Real – Time Whiteboard

For

Ace Chat System

Prepared by

Ng Yih Young (WET990056)

Supervisor *: Ms. Nor Aniza Abdullah*

Moderator *: Dr. Rosli Salleh*

Faculty of Computer Science and Information Technology

University Malaya

2002

ACKNOWLEDGEMENT

The material for this project was developed with the help from many people. Here, I'd like to thank everyone who helped to see this project to completion. In particular, I would like to express a great appreciate to my supervisor, Ms. Aniza for her greatly benefited guidance, support and comments throughout the planning and development of this project.

I owe a huge thanks to my moderator, Mr. Rosli for his encourage and advise in this few months. I will appreciate his kindly advise on my project.

Besides, I would like to express my appreciation and thankful to all the great folks and course mates for their terrific support and help – especially Mr. Tan Check Leam and Ms. Tan Su Yuen who are my team partners, Mr. Chong Bing Suang and Mr. Lim Boon Hong who are also my best friends helping me in information gather and system analysis. Finally, the completion of this report depends largely on the resources available through the Internet, reference books, and past year thesis reports.

Content

Chapter 3: System Analysis

Chapter 4: System Design

Chapter 5: System Implementation

Chapter 6: System Testing

Chapter 7: System Evaluation and Enhancement

List Of Figures

List of Tables

ABSTRACT

Animated chat room is a system that combines chat application, animation, Text – To – Speech system, and real time activity. This huge system is divided into three major parts which are chat room application and graphical user interface (GUI) design, animation and Text – To – Speech (TTS) design, and real time activity – whiteboard. Mr. Tan Check Leam will in charge in the chat room application, Ms. Tan Su Yuen will cover the animation and TTS design and I myself will involve in the real time activity – whiteboard.

The whiteboard system is an enhance feature for the chat room application. Whiteboard allows users at the same session (on-line at the same time) to post object or to draw something for others users to see. Whiteboard shares users creativity. Nowadays, although there are a lot of whiteboard system, but this whiteboard will still show the different from them because this system will focus on real time activities and user-friendly interface.

A lot of the research and review had done to complete this system. For instance, Real-time Transport Protocol (RTP), Object – Oriented Programming language, methodology, Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and so on.

The whiteboard module can be divided into 4 sub modules, there are drawing module, text module, colour module and manipulation module. Each module will focus on different parts of the system.

CHAPTER 1 PROJECT REPORT

1.1 Project Definition

1.1.1 Introduction

Nowadays, chat application is widely used to 'connect' people from different locations. If we look at the development of the chat application, we will discover that the chat application nowadays has becomes more and more complex as people demand more on its functionality and effectiveness. Research and study on chat room are being carried out intensively in order to make it more user-friendly and multifunction.

Animated chat room is a way to enhance the chat room by adding the multimedia component such as animation and graphics on the chat room to make it more attractive. Animated chat room can be divide into 3 major parts, there are:

1. Online chat room and graphical user interface (GUI) design

   This part will focus on how to setup an online chat room, this include design the database and setup all the connection for the multicast chat room, and design an attractive graphical user interface.

2. Text-to-Speech system and animation design

   Animated chat room will provide text-to-speech features, this means when user type in a word or sentences, system will pronounce what user had type in.

   Besides, the job scope for this part also includes design animation for the chat application.

3. Real time activity

   The third part for the animated chat room is to setup a real time drawing tool, whiteboard, for the chat room. My thesis will focus on this part. Whiteboard is an application that provides an area for user to post object and share among the users at the same session. Besides, user can receive other users' posts too. When a user posts an object, other users who logon at that time (at the same session) can see it. Basically, a whiteboard will provide text tools and drawing tool, the user's drawing will share among all the users and it also allow multiple users to work on the same drawing.

1.1.2 Objective

The real time drawing tools, whiteboard, is an enhance function for the chat room. This additional feature enables users to draw something besides chatting and share it among the users. The objectives of the project are:

1. To create an infinite two-dimensional space in which user can post objects for all users to see.
2. To share the whiteboard among multiple users.
3. To share users' creativity.
4. To provide a more expressive way instead of text chatting.
5. To create a user-friendly environment for chat room.

1.1.3 Problem and inconveniences

There are a lot of chat application that provide the real time drawing feature, but several problems and inconvenience still exist. The problems and inconvenience include:

1. Limited functionality.

   The drawing editor not provides the enough functionality for the users to draw. Many of the drawing editors only provide the basic freehand drawing tools, and most of the users are not satisfy with the existing tools.

2. The drawing tools are difficult to use.

   This means that the drawing editor is not user-friendly enough. Some online drawing editors provide enough functionality but the weakness is the graphical user interface, GUI, for the drawing editor is difficult to understand or sometimes make the users confuse.

3. Data transfer is not smooth enough.

   Some of the input reach their partner/partners side at different transfer rate. It is a big problem for users of the application because they have to wait each time they draw. This may cause the program slow down.

1.1.4 Expected outcome

As we see that there are still a lot of problems for the real time drawing activity, therefore, this application must show some improvement compared to the existing application. The expected outcomes are as below:

1. User friendly.

   The application must be easy to use and understand for the users from any levels.

2. More attractiveness.

   Besides the enough functionality and features, the graphical user interface must be nice and user friendly to increase the attractively. Therefore, users will satisfy with the program.

3. Easy for enhancement and maintenance.

   The coding for the application must be very clear and systematic; hence, the application will become more flexible. If there are any problems, it will be easy to enhance and maintain.

4. Smooth and fast enough data transfer.

   When we look forward to the problem cause by the time delay of the data transfer, it is possible to design a more effective architecture for this application to solve the problem. The participants will satisfy with the application only if the data transfer is smooth and fast enough.

1.2 Project Scope

1.2.1 Features

The features of the animated chat room are like below:

1. Real time drawing tools.

   The application might provide an area for the users to draw and display the output, act like a whiteboard. It is a real time activity that all the users at the same session can see the output.

2. Freehand drawing tools.

   The application will provide freehand drawing feature for the users.

3. Basic shape drawing tools.

   Users can select the basic shape and drag it into the drawing area instead of use freehand drawing tools. The basic shape includes rectangular, oval, polygon, and so on.

4. Text editor.

   Users can input text into the drawing area as the description of the drawing.

5. Eraser tool.

   This feature let users improve their drawing.

6. Color and drawing tools setting.

   Users can select the color and other setting according to their preference. For example, users can select the different pen or brush size to draw. Users are allowed to select the line's width too.

1.2.2 Conceptual design

Conceptual design means to describe the system in language that the customers can understand. In this session, developer tells the customer exactly what the system will do.

Whiteboard system is an application that provides an area for the users to post the drawing. Whiteboard is an enhance features for the chat room and it is easily to use. Users can draw anything on the whiteboard area. Whiteboard will provide a set of drawing tools button for users to use. For instance, when user needs to draw a circle on the whiteboard, user can either select the basic shape button or freehand drawing button. With the basic shape button, circle shape is provided and what user does is just click on the circle button and drag it into the drawing area. User can adjust the circle's radius too. If user selects the way to use the freehand tools to draw a circle, what user does is select the freehand button and then draw a circle on the drawing area by mouse.

Whiteboard can let users to select their preference colour to draw or fill in. Besides, the text size can also change by the users. Users can change the colour and the text size from the drop down list. Eraser tool is provided to improve users' drawing.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

Before start develops the whiteboard system, developer must gain some knowledge and background study. The purpose of having literature review is to make a better understanding about the development tools used.

The literature review on this chapter can be divided into 4 parts, which are: -

- Methodology
- Real time Transport Protocol
- Transport Protocol
- Programming Tools

2.2 Methodology

2.2.1 Prototyping Model

Prototyping is the technique of constructing a partial implementation of a system. It involves users, customers and developers to learn more about the problem or solve the problem together. Prototype can be defined as a first full-scale and usually functional form of a new type of design of a construction while software prototyping is an information system development methodology based on building and using a model of a system for designing, implementing, testing, and installing the system.

Prototyping is based on building a model of a system to be developed. The initial model should include the major program modules, the database, screens, reports, and the inputs and outputs. Prototyping allows all or part of a system to be constructed quickly to understand or clarify issue; it has the same objective as an engineering prototype, where requirements or design require repeated investigation to ensure that the developer, user, and customer have a common understanding both of what is needed and what is proposed. One or more of the loops for prototyping requirements, design, or the system may be eliminated, depending on the goals of the prototyping. However, the overall goal remains the same – reduce the risk and uncertainty in development.

System developer first begin a nominal set of requirements supplied by the users and customer, then, alternatives are explored by having interested parties look on system output. Customers and users will give feedback on the prototype and the requirements are revised until the developers, users, and customers are happy with the result. Once there is a common agreement, the system developer will move on to the next stage and again the alternatives are explored. The same procedures will loops. The prototyping model are shown as below:

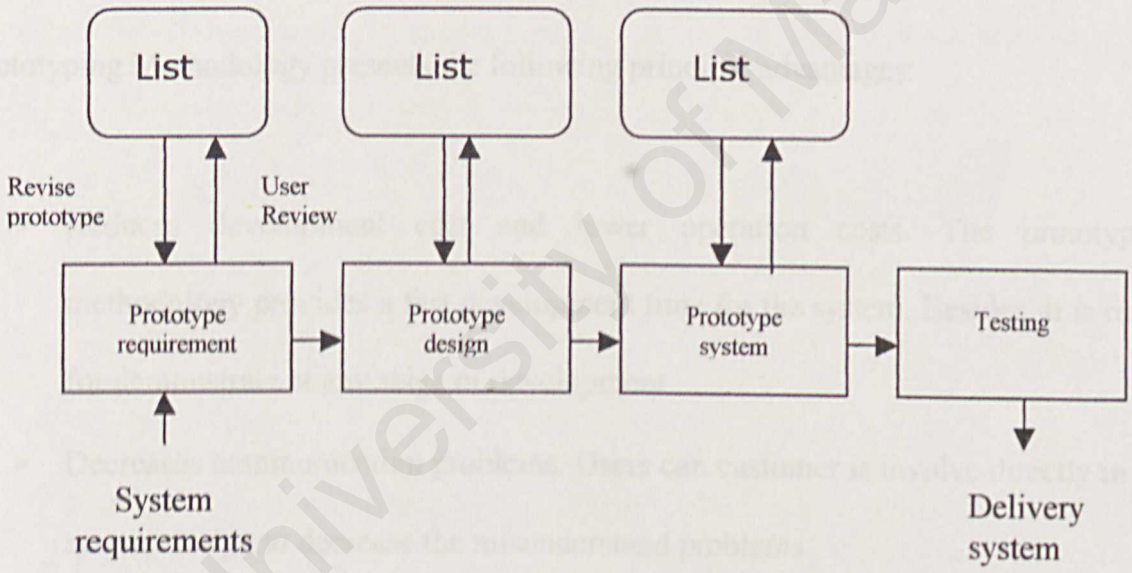


Figure 2.1 the prototyping model

Prototyping Methodology

1. Define Problem
2. Gather General Information about Study Area
3. Understand Existing System
4. Develop Model of New System
5. Users Evaluate Model
6. Revise Model & Re-evaluate until user satisfied with the design

Prototyping Methodology presents the following principal advantages:

- Reduces development cost and lower operation costs. The prototyping methodology provides a fast development time for the system. Besides, it is ready for demonstrate at any stage of development.
- Decreases communication problems. Users can customer is involve directly in the system design to decrease the misunderstand problems.
- Find out the requirement errors earlier. Requirements validation is performed since users are able to experiment with requirements and the system. The requirements validation process consists of seven factors: correctness, consistency, traceable, realistic, needed, verifiable, and completeness.
- Produces the right system the first time
- Slashes calendar time required
- Please users

However, Prototyping Methodology has also disadvantages. Here are some of the possible ones:

- Visible use of computer resources
- Object system may be less efficient
- Requires cooperation between user and Information Systems

There are two primary schools of prototyping – **throwaway** and **evolutionary**. In the **evolutionary approach**, the prototype is constructed in order to learn more about the problem and its solution; the prototype will reuse after the revision, this means the prototype is adapted to satisfy the now better-understood needs from the users and customers. This process is repeating until the prototype is satisfy all needs and has thus evolved into the real system.

