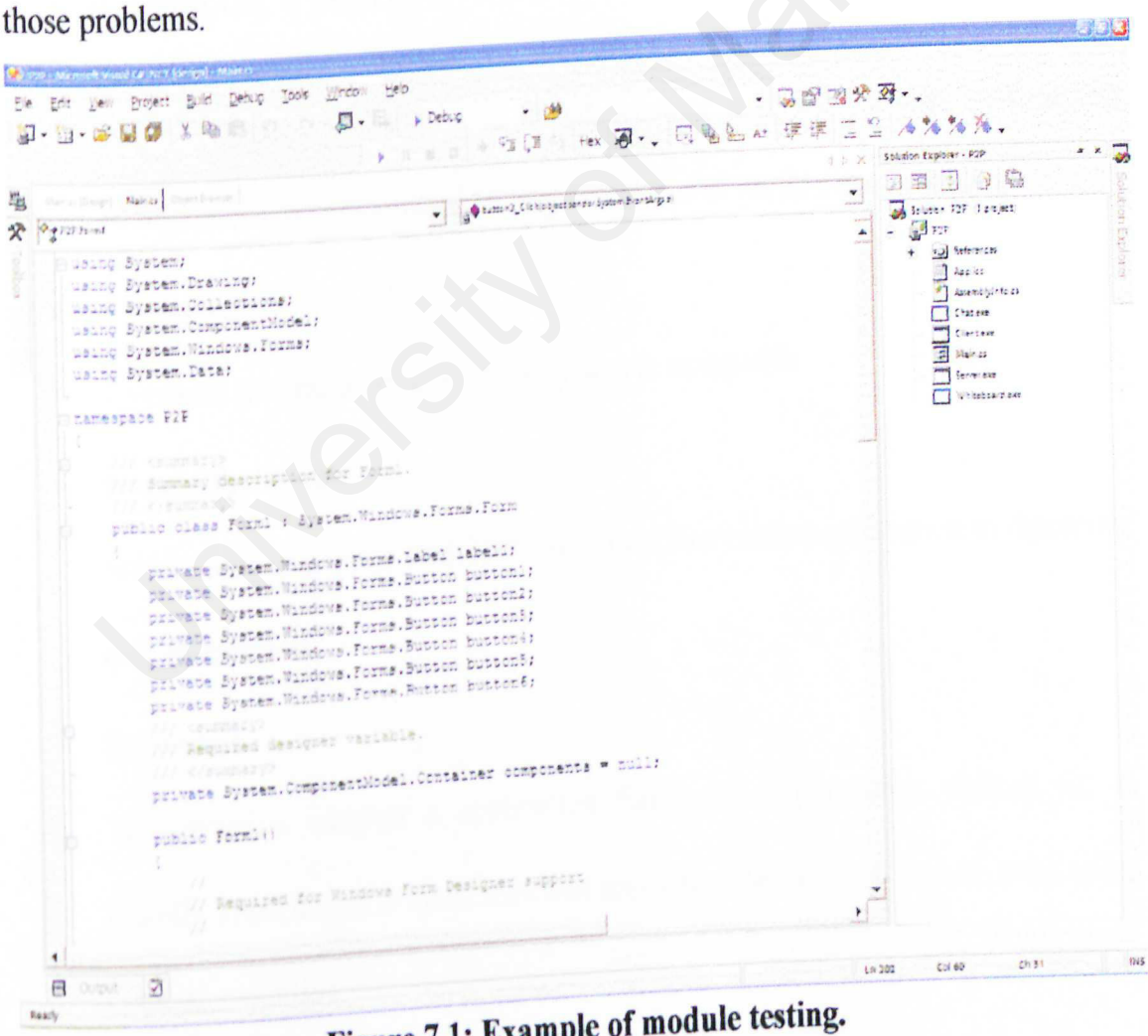


Figure 7.1: Example of module testing.

7.3 SYSTEM TESTING

System testing is different from other testing. System testing is the ultimate testing procedure. System test study all the concern issues and behaviors that can only exposed by testing the entire integrated system or major part of it. Under system testing, the whole process was simulated and followed through until the end.

Although some of the sub module had been tested for its functionality in the unit and module testing, now is the really testing to see the interaction from different system. Subsequently, corrections are done to the relevant components upon detection of faults or errors.

7.4 PERFORMANCE TESTING

P2P Workspace application is tested whether it meets specific performance efficiency objectives in performance testing. The types of performance tests are:

- **Timing test**

  To ensure the response time of the system is acceptable

- **Human factor test**

  Simple forms, displays and interactive interface related application to determine user friendliness.

- **Stress test**

  To determine whether a application fulfill the requirements defined for it. Equally important is to make sure that application works, as it should, even under extreme condition.

CHAPTER 8

SYSTEM EVALUATION

8.0 INTRODUCTION

System evaluation is the final phase of developing the P2P Workspace. System evaluation involves determine the problems and difficulties which arise during and after the application coding phase, recognizing the system strengths and weaknesses and finally draft out the system limitations and also its future enhancements.

8.1 SYSTEM STRENGTHS

As this project is about developing a P2P Workspace, here is some strength that this application has managed to reveal:

8.1.1 Auto discovery of each peer

This P2P Workspace has successfully implemented an auto discovery mechanism. Each user that uses the P2P Workspace will automatically being discover by entering the chat module.