The **throwaway prototyping** is different from the evolutionary prototyping because the prototype software is constructed in order to learn more about the problem or its solution and is discarded after the desired knowledge is gained.

However, customers and end-users should not take the Throw - away and make it into a final delivered system due to the following reasons:

- Important system characteristics may have been relaxed during rapid prototype development (security, robustness or reliability).
- Changes made during prototype development are usually made in an uncontrolled environment. The prototype code may be the only design specification.
- The changes made during prototype development can degrade the system structure, making the system difficult and expensive to maintain.

2.2.2 Waterfall Model

The waterfall model is a systematic sequential approach to software development modeled after a conventional engineering cycle. It also known as the life-cycle paradigm, one phase must complete before move to another phase. The waterfall model consists of system engineering and analysis, software requirement analysis, design, coding, testing and maintenance phase.

- System engineering and analysis

  This phase focus on gathering all system elements, which then reduced to a subset of requirements that are relevant and suitable to the software being developed.

- Software requirement analysis

  In software requirement analysis phase, developer will defines functional capabilities, performance, design constraints and system interface. It provides the software designer with representation of information and function that can be translated to data, architectural and procedural design.

- Design

  Design phase means translate all the requirement of software into representation of software that can be assessed for quality before coding begins. The output of this activity included data structures, architecture, procedural detail and interface characterization. There are two sub-phases in this phase.

  - Preliminary design to transform requirements into architecture.
  - Detailed design to define and document algorithms for each module in the design tree that will be realized as code.

- Coding

  This phase focus on transform the design specification into software program.

- Testing

  Testing is an important phase to ensure that all the system requirements is fulfilled and the system is bug free. Testing phase can be divided into three minor phases: -

  - Unit testing is use to check each coded module for the presence of bugs.
  - Integration testing is use to check modules are working properly while there are interconnects.
  - System testing is use to ensure that the system does what the customers want it to do.

- Maintenance

  Maintenance phase is to make the internal and external changes on the particular system. This phase will focus on error detection and collection too.

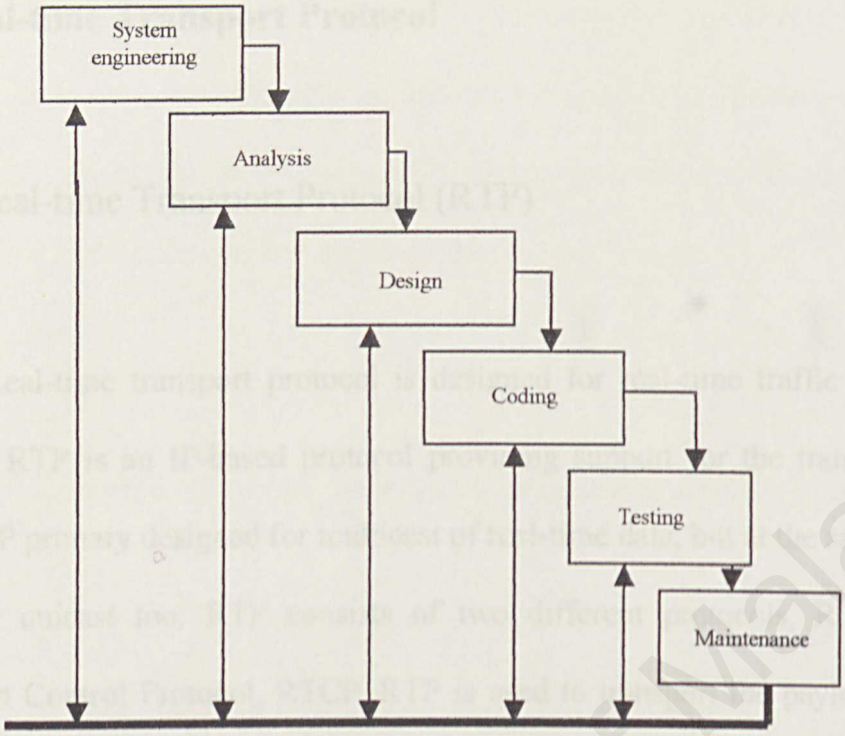


Figure 2.2 the Waterfall Model paradigm of software development

2.3 Real-time Transport Protocol

2.3.1 Real-time Transport Protocol (RTP)

Real-time transport protocol is designed for real-time traffic over Internet and intranet. RTP is an IP-based protocol providing support for the transport of real-time data. RTP primary designed for multicast of real-time data, but at the same time, it can be used for unicast too. RTP consists of two different protocols, RTP and Real-time Transport Control Protocol, RTCP. RTP is used to transport the payload data while the RTCP is used for exchange meta-information about the current session.

RTP services

RTP provides timestamping, sequence numbers, payload type identifier, and source identification. Though these mechanisms, RTP provides end-to-end transport for real-time data over datagram network.

Timestamping is the most important information in the RTP header. The sender timestamps each RTP packet according to the first sampled in the packet was encoded. The receiver will uses the timestamp to reconstruct the original timing in order to play out the data in the correct rate. However, RTP does not responsible for the synchronization itself, this has to be done in the application level.

RTP is typically run on the top of *User Datagram Protocol, UDP*. This is to make use of its multiplexing and checksum service, but UDP does not deliver packets in timely order, thus, sequence numbers are used to place the incoming data packets in the correct order. Sequence number is used to detect the lost packet too.

Payload type identifier is used to let the receiving application knows how to interpret the payload. Examples for payload formats are different versions of PCM, and JPEG.

Source identification is another function in RTP to allows the receiving application to know where the data is coming from.

RTP Architecture

An RTP *session* is an association among a set of applications communicating with RTP. A network address and a pair of ports identify a session. One port is used for the media data and the other is used for control (RTCP) data.

A *participant* is a single machine, host, or user participating in the session. Participation in a session can consist of passive reception of data (receiver), active transmission of data (sender), or both.

The RTP header has the following format:

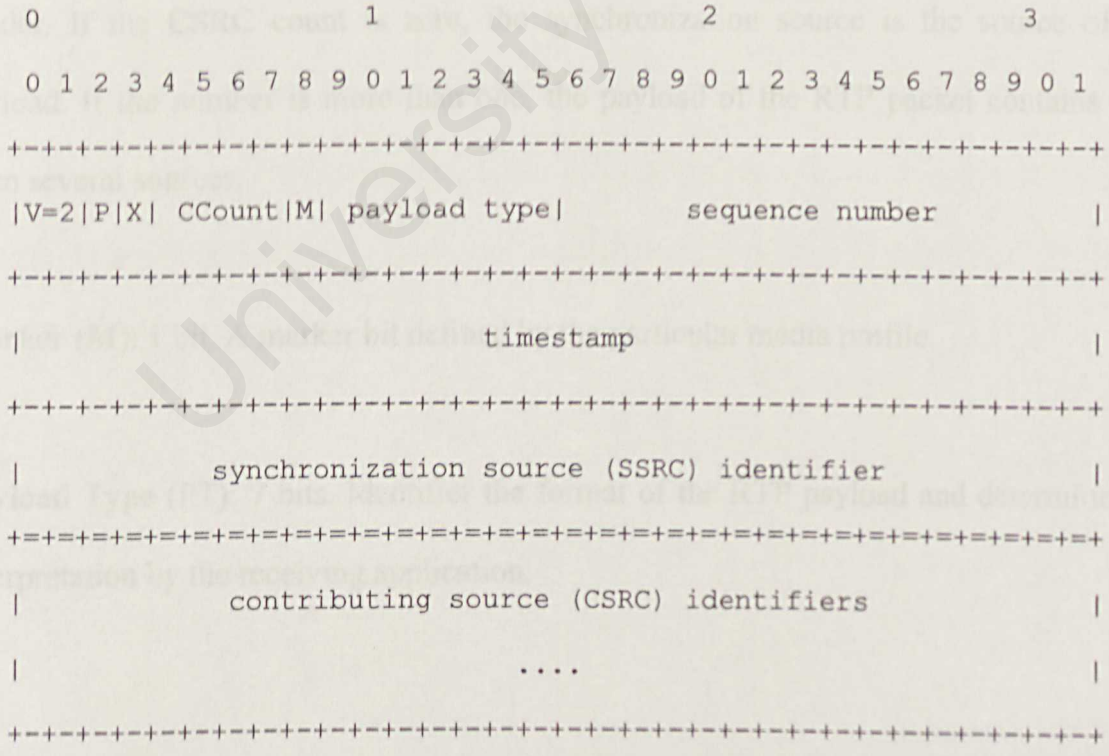


Figure 2.3 RTP Packet Header

The RTP version number (V): 2 bits. The newest version is 2.

Padding (P): 1 bit. If the padding bit is set, there are one or more bytes at the end of the packet that are not part of the payload. The very last byte in the packet indicates the number of bytes of padding. Some encryption algorithms use the padding too.

Extension (X): 1 bit. If the extension bit is set, the fixed header is followed by one header extension. This extension mechanism enables implementations to add information to the RTP Header.

CSRC Count (CCount): 4 bits. The number of CSRC identifiers that follow the fixed header. If the CSRC count is zero, the synchronization source is the source of the payload. If the number is more than one, the payload of the RTP packet contains data from several sources.

Marker (M): 1 bit. A marker bit defined by the particular media profile.

Payload Type (PT): 7 bits. Identifier the format of the RTP payload and determines its interpretation by the receiving application.

Sequence Number: 16 bits. A unique packet number that identifies this packet's position in the sequence of packets. The packet number is incremented by one for each packet sent.

Timestamp: 32 bits. Reflects the sampling instant of the first byte in the payload. Several consecutive packets can have the same timestamp if they are logically generated at the same time--for example, if they are all part of the same video frame.

SSRC: 32 bits. Identifies the synchronization source. If the CSRC count is zero, the payload source is the synchronization source. If the CSRC count is nonzero, the SSRC identifies the mixer.

CSRC list: 0 to 15 items, 32 bits each. Identifies the contributing sources for the payload. The CSRC count field indicates the number of contributing sources; there can be up to 16 contributing sources. If there are multiple contributing sources, the payload is the mixed data from those sources. The CC field gives the number of identifiers.

2.3.2 Real-time Transport Control Protocol, RTCP

RCTP is the control protocol designed to work in conjunction with RTP. RTCP packets are sent periodically to all participants in the same session. The main function of the RTCP packets is to provide feedback on the quality of the data distribution. RTCP packets can contain information about the quality of service for the session participants, information about the source of the media being transmitted on the data port, and statistics pertaining to the data that has been transmitted so far.

The several types of the RTCP packets are:

- **RR:** receiver report. Participants that are not active senders generate receiver reports. They contain reception quality feedback about data delivery, including the highest packets number received, the number of packets lost, inter-arrival jitter, and timestamps to calculate the round-trip delay between the sender and the receiver.
- **SR**: sender report. Active senders generate sender reports. In addition to the reception quality feedback as in RR, the sender report (SR) contains the total number of packets and bytes sent as well as information that can be used to synchronize media streams from different sessions.
- **SDES**: source description items. All the compound RTCP packets must include a SDES that contains the canonical name, CNAME that identifies the source. In this packet, the additional information must be the email address,

phone number, source's name, application name or a message describing the current state of the source.

- **BYE**: when a source is no longer active, it sends a BYE packet. The packet might contain the reason why the source is leaving the session.
- **APP**: application specific functions. It is now intended for experimental use as new applications and new features are developed.

The first packet in a compound RTCP packet has to be a report packet, even if no data has been sent or received, in that case, an empty packet is sent.

Through the control information packets, RTCP provides the following services:

- **QoS monitoring and congestion control.** This is the primary function of RTCP. RTCP provides feedback to an application about the quality of data distribution. The control information is useful to the senders, the receivers and third-party monitors. The sender can adjust its transmission based on the receiver report feedback.
- **Source identification.** RTCP SDES (source description) packets contain textual information of the session participants. It may include user's name, telephone number, email address and others information.
- **Inter-media synchronization.** RTCP sender reports contain an indication of real time and the corresponding RTP timestamp. This can be used in inter-media synchronization like lip synchronization in video.

- **Control information scaling.** When the number of the participant of the session increase, it is necessary to balance between getting up-to-date control information and limiting the traffic. Therefore, to scale up a large multicast group, RTCP has to prevent the control traffic from overwhelming network resources. This enforced by adjusting the RTCP generating rate according to the number of participants.

As conclusion, the useful of the RCTP feedback are:

- To adapt the different network condition
- To locate problems and diagnose faults
- To keep track of participants when their interval identifier change
- To distribute information about all the participants in a session

Nowadays, the efforts have been made to make the RTP transport-independent so that they can be connect through AAL/ATM, IPX (Inter-network Packet Exchange) and others protocol.

2.4 Transport Protocol

2.4.1 Transmission Control Protocol and Internet Protocol (TCP/IP)

Introduction

When communication is desired among computers from different vendors, the software development effort can be a nightmare. Different vendors use different data formats and data exchange protocols. TCP/IP provides reliable connection oriented, peer to peer communications and many services like FTP, Telnet, email can provided by implementation of TCP/IP.

TCP/IP Protocol Architecture

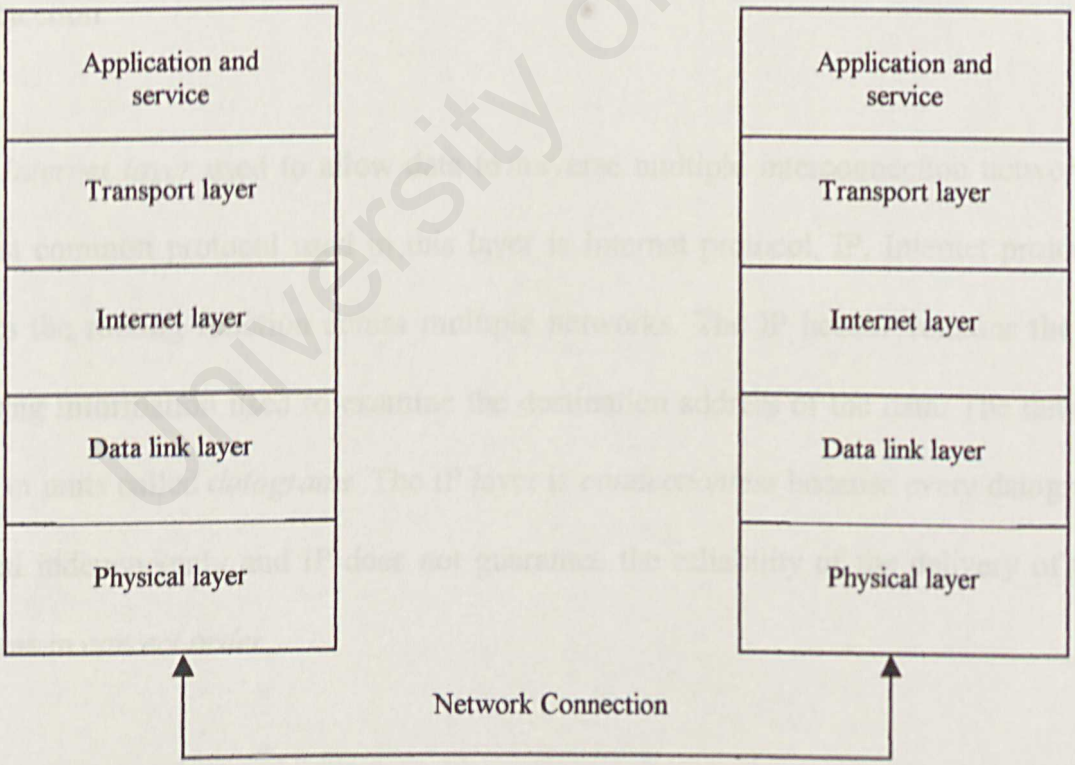


Figure 2.4 Data flow through layers of protocol over the network

Physical layer covers the physical interface between a data transmission device (e.g. workstation and computer) and a transmission medium or network. This layer is concerned with specifying the characteristics of the transmission medium, the nature of the signals, the data rate and related matters.

At *Data Link Layer*, the data is organized into units called *frames*, which have a header and a trailer. The header consists of the address and control information. If the data transmission is within a Local Area Network (LAN), the addresses are the identifier of the source and destination network interface card. If the transmission were over the LAN, the address would be the identifier of the frame relay circuit. The trailer is used for error detection.

Internet layer used to allow data to traverse multiple interconnection networks; the most common protocol used in this layer is Internet protocol, IP. Internet protocol provides the routing function across multiple networks. The IP header contains the IP addressing information used to examine the destination address of the data. The data is carried in units called *datagrams.* The IP layer is *connectionless* because every datagram is routed independently and IP does not guarantee the reliability of the delivery of the datagrams in *correct order.*

The *TCP (Transmission Control Protocol)* is used in the *transport layer* because it provides reliable data connection services to applications. It guarantees that data is delivered error-free, which means the data are in sequence, no missing data, and no duplication of data. TCP fulfills this function by adding the header to form a *segment* and examines the header when it is delivered. The segment is the delivered data unit in the TCP layer. TCP is one of the layers that interface with applications. There are also others protocol that implemented in the transport layer like *User Datagram Protocol (UDP)*.

The TCP and IP header is shown below:

<table>
<tr><td colspan="3">Source port</td><td>Destination port</td></tr>
<tr><td colspan="4">Sequence number</td></tr>
<tr><td colspan="4">Acknowledgment</td></tr>
<tr><td>Header length</td><td>Unused</td><td>Flags</td><td>Window</td></tr>
<tr><td colspan="3">Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="4">Option + padding</td></tr>
</table>

Figure 2.5 TCP header

TCP/IP Addressing

There are two kinds of addresses for TCP/IP protocol to identify the machine. One is *domain address* and the other is *IP address.* Both kinds of addresses can identify the machine uniquely. For routing over Internet, the data is guided by the IP address. The domain address can be translated into IP address. In the Internet address family, some addresses are reserved for special purposes. They are *special addresses.*

There is an interesting address 127.0.0.1, called *"loopback"*, which enables client and server to work in the same host. The characteristic of this address makes it easy to develop client/server application in a *stand along computer.*

To connect, the applications must identify each other not only by IP address of the machines they are sit on, but also by the *port numbers* which are used to identify the processes or services. The available port numbers range from 0 to 65,535, of which numbers 0 to 1023 are reserved for well-known services, such as FTP and Telnet. The rest of the numbers are enough for user-defined applications.

The client uses *socket address* (i.e. the combination of IP address and port number) to identify its connection to the server makes it easy for the server to keep track of multiple client connections.

2.4.2 User Datagram Protocol (UDP)

Besides TCP/IP protocol, there is one other transport-level protocol that is in common use as part of the transport layer, the **user datagram protocol (UDP).** UDP is different from TCP because it provides a connectionless service for application-level procedures, which means that it does not guarantee delivery, preservation of sequence, or protection against duplication.

UDP can send messages to other procedures with a minimum of protocol mechanism. The UDP header is shown as below:

Source port	Destination port
Segment length	Checksum

Figure 2.6 UDP header

2.5 Programming Tools

2.5.1 Object-Oriented Programming (OOP)

In an Object-Oriented Programming language, every program's element can be considered as object. Objects are "black boxes" which send and receive messages and if properly used, OOP improves the maintenance, reusability, and modifiability of software. There are three major concepts that are the backbone of OOP which are *encapsulation*, *inheritance*, and *polymorphism*.

Encapsulation

Encapsulation enables the programmer to hide, inside the object, both the data fields and the methods that act on that data. This can help in control the access to the data, because all the object's data is always private to the object and other parts of a program should never have direct access to that data. This is very simple to done by just making the data local to the function. When programmer needs to make the data of one function available to other function, the way to do this is just to make the data global to the program.

As conclusion, encapsulation provides two primary benefits to software developers:

- Modularity. All the source code for an object can be written and maintained independently. Object can be easily passed around in the system.

- Information hiding. Object can act as private or global object. When an object has a public interface with other objects, it can be use to communicate with others, but the object can maintain private information and methods that can be changed at any time without affecting the other objects that depend on it.

Inheritance

Objects are designed in terms of classes. Inheritance is a ability of class to enable programmer to create a class that is similar to a previously defined class, but one that still has some of its own properties. In OOP, programmer can create a new class by inheritance, which means that the new class inherits all the data and methods from the tested base class. This is different from the non-OOP language that programmer might have to modify the existing code extensively and introduce bugs into code that worked fine before your use it. Moreover, programmer can control the level of inheritance with the *public*, *private* and *protected* keywords.

The major benefits of inheritance are:

- Subclasses provide specialized behaviors from the basic of common elements provides by the superclass. Through the use of inheritance, programmer can reuse the code in the superclass many times and save time.

- Programmers can implement superclasses called abstract classes that define "generic" behaviors. The abstract superclass defines and may partially implement the behavior but much of the class is undefined and unimplemented.