8.1.2 Easy to use

Ease of use is the most important aspect in this application. This application is designed to allow a more efficient control over the P2P Workspace resources and time need to discovering other peers. This application also provides easy to use and user friendly interface, no training is needed to be specially conducted to learn how P2P Workspace works.

8.1.3 Organized and user friendly interface

All the modules in P2P Workspace have its own and standardized user interface, no command line is needed and easy to learn. Standardization in user interface design is making P2P Workspace flow easier to understand and reduce the complexity of the system.

8.2 PROBLEMS ENCOUNTERED

Several problems have occurred during the development and implementing of P2P Workspace. Some problems have been discovered and solutions have been solved during testing and reference check on the information through Internet and C# programming books.

8.2.1 Lack of knowledge

The most challenging part in developing this application was to understand, to master and to get familiar with the programming language, for example to master C#.Net in a short period of time. Other than that, is to understand how peer to peer application works, for example the transferring of packets and data across the network.

8.2.2 Understand the current system

It is quite difficult to understand in details on current peer to peer applications. Thus, we face problem on some details procedure on discovering other peers. Maximum hop, time to live expired and routing cannot be passed is the limitations in this part.

8.2.3 Difficulties in choosing the right platform and programming language

There are several platforms that we can use in developing a peer to peer application such as the Java platform and .Net framework. Choosing the right programming language was a critical process as each of the language has their strengths and weakness.

8.3 SYSTEM LIMITATION

There are several limitations in this P2P Workspace application. Although being able to perform the major task, the P2P Workspace still lacks of certain functionality such as :

- Resumable and multisource downloading
- Instant messaging and private chat

However, this functionality may provide the ideas for the future enhancements of P2P Workspace.

8.4 FUTURE ENHANCEMENTS

There is some functionality and module of the application can be enhanced in order to improve the quality of P2P Workspace. Below are some ideas to enhance the application:

- **Shared calendar module**

  Allows a user to put in their schedule, event or task in the calendar where other user can view it. A user can simply view and edit their own calendar to update the activities that is going to be held.

- **Module enhancement**

  - Resumable downloading is to resume a halfway downloads due to sudden offline and downloading is loss.
  - Multisource downloading is to save time of downloading a file from others.
  - Instant messaging and private chat in the chat module.

CONCLUSION

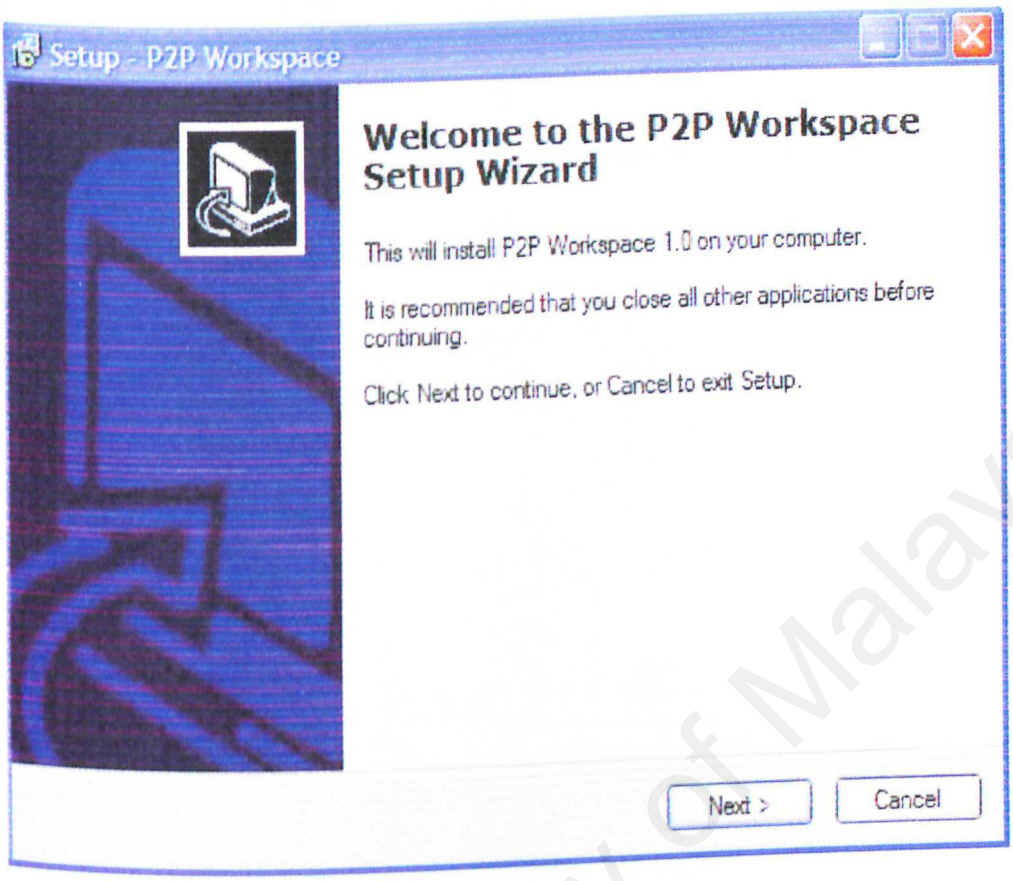
The target of this project is to develop a P2P Workspace which meets user expectation and requirement based on the included features like file sharing, chatting, shared whiteboard, and shared calendar. This P2P Workspace will provide a cost-effective and better solution for the user that want create a community although from different places.