Polymorphism

Polymorphism is the last major feature of OOP language. By using polymorphism, programmer can create a new objects that perform the same functions as the base object but which perform one or more of these functions in a different way. For instance, programmer has a shape object that draws a circle on screen, with polymorphism; programmer can create a shape object that draws a rectangle instead.

2.5.2 Java

Java is the latest, flashiest object-oriented language. It has taken the software world by storm due to its close ties with the Internet and web browsers. It is designed as a portable language that can run on any web-enable computer via computer's web browser. With a compiler, Java program can be translated into an intermediate language called Java byte codes.

Java application can be divided into 4 types, mini application, application, line application and libraries.

- Mini application

  Mini application also called Applet. Applet is a program embedded in a web page, which is downloading over the Internet and run in a Java enable browser like Internet Explorer or Netscape.

- Application

  Graphical User Interface application (GUI application) is an application that not requires any window browser.

- Line application

  Line application is an application that runs on MSDOS command prompt or UNIX shell prompt.

- Libraries

  It is similar to C++ class library. It cannot define as an application. It is more a collection of class that belongs to one package.

2.5.3 Virtual Basic

Visual Basic is a Microsoft Windows programming language. It is derived from the BASIC programming language and the BASIC language is a language used by more programmers than any other language in the history of computing. Virtual Basic gives the enhance features in graphical user interfaces design, event handling, error handling, access to the Win32 API, structured programming and much more.

2.5.4 Smalltalk

Smalltalk is a pure object- oriented language. It uses run-time binding, which means that nothing about the type of an object need be known before a Smalltalk program is run. Smalltalk programs are considered by most to be significantly faster to develop than C++ program. A rich class library that can be easily reused via inheritance is one reason for this. Another reason is that Smalltalk's dynamic development environment. It is now explicitly compiled, like C++.

2.6 Summary

TCP and UDP are the most commonly used transport protocols on the Internet. UDP is chosen as the target transport protocol for RTP because of the following reasons:

- RTP is primary design for multicast. TCP is connection-oriented and does not scale well for the multicast.
- Timely delivery is the main concern issue for real-time activities, it is even more important from the reliability (provides by the TCP/IP protocol). For instance, when there is network congestion, if the protocol insists a reliable transmission, the retransmitted packets could possibly increase the delay.

Applications typically run RTP on top of UDP as part of the transport layer protocol, as shown in below:

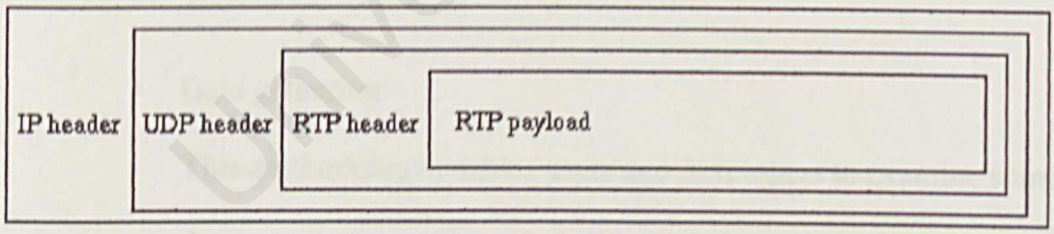


Figure 2.7 RTP data in an IP packet

Besides, Java is chosen as programming tool and prototyping methodology is chosen as the whiteboard methodology. For the further explanation, please refer to chapter 3, system analysis.

CHAPTER 3 SYSTEM ANALYSIS

3.1 Methodology

Prototyping methodology is chosen for the real-time activities, whiteboard because the prototyping methodology is easier to repeat the investigation of the system requirement or the design of the system. When the users are not able to state out all the requirements for the system at the beginning of the system, this approach allows users to get more involve in system, thus, system developer can seek user's reactions, suggestions, innovations and revision plans in order to make improvement to the prototype and system. Since the whiteboard is relatively smaller in size, it is encourage using the prototype methodology in system development.

Besides, whiteboard system is suitable to use the prototyping methodology because of following advantages: -

- Data gathering

  This methodology enables users and developers to examine some aspect of the proposed system and decide if it is suitable or appropriate for the finished product. It allows the developers to seek user's reactions, suggestions, and innovations too.

- Development cost relatively low

  Prototyping methodology can reduce the development cost because users have directly or indirectly involve in the system development, this make less effort in data gathering for system developer. Furthermore, users sure will satisfy with the system and reduce the risk of misunderstanding the requirements stated by users.

- End-user system

  Whiteboard is an end-user system, where users' requirement is very important in developing the system. It is encourage users can involve along system development to ensure users current requirement is fulfill.

- Never experience in similar design

  When there are no experience in develop the similar system, it is encourage to use prototyping for reduce the system's risk.

- Relatively smaller in size

  Whiteboard is a simple program. It changes easily if we use prototyping methodology to develop. With prototyping methodology, the whiteboard is developed rapidly too.

Research plan for the project is categorize into 5 components as below: -

- Data gathering

  In this phase, the data or information that required by the project is collected through various techniques. Interviews, and Internet surfing is among the techniques of requirement gathering used in this project. Internet is an information bank that provides a lot of information developer need. Through the Internet, developer can make comparison among the existing products. *Frequency question answer (FAQ)* is another alternative way to collect the useful data while developer surf net.

- Analysis phase

  In this phase, scope of the project is determined and the tools to use are finalized after analyze all the requirement and information.

- Designing phase

  The designing phase completes all the design specification. Data flows of the system are finalized in this phase. Definition of the system structure, interface design, data structure, and database design must come out too.

- Implementation phase

  Basically in the implementation phase, the only thing is to do the coding and the implementation of the system.

- Testing and evaluate phase

  This phase leads to the testing of the complete system.

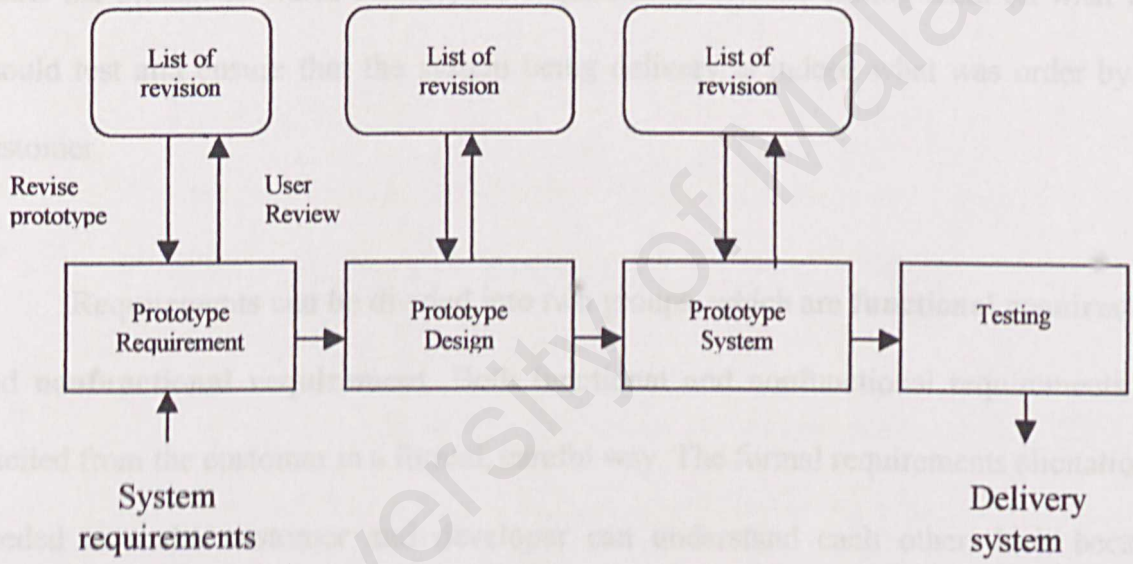


Figure 3.1 Prototyping model

3.2 System Requirement

Requirement describes a system's behavior. It describes not only the flow of information to and from system and the transformation of data by the system, but also the constraints on the system's performance. The requirements serve for three main purposes. Firstly, they tell the developer on what functionality and features the final system should have. Secondly, they allow developer to explain their understanding of how the customer wants the system to work. Finally, the requirements tell the testing team on what they should test and ensure that the system being delivery is indeed what was order by the customer.

Requirements can be divided into two groups, which are **functional requirement** and **nonfunctional requirement**. Both functional and nonfunctional requirements are elicited from the customer in a formal, careful way. The formal requirements elicitation is needed to make customer and developer can understand each other. It is because sometimes customers are not good at describing exactly what they want and developers are not always good in understand customer's requirements, so, a formal requirement elicitation is needed.

3.2.1 Functional Requirement

A functional requirement describes an interaction between the system and its environment. It describes on how the system should behave given certain stimuli.

The whiteboard module can be divided into 4 sub modules, there are drawing module, text module, colour module, and manipulation module.

- Drawing Module

  Drawing module consists of 3 minor parts, there are freehand drawing, basic shape drawing and eraser tool.

  - Freehand Drawing

    Users can draw anything on the whiteboard by mouse. This sub module is used to manage users' freehand drawing.

  - Basic Shape Drawing

    Whiteboard provide some basic shape for users to choose, including circle, triangle, rectangle, polygon and so on.

  - Eraser Tool

    Eraser tool is use to enhance users drawing. The whiteboard will not provides the "undo" function for the users, therefore, if there are any mistakes on the drawing, users must use eraser tool to correct it.

- Text Module

  Text module is use to manage the text editing for the whiteboard. For instance, this module will decide how many fonts will provide by the whiteboard to the users.

- Colour Module

  Colour module will focus on the colour templates of the whiteboard. There is a linkage between text module, drawing module and colour module.

- Manipulation Module

  This module is divided into object manipulation and drawing area manipulation.

  - Object manipulation

    Object manipulation lets users to edit the drawing object. For instance, users can select a small area on the whiteboard and edit. Editing means users can either delete the selection or copy the selection.

  - Drawing area manipulation

    Users can change the whiteboard's background colour themselves. If users do not like to look at the whiteboard, the users can just disable the whiteboard. All of this is done in the drawing are manipulation.

3.2.2 Nonfunctional Requirement

Nonfunctional requirement describes a restriction on the system that limits our choices for construction a solution to the problem.

- Reliability

  System will not produces any dangerous when it is used in a reasonable manner, which means in a manner that a typical user expects is normal. In the others word, reliability is referred to the expectation of a system to perform its intended function accurately. Whenever a button is clicked, the system should be able to execute that particular function or generate some message to inform the user what is happening.

- User- friendly

  The system should be user – friendly. All the features and functionality provide by the system should be easy to use and can be understood by users from any level.

- Platform independence

  This system should be platform independence which means the system can be run or execute in any platform including Window 95/98, Windows NT/2000, Linux, Unix, Macintosh, Solaris and so on. All the settings and information are consistent in any platforms.

- Reusability, expandability and maintainability

  The coding for the system must be flexible for expandability and maintainability. The system can be reuse for the purpose of version upgrade.

- Security

  The system must be away from the crisis of being corrupted.