In general, the P2P Workspace will provide the users with a user friendly, reliable and flexible workspace which make the services are efficient and easy to use.

P2P WORKSPACE
USER MANUAL

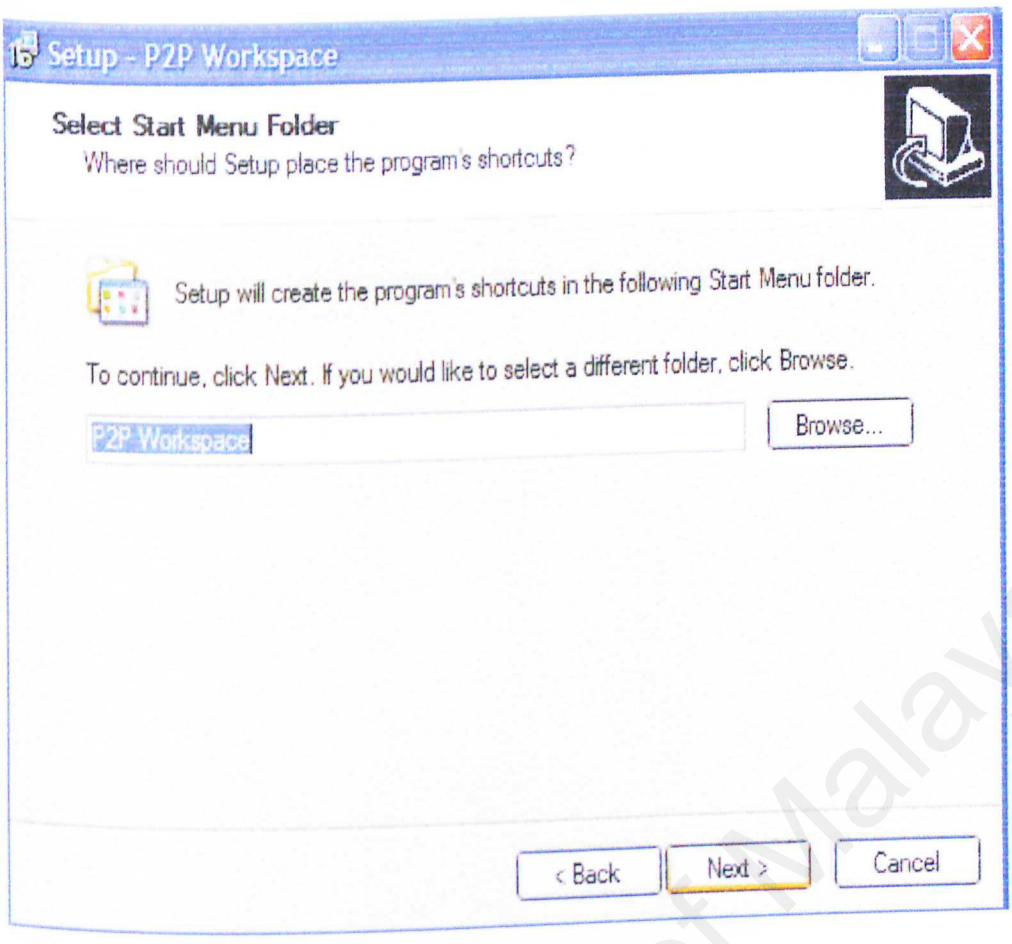
INSTALLATION GUIDE

STEP 1:



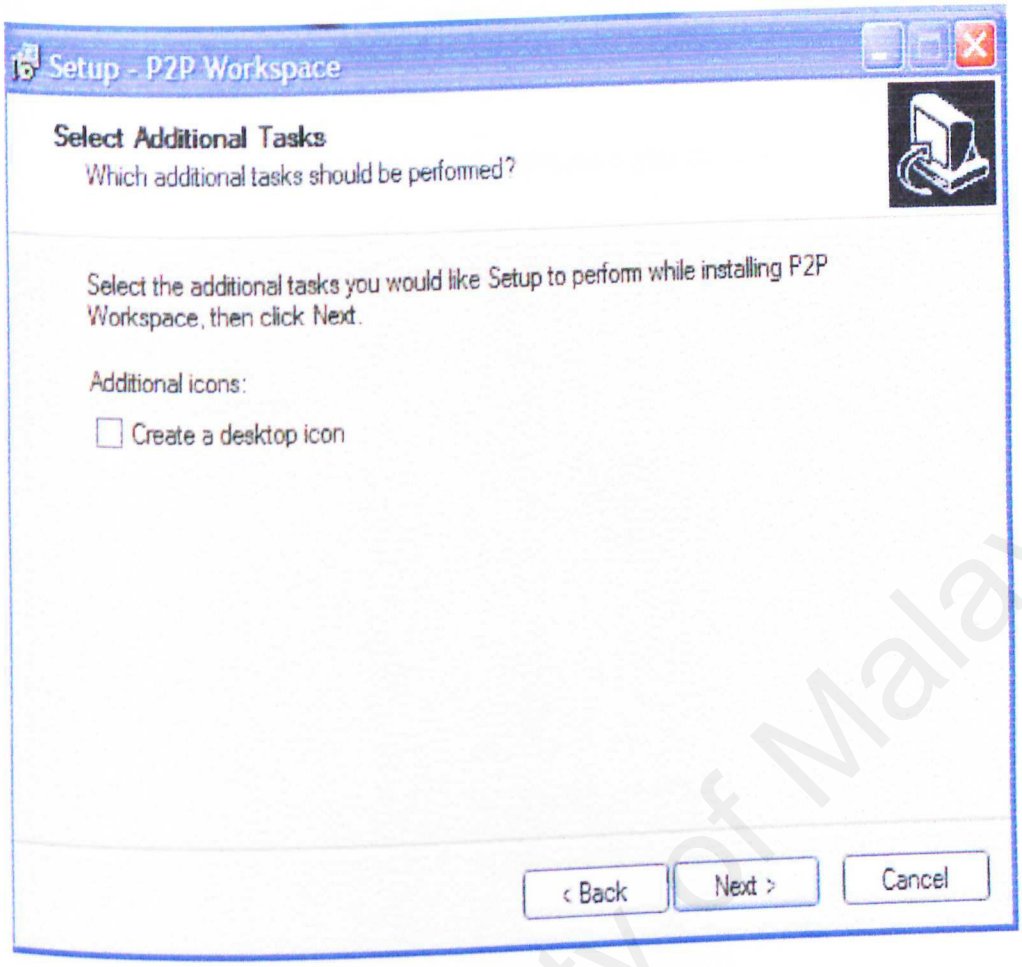
1. When you execute the P2P Workspace installation setup, this setup wizard will appear and click the 'Next' button to continue.

STEP 2:



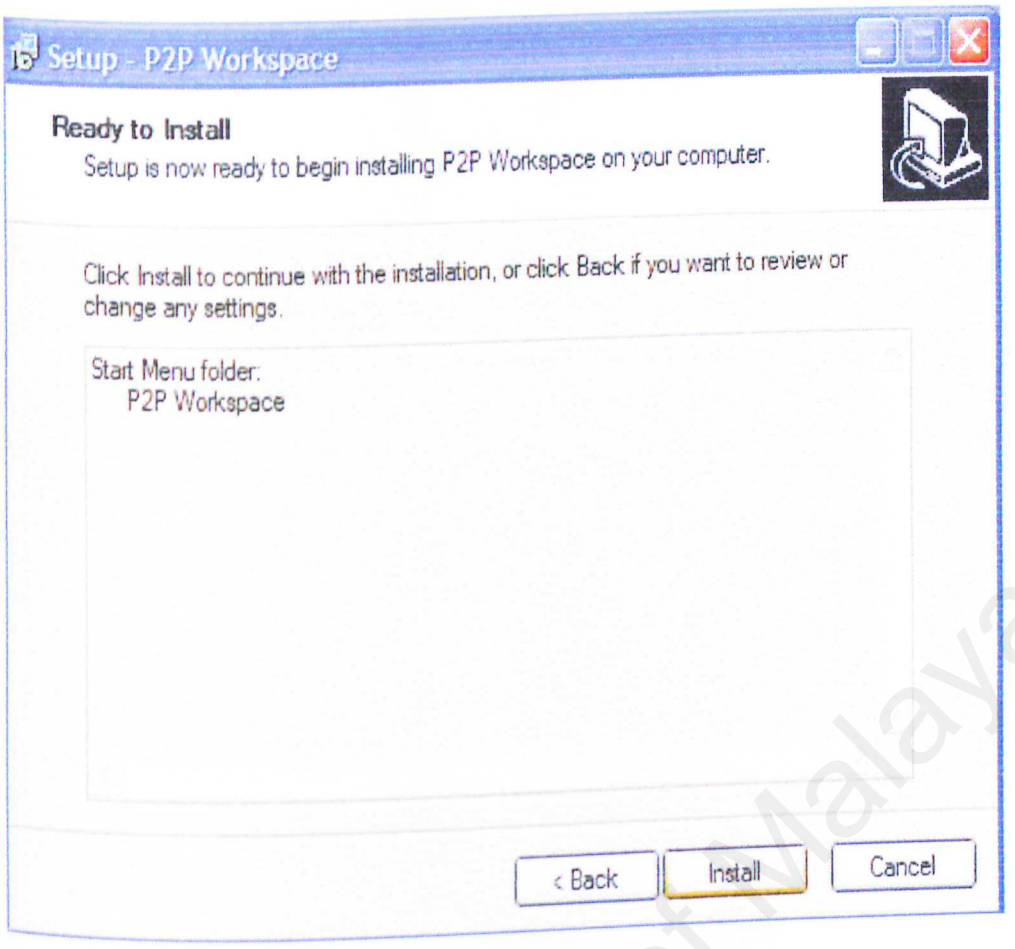
2. This wizard allows you to select the start menu folder on the start menu program. Click 'Next' to proceed.