3.3 Feasibility analysis

After consider all the alternative way and solution to build the whiteboard system, developer will analyze the feasibility for the system and choose the best choice to complete the system. Below will explain the advantages of using the Java programming tools.

3.3.1 Programming Language – Java

Java is a programming language that introduced by sun Microsystems in June 1995. Java is an object-oriented, robust, secure, portable, high performance, platform independence, and multithreaded language.

Everything is considered as a class in Java regardless of what it does. Java bean, Java Applet and Java application are all Java classes. Java shown the explosive growth in popularity because the design and implementation of Java allow compiled program to be shipped across the internet to run on the client system and the simplicity of the language when compare with other language like C++.

Java provides a lot of benefits that are very useful in develop the whiteboard system. The benefits show below.

- Platform independence

  Java provides write-once, run –anywhere philosophy. This means if the system is developed by Java language, it can be run in any platforms, for example Windows, Linux, Unix and so on.

- Multithreaded

  Multithreaded occur when an operating system run more than one task simultaneously. This is a very useful feature in develop an real time activities system like whiteboard because it can increase real time performance and make real time activities much more exacting. With multithreaded feature, system can prioritize its threads so that threads of low important are delegated to a low-priority mode. Java can run multiple tasks within the context of one large application because of multithreaded.

- Dynamic

  Java can adapt to an evolving environment. It allows for new instance variables and methods in a library object without affecting the dependent client object. Java also can relieve the dependency of complete reconciliation when a parent class is changed. Java also easy to make

adjustment and even from across a network, Java's dynamic abilities will permit the loading of class.

- Secure

  There are 3 features in Java's runtime environment that make the system secure:

  - Runtime memory layout
  - Byte code verifier
  - File access restrictions

- Distributed

  Client – server application are said to by distributed. They distribute their processing workload as well as their information for sharing and collaboration. Because of Java supports application on networks, then it can define as a distributed language.

3.4 Project Schedule

	JUN 2001	JULY 2001	AUG 2001	SEPT 2001	OCT 2001	NOV 2001	DEC 2001	JAN 2002	FEB 2002
Requirement Elicitation									
Project definition & Scope									
Conceptual Design									
Literature Review									
Analysis Phase									
Requirement Analysis									
Determining System									
Designing Phase									
Database Design									
Interface Design									
Coding									
Implementation Phase									
Testing & Evaluation									
Integration Phase									
Documentation									

Table 3.1 Project Schedule

3.5 Hardware and Software Requirement

3.5.1 Development environment

- Pentium II 400 Mhz PC
- 256 MB SDRam
- 20 GB hardisk
- 44x CD-ROM
- 16 MB Voodoo3 2000 VGA card
- Creative sound blaster live! Value sound card
- Standard mouse and keyboard
- 56K Aztech external modem
- 265SP Canon BJC Printer
- 17 inch Samsung SyncMaster Monitor
- Java SDK 1.3
- Jbulder IDE
- Jcreator IDE
- Java Web Server
- Internet Explorer 5.5
- Windows 98 Second Edition
- Windows 2000
- Notepad

- MS SQL 7.0
- Microsoft Access
- IIS and Personal Web Server
- Java Virtual Machine

3.5.2 Runtime Environment

- Pentium II and above
- 64 Ram and above
- 6.4 GB hardisk
- Sound card
- 8 MB VGA card
- Standard input and output device
- Java Virtual Machine
- Internet Explorer 4.0 and above
- Any operating system

Chapter 4 System Design

Introduction

System design is a very important step in the system development because it determines how success is the system. This phase will try to translate all the problems to the solution. Requirements that are found in analysis stage are the one actually translated into design specification. System design will cover two major parts, there are conceptual design and technique design.

4.1 Conceptual Design

Almost all the clients cannot understand the programming language. They paid to the developer due to design a suitable system for them. Conceptual design means to describe the system in language that the customers can understand. The developer tells the clients how the system works by the command language.

AceChat whiteboard system provides a drawing area for the users to post their drawing and all the users at the same session can see their drawing. It is an enhancement feature for chat room. A set of drawing tools is prepared to ease the users. Besides, the system will also contain the tools preference such as drawing thickness or size, and fill type. *AceChat whiteboard* can let users to select their preference colour to draw or fill in too.

4.2 Technique Design

- **Log in and log out**

  Users must login to the *AceChat whiteboard* system before use it. Some necessary information such as nickname, password, port and server name need to be key in while we want to logging in to the system. Users need to log out also when they want to leave the system.

- **Drawing Line**

  Users can draw a straight line on the drawing area. When the users click on anywhere on the drawing area and drag the mouse, a draft line will be shown. Users can confirm to draw a line on the drawing area by release the mouse. All the users in the same session can see this line real time.

- **Freehand Drawing**

  Freehand drawing tool let the users draw anything according to their wish. Usually users like to use this drawing tool because it is no limitation. This means that with this tool, users can draw rectangle, circle, text, line or anything into the drawing area but the drawing will not smooth enough if compared with the specify drawing tools.

- **Object Drawing**

  AceChat whiteboard system provides the basic shape module. Object such as circle, rectangle and 3D rectangle is prepared. Users can select the shape and drag it into the drawing area. Before the users confirm the drawing, a draft drawing will be shown.

- **Text Drawing**

  Different from the text chatting, the text drawing is more flexible. Users can selected their preferred colour, size, font type, font style for the text they input. Before the users decide where to paste the text, the text will follow the mouse pointer and users can confirm to draw the text by left click the mouse. The text can be used as description for the drawing.

- **Clear the Drawing Area**

  If the drawing area is no more space to draw, the users can clear the drawing area. All the drawing on the drawing area will be cleared. This action may take effect on all the users at the same session.

- **Erase the Drawing**

  Eraser tool is provided due to improve the drawing. Since the whiteboard did not provide *undo* and *redo* function, eraser tool is important for the users to correct or improve their drawing.

- **Change the Drawing Colour and Background Colour**

  Users can select their preferred colour to draw. A window will pop up for the users to select the colour through *RGB mode* or *HSB mode*. Besides, users can change the background colour for the drawing area too. When the user confirms the transmission, the background colour for all the users at the same session will change.

- **Change the Drawing Tools Preference**

  Every drawing tool has its own preference. For example, users can select different thickness for the line drawing, freehand drawing and object drawing. Users can change the colour for the drawing tools. The *fill* or *outline* style is provided for the object drawing tools to make it more flexible.

4.3 Data Flow Diagram

A data flow diagram (DFD) shows how data moves through the *AceChat whiteboard* system. Although DFD do not show program logic or processing steps but it represents a logical model that shows what the system does, not how it does it. Data diagrams. This distinction is important because focusing on implementation issues at this point would restrict your search for the most effective system. There are several versions of DFD symbol exist but all serve the same purpose. The Gane and Sarson symbol set will be used to model the data moving of this system.

4.3.1 Data Flow Diagram Symbols

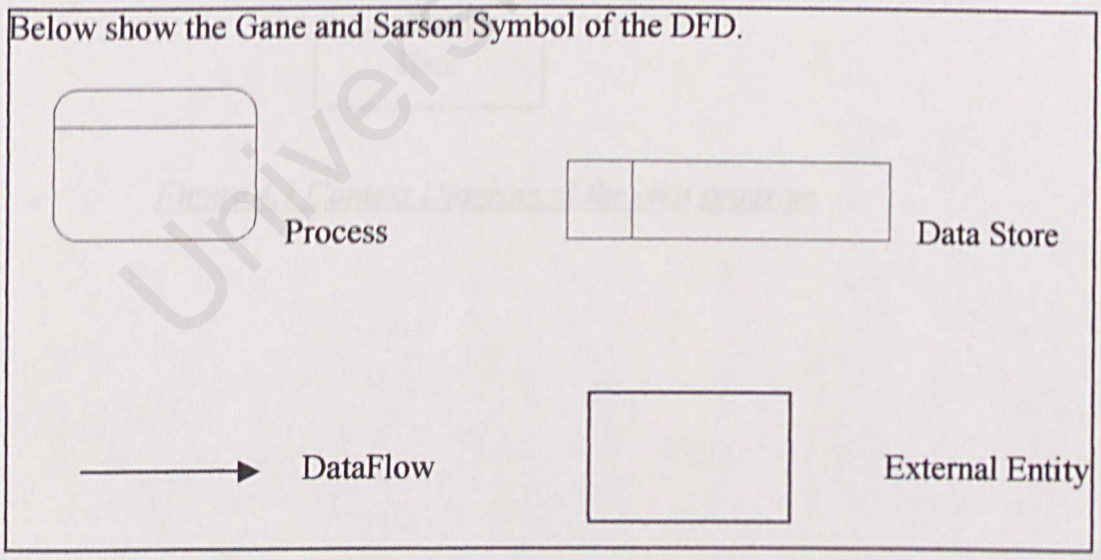


Figure 4.0 Symbol of DFD

4.3.2 Context Diagram

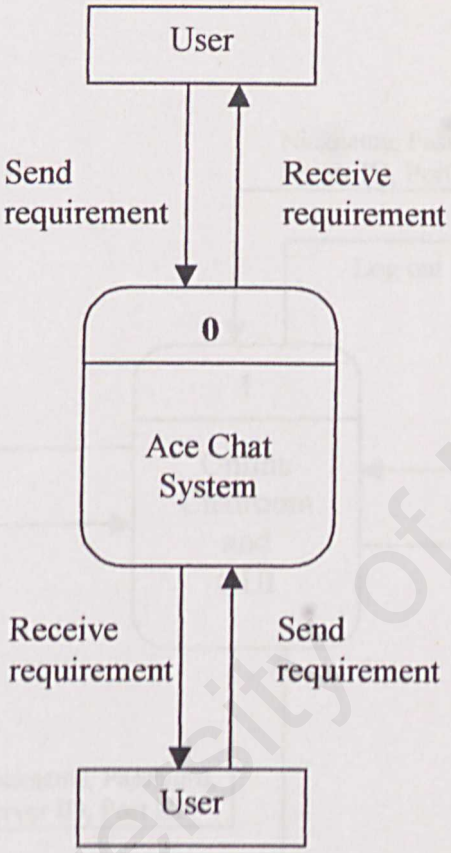


Figure 4.1 Context Diagram of the chat program

4.3.3 Ace Chat DFD – Level One

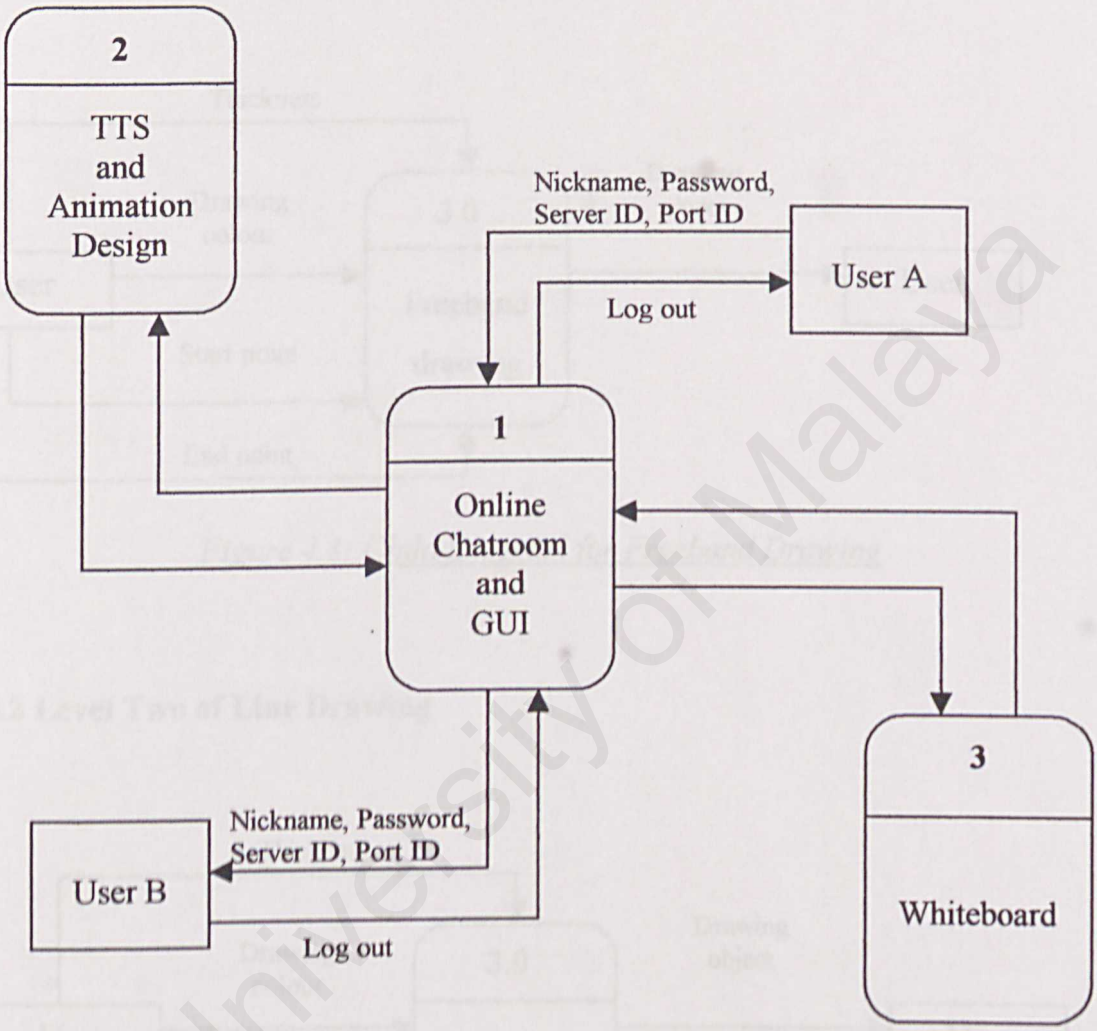


Figure 4.2: Ace Chat Application DFD – Level One

4.3.4 Ace Chat DFD – Level Two (Sub Module: Whiteboard Module)

4.3.4.1 Level Two of Freehand Drawing

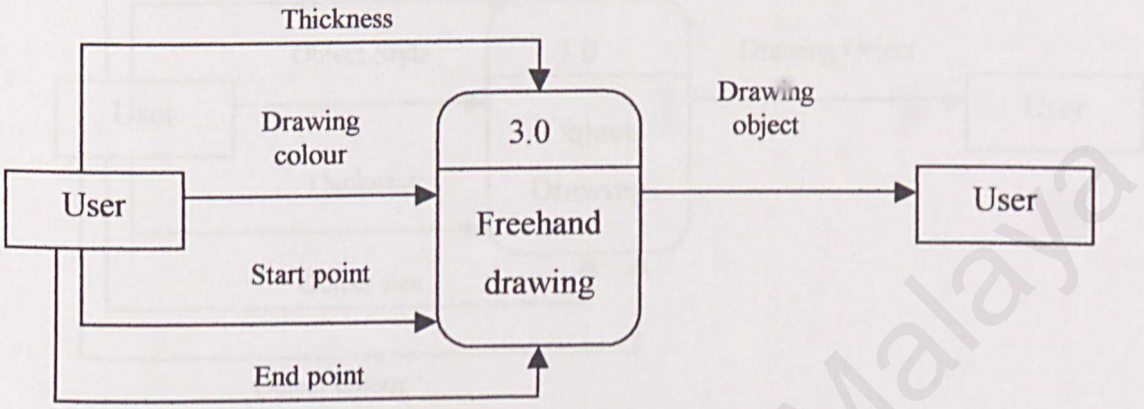


Figure 4.3: Child Diagram for Freehand Drawing

4.3.4.2 Level Two of Line Drawing

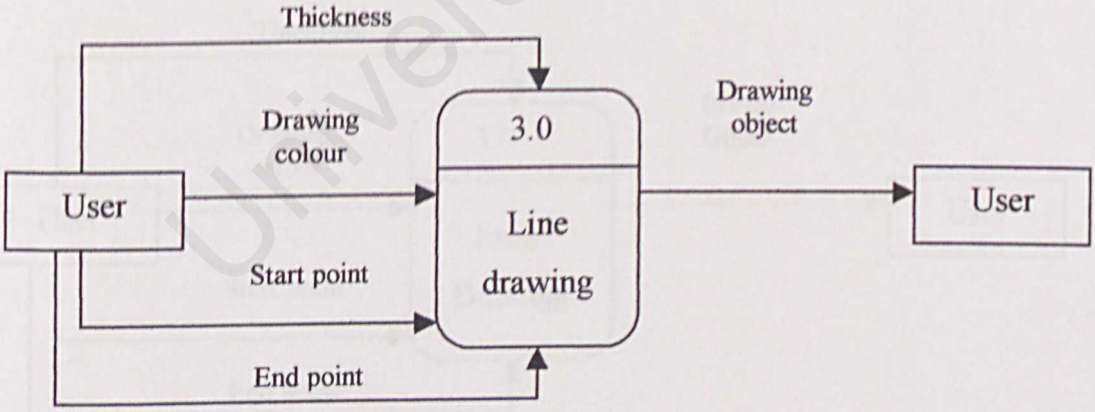


Figure 4.4: Child Diagram for Line Drawing

4.3.4.3 Level Two of Object Drawing

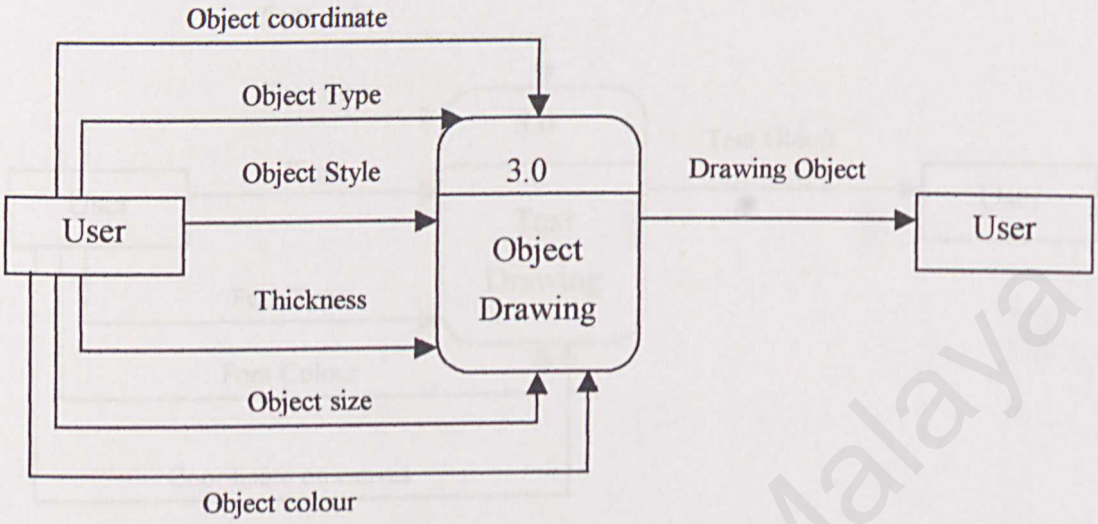


Figure 4.5: Child Diagram for Object Drawing

4.3.4.4 Level Two of Eraser Drawing

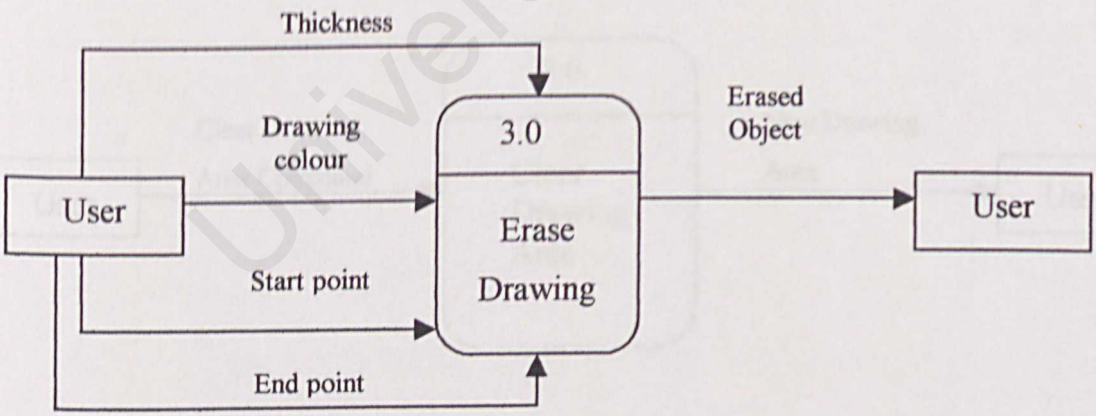


Figure 4.6: Child Diagram for Eraser Drawing

4.3.4.5 Level Two of Text Drawing

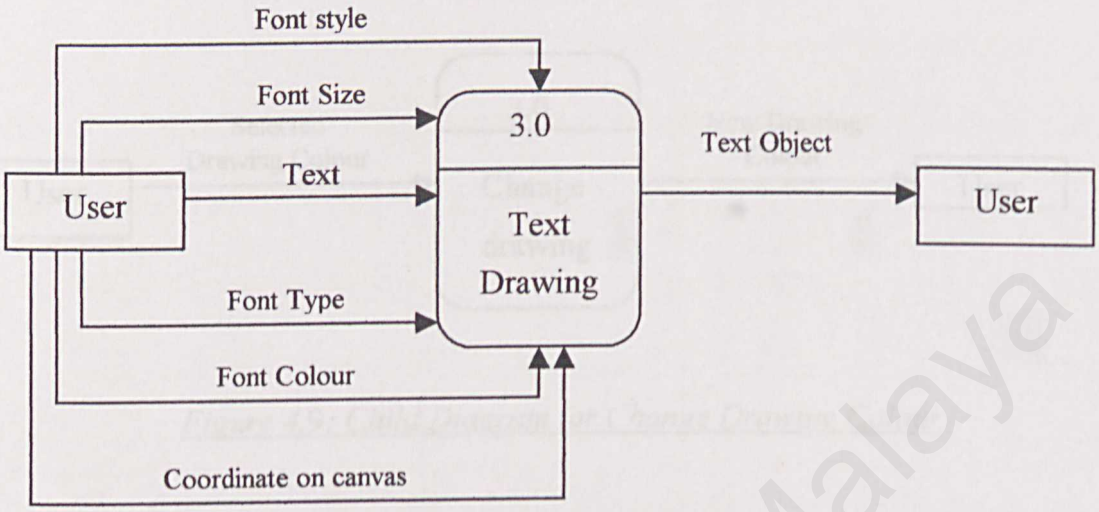


Figure 4.7: Child Diagram for Text Drawing

4.3.4.6 Level Two of Clear Drawing Area

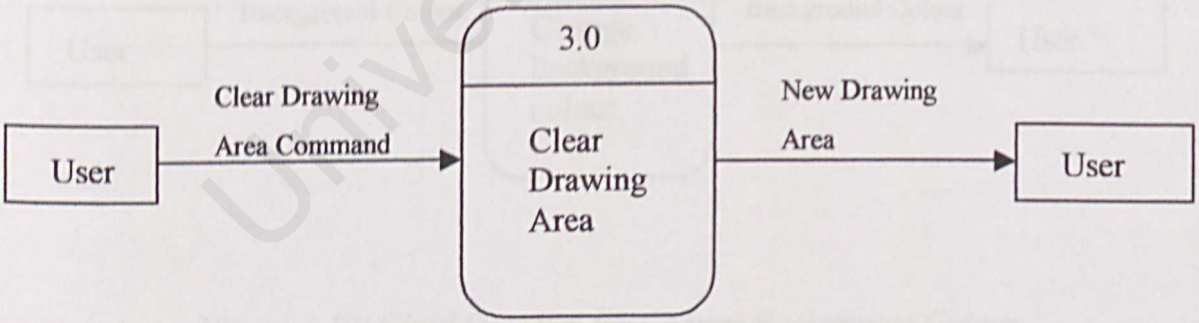


Figure 4.8: Child Diagram for Clear Drawing Area

4.3.4.7 Level Two of Change Drawing Colour

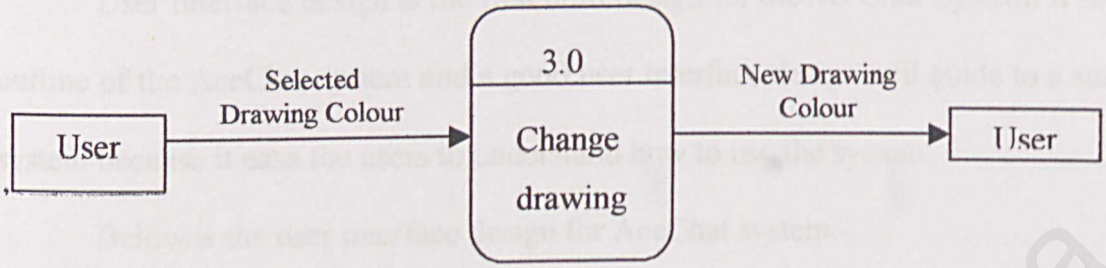


Figure 4.9: Child Diagram for Change Drawing Colour

4.3.4.8 Level Two of Change Background Colour

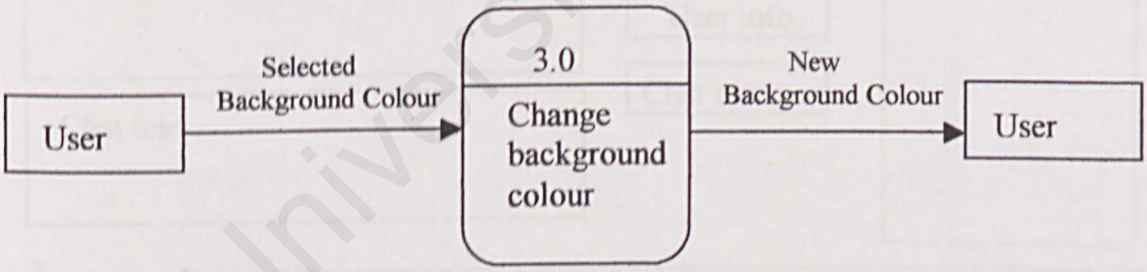


Figure 4.10: Child Diagram for Change Background Colour

4.4 User Interface Design

User Interface design is the first draft design for the *AceChat* System. It shows the outline of the AceChat system and a good user interface design will guide to a successful system because it ease the users to understand how to use the system.

Below is the user interface design for AceChat system.

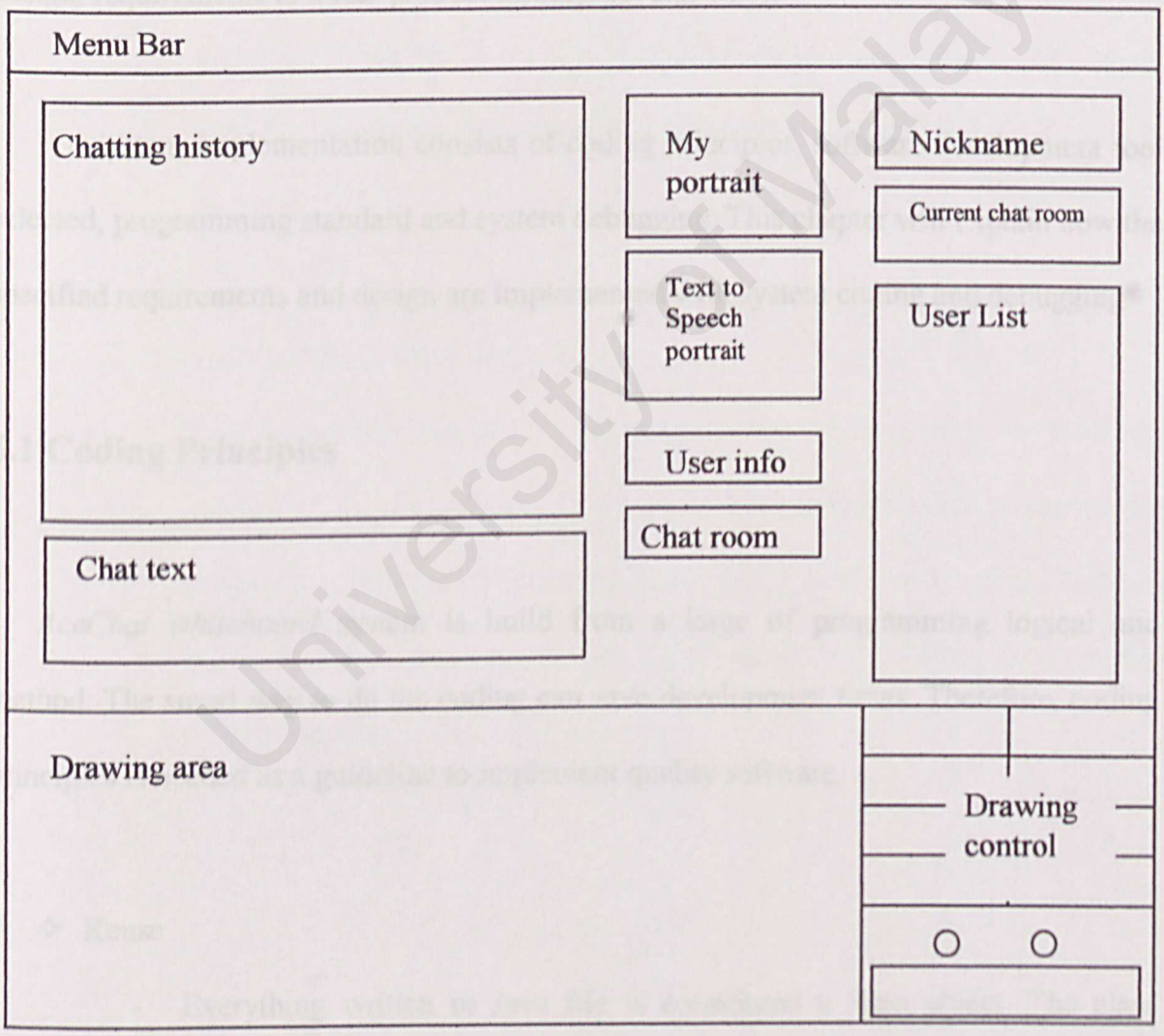


Figure 4.11: User Interface Design for AceChat System

Chapter 5 System Implementation

Introduction

This Chapter will focus on how to implement *AceChat whiteboard* from the beginning to the end. System implementation means how to convert all the conceptual design, requirements to a real product through several ways.

System implementation consists of coding principles, software development tool selected, programming standard and system debugging. This chapter will explain how the specified requirements and design are implemented with system coding and debugging.

5.1 Coding Principles

AceChat whiteboard system is build from a large of programming logical and method. The smart way to do the coding can save development times. Therefore, coding principles is needed as a guideline to implement quality software.

✧ Reuse