STEP 3:



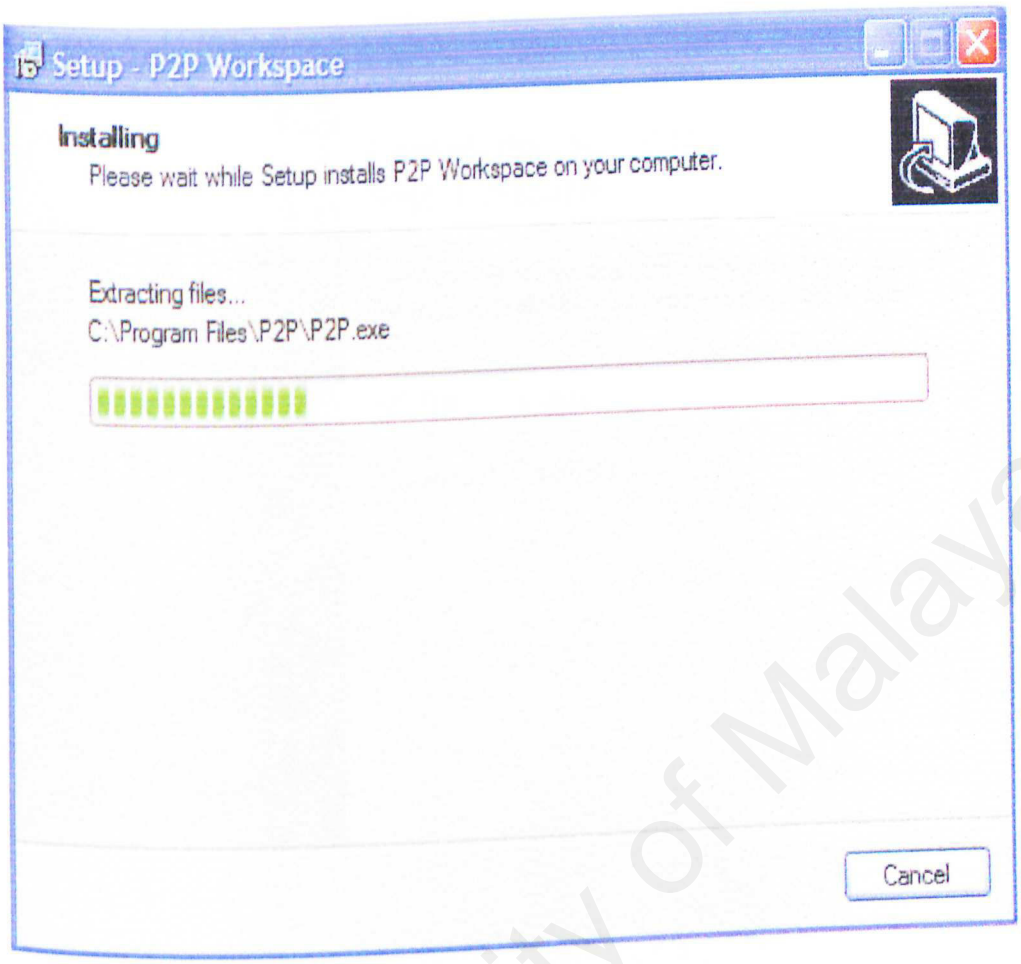
3. This wizard let you to choose whether to create a desktop icon or not. Then click 'Next' to continue.

STEP 4:



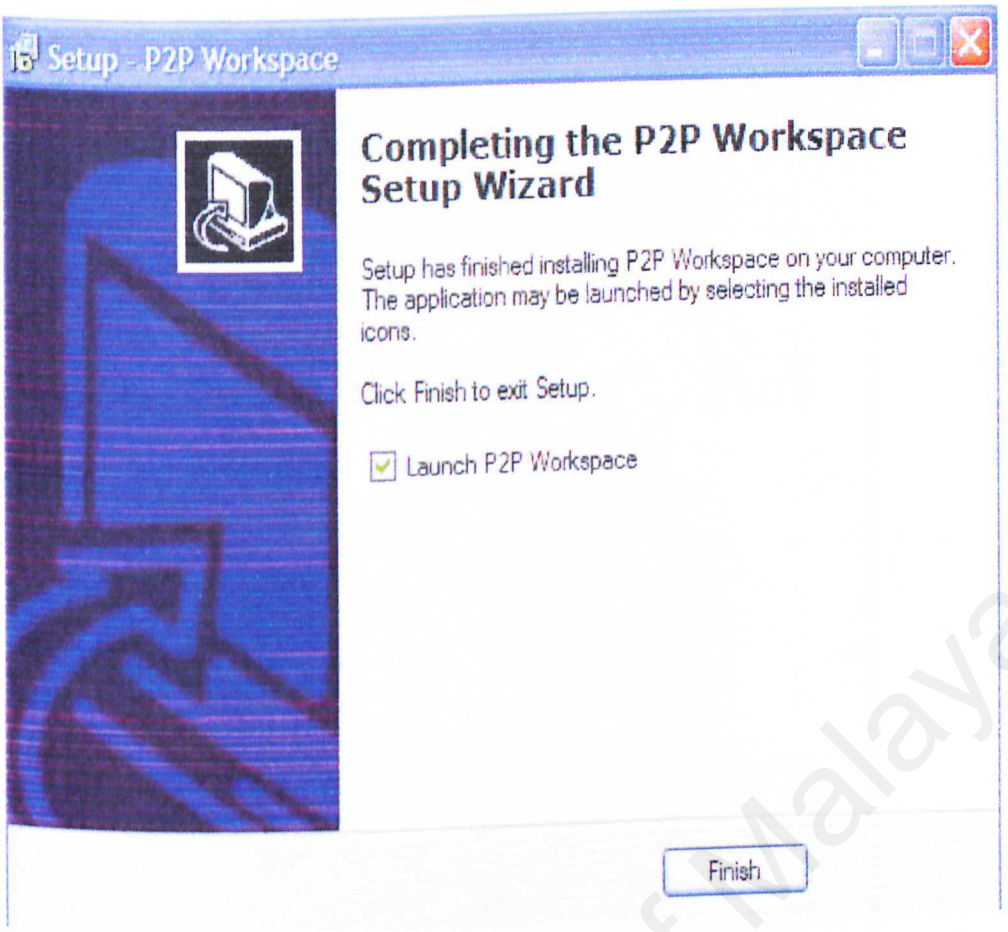
4. This wizard show the information of what are you going to install. Click 'Install' to start installation.

STEP 5:



5. Installation in progress. Wait until installation finish.

STEP 6:



6. The P2P Workspace has been installed. Tick the Launch P2P Workspace box and click 'Finish' button to start using the application.

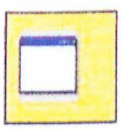
HOW TO USE THE P2P WORKSPACE

This is the main screen of the P2P Workspace application. There are 5 button which is for Chatting, File Sharing (Client & Server), Whiteboard, and to see user IP.

1. Click the  button to start chatting.

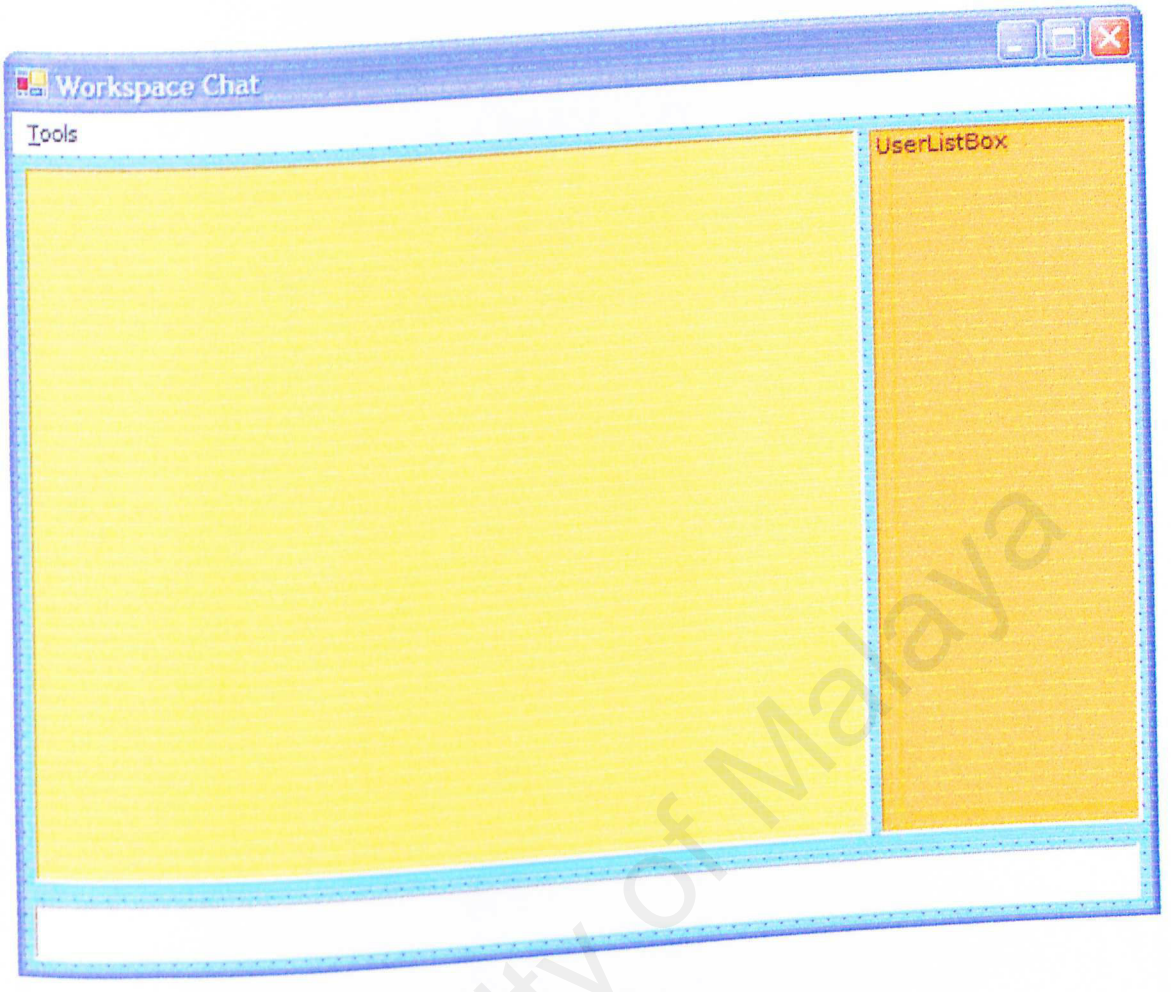
2. Click the 

button to start using file sharing.