- Everything written in Java file is considered a class object. The class object created can be reused in subsequent and related applications. Productivity can be increased not only by reducing coding time, but also

reducing testing and documentation times. Base on this principle, a reusable socket class is build to be reused with least modification. There are a lot of reuse examples in *AceChat whiteboard* system. For instance, a *"line"* class can be use either in *line drawing* or *freehand drawing*.

✧ Readability

- A readable programming style ease future enhancements. By using meaningful variables, label's name and comments, each developer can enhance the system's functionality without verifying all the coding line by line.

✧ Maintainability and Ease of Testing

- Maintainability and testing can be done easier by reduce the coupling between files. This means with a good management of files or classes, the system can separate all the functional classes well. For instance, files can be categories into client side and server side. Thus, testing and maintainability can be done easier.

5.2 Development Environment

Development environment take impact on the development of the system. By using the suitable hardware and software will lead to save the system development times. A part from that, it will also determine the success of the project. The hardware and software tools used to develop the entire system are as below:

5.2.1 Hardware Requirements

The hardware used to develop the system are listed below:

- Pentium II 400 MHz computer
- 384 MB RAM
- 20 GB Hard Disk
- Other standard PC component

5.2.2 Software / Software Tools Requirements

Software /Software Tools	Purpose	Description
Microsoft Window 98 Second Edition	System requirement	Operating system (OS)
Java Software Development Kit 1.3 Standard Version	System requirement	Java Virtual Machine
JCreator	Software development tool	Java Technology's IDE
Adobe Photoshop 6.0	System development	Button and Splash Screen Design
Microsoft Access 2000	System development	Database Server (Build the database to store and manipulate the data)

Table 5.1: Software Tools Requirement

5.3 Software Development Tool

Java files need a virtual machine to run. Therefore, *software development kit* needs to be installed into the operating system to run the java files. For *AceChat whiteboard* system, *Software Development Kt (SDK) 1.3 Standard Edition* is used.

JCreator is the software development tool selected to implement the AceChat System. *JCreator* is a *Freeware* and designed for Java Technology. It is a very powerful

and user – friendly IDE, developer can compile and run the program in this editor instead of run it at *MS DOS prompt*. The *JCreator's* interface is shown in *figure 5.1*.