3. Click the  button to start using the whiteboard.

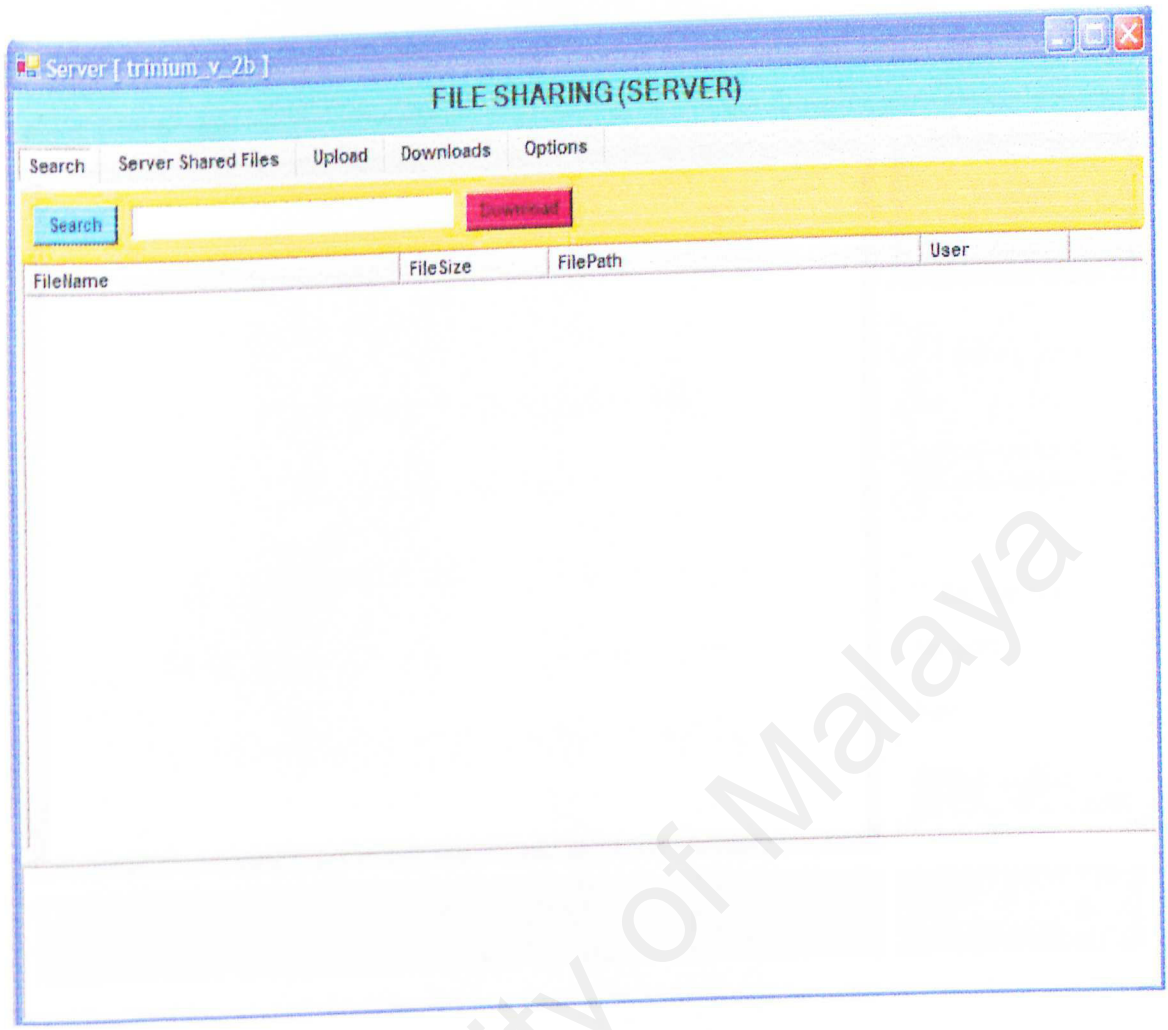
4. Click the  button to see your own IP address.

Chatting Module



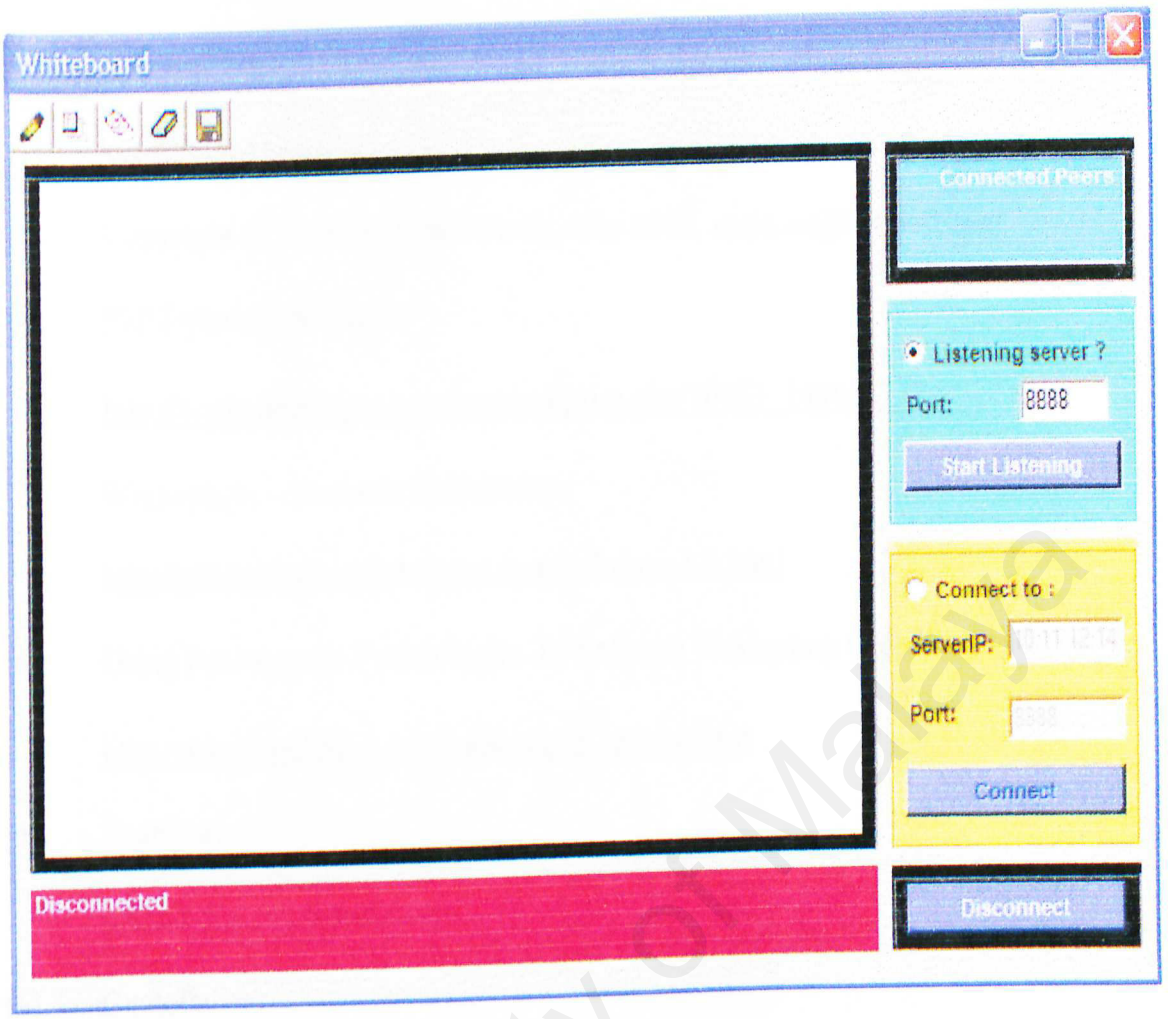
1. The tools option is for changing the username.
2. The user list box will display the entire user that has entered the chat room.
3. The yellow screen is where all the messages typed by user will be displayed.
4. The white box below the yellow screen is for user to enter the messages.

File Sharing Module



1. Search button is for user to enter their search file to be download.
2. The download button is to start downloading the file.
3. Shared Files button will list all the files shared by the user.
4. Upload / Download button will display the progress of uploading / downloading which consist file name, file size and percentage progress.
5. Options button allow user to change the setting of the file sharing module which consist the shared file path, and to enter the server IP address.

Whiteboard Module



1. The drawing tools button which consist pencil, rectangle, circle, eraser and the save image option will support user to start drawing.
2. The white screen is where the user starts drawing.
3. The red box below the white screen will show the user that connection has been made.
4. The blue box will list all the connected peers that using the whiteboard.
5. The green box is for the user to start listening as a server while the yellow box below is for another user makes a connection as a client.
6. The disconnect button is for user to stop using the whiteboard.

REFERENCES

Internet Resources

[1] Whiteboard Module

Copyright (C) Ranjeet Chakraborty, July 2002, ranjeetc@hotmail.com

[2] P2P Dynamic Networks

http://www.developer.com/java/ent/print.php/10933_1496861_1

[3] Workspaces – Business Collaborator

http://ghi.businesscollaborator.com/help/sec-4-1.html

[4] Using Peer-to-peer Technologies To Enhance Workgroup Collaboration

http://www.coulthard.com/papers/p2p_groove.htm

[5] Bearshare

http://www.bearshare.com

[6] Gnutella

http://www.gnutella.com

[7] Kazaa

http://www.kazaa.com

[8] Limewire

http://www.limewire.com

[9] Groove Networks

http://www.groove.net

[10] Morpheus

http://www.morpheus.com

[11] Project JXTA

http://www.jxta.org/

[12] P2P Network Architecture

http://wiki.cs.uiuc.edu/cs427/p2p

[13] Microsoft

http://www.microsoft.com

[14] Code Project

http://www.codeproject.com

[15] Planet Source Code

http://www.planet-source-code.com

[16] SourceForge

http://www.sourceforge.net

Articles and Papers

[17] Peer-to-Peer Networks

Marton Szoke, TENSA Technology Ltd, January 2002

[18] Peer-to-Peer Computing – A Technology Fad or Fact

Lisa Gutberlet

[19] Peer-to-Peer Computing

Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja , Jim Pruyne,

Bruno Richard, Sami Rollins ,Zhichen Xu, HP Laboratories Palo Alto,

8 March 2002