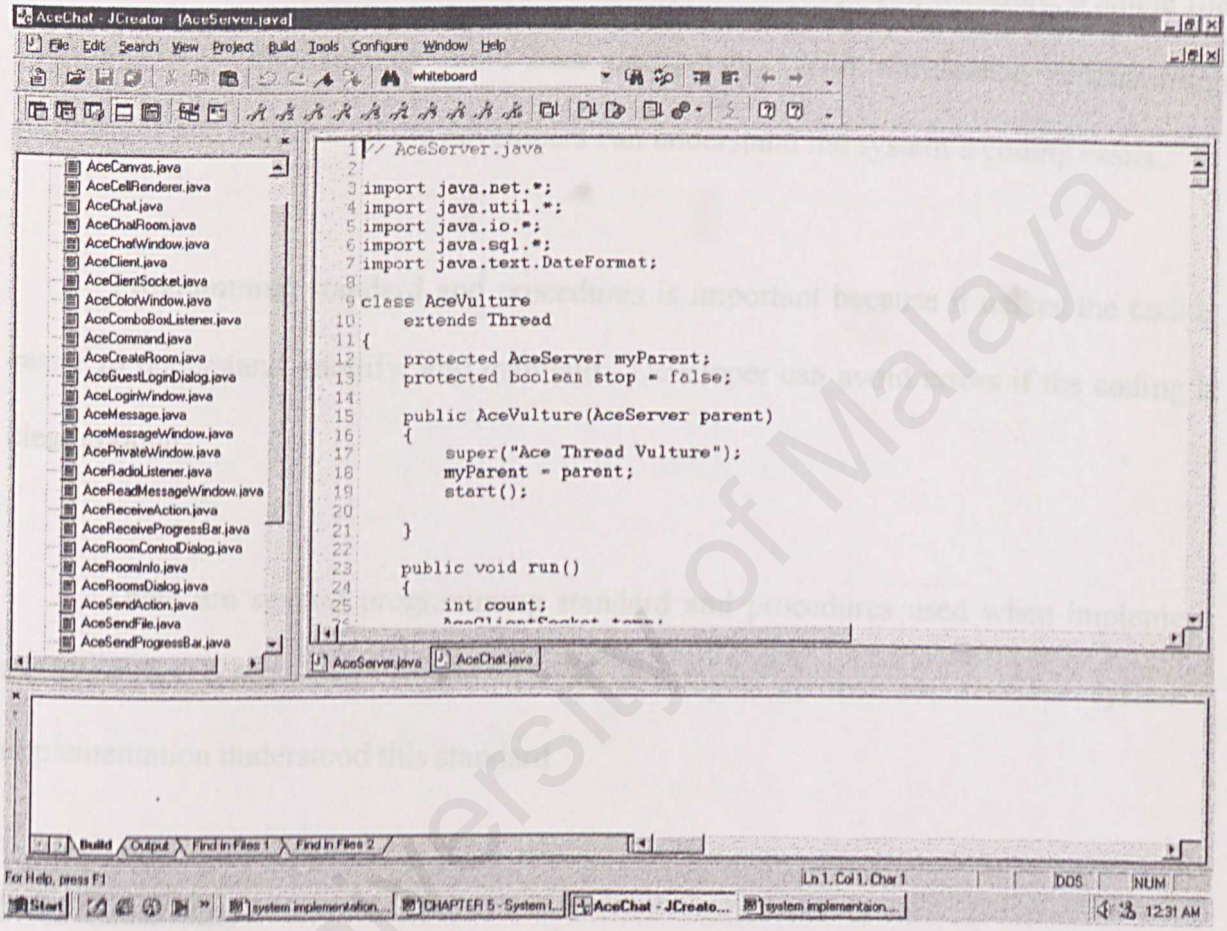


Figure 5.1: The JCreator IDE Interface

5.4 Programming Standard and Procedures

More than one person involve in develop *AceChat* system, therefore, a single file may access or modified by more than one person. With standardize programming procedures and method, all the developers can understand the system's coding easier.

Programming standard and procedures is important because it makes the coding easier to understand, modify, and maintain. Developer can avoid errors if the coding is clear enough.

There are several programming standard and procedures used when implement the *AceChat whiteboard* system. Every developer involve in *AceChat* system's implementation understood this standard.

5.4.1 Comment

Comment is used when developer wants to add remarks for some coding. Comment statement can be divided into *header comment block* and *internal programming comment*.

5.4.1.1 Header Comment block

Header comment block is adding at the top of the file. It describes the general information of a file like file name and the purpose to create this file. Sometimes, there will contain the information like date create, creator's name, date modify and so on.

```
/*

file name: AceColorWindow.java
creator  : ng yih young
date     : 3/1/2002
description: this file is used to return a color chooser window
             when background color button or drawing color button
             is pressed.
*/
```

5.4.1.2 Internal Programming Comment

Internal programming comment is adding to the source file for make the coding clearer. It is insert between the coding. It is very useful because it lets the developer keep track of what they have done before without any confuse.

```
public void mousePressed (MouseEvent e)
{
        oldx = e.getX();  // get the coordinates x and y
        oldy = e.getY();

        if (showingPicture)  // if it is first time login to the system, clear the pics!
                clear();
}
```

5.4.2 Meaningful Variables and Labels' Name

A lot of variables and labels' name is declared for a big system like *AceChat* System. Meaningful variables and labels' name is important because it will ease the developer to follow up. Below shows some meaningful declaration for variables.

```
protected JMenu whiteboardMenu;    // declared submenu under
                                   //whiteboard menu
protected JMenu toolsSubMenu;
protected JMenuItem menuFreehand;
protected JMenuItem menuLine;
protected JMenuItem menuCircle;
protected JMenuItem menuRectangle;
protected JMenuItem menuRectangle3d;
protected JMenuItem menuText;
protected JMenuItem menuEraser;
protected JMenuItem menuClear;
protected JRadioButtonMenuItem menuFill;
protected JRadioButtonMenuItem menuOutline;
protected JMenuItem menuBackgroundColor;
protected JMenuItem menuDrawingColor;
protected JRadioButtonMenuItem menuEnable;
protected JRadioButtonMenuItem menuDisable;
```

5.5 System Implementation

5.5.1 Object Oriented Programming (OOP)

Java, an Object Oriented Programming Language (OOP), is needed to implement *AceChat whiteboard system*. OOP is different from traditional procedural programming styles, and it is real world much more closely than conventional programming methods. Every file created by Java consider a class, therefore, it can be reuse. For every class, there will be assigned different attributes and methods. These definitely save a lot of development times if wholly program is design and implements by Java.

5.5.2 Graphical User Interface (GUI) design

Graphical User Interface (GUI) design is the first step when programmer starts to develop the system. An attractive GUI design is needed for every program. Thus, a *swing style* GUI design is select to be used while develop *AceChat whiteboard* system. The *swing style* GUI is provided by Java programming language as one of its powerful class.

The interfaces that provided by *Java swing* can easily recognize from interface that provided by others language. With *Java swing*, developer can set the interface's layout. Moreover, Java swing provides a lot of components due to ease the work of GUI design, for instance, *JText* Area component, *JComboBox* component and so on.

5.5.3 Java Files

AceChat whiteboard system is a *client-server system*. All the Java files is separated into client side and server side base on its functionality. Multi- threading is used to implement this system. This programming method allows programmers develop efficient and robust server that processes separate client connection in separate threads. In the others hand, means that every client socket will connect through its own thread. This is useful to insulate different client handlers from each other and to insulate applications from communication failures.

Below are the files created for *AceChat whiteboard* system and its briefly descriptions.

5.5.3.1 Building Client

File name	Description
AceCanvas.java	• To create all the drawing object classes like circle class, rectangle class and line class. • To provide a *double buffering* drawing area. • To handle any drawing event.
AceChatWindow.java	• To build an Interface for all the drawing button and drawing area. • The main console of the AceChat program • Handled users connection to the server
AceClient.java	• To create a connection to the server side. • To send and received all the drawing events to or from the server side.
AceColorWindow.java	• To build the *Colour Chooser Window*. • To handle all the events relevant to colour module.
AceComboBoxListener.java	• To keep track of what user select from the combo Box. • To handle events of drawing thickness.
AceCommand.java	• Class file that contain all the *static declaration variables* that used in the client – server connection.
AceRatioListener.java	• To handle the event of fill or outline check box button.

Table 5.2: List of Client Side Classes

5.5.3.2 Building Server

File name	Description
AceServer.java	• To build the server of AceChat system. • To handle the client who login to the server by assign them with a socket – AceClientSocket for transporting command.
AceClientSocket.java	• Class file that created by AceServer.java when a new user login to the system. • To create a connection to the client side. • To send and received all the drawing events to or from the client side • Handled user's request on certain activity

Table 5.3: List of Server Side Classes

5.5.4 Class Files

As what mention previously, all the Java files consider a class or object. After a Java file was successfully compiled, a class file was automatically generated in a same directory of source file. A program cannot run or execute without a class file (*.class*) but program can execute without source file (*.java*).

5.6 System Debugging

System debugging is a necessary part of the development process because there are no perfect system, every system have bugs (errors). Bugs was a natural part of the development process because of developer's logical error or others limitation. Thus, debugging is done in order to track and correct program bugs before a system is launch. When a test case uncovers an error, debugging is likely the process that results in the removal of the error.

5.6.1 Exception Handling

- Exception handling is a feature provide by java. It is a useful *prevention debugging mechanisms* and responding to unexpected events at runtime. Potentially troublesome codes are enclosed within a *try clause*. Another piece of code is needed in a corresponding *catch clause* that responds to errors cause by the code in the *try clause*. When there is a problem or error occurs in the *try clause*, then the exception will catch in the *catch clause*, this is known as an exception handling method. Below is an example of exception handling.

```
try {
            theClient = new AceClient( tmphost,
                        tmpname, tmpport, this);
}
catch (UnknownHostException a) {
            JOptionPane.showMessageDialog( this,
                                    "Couldn't find the
                                    server",
                                    "Connection error",
                                    JOptionPane.ERROR_MESSAGE)
                                    ;
      return;
}
catch (IOException b) {
            JOptionPane.showMessageDialog( this,
                                    "Couldn't find the port",
                                    "Connection error",
                                     OptionPane.ERROR_MESSAGE);
       return;
}
```

The *try clause* in the coding above is trying to open a connection to the server side. When there is error occurs, exception will be *catch* as *unknownHostException* or *IOException* and an error message will displayed.

5.6.2 Hidden Member variables

- Object Oriented Programming Language may cause hidden member variables sometimes. A hidden member variable exist when user are declare a new variable of the same name but not in the same file. Thus, the code design in this project implicitly prohibits variables hiding. For an example, the *AceChatWindow.class* declares a variable *fontType* for represent the font type selected in text chat. The *AceCanvas.class,* which is derived the *AceChatWindow.class* also declares a variable *fontType* to represent the font type selected in text drawing method. Developer must take notes when this is happen because it may cause confuse.

5.7 Bug Detection

Developer can do the bug detection during the system implementation. The simple way is print out some value as an *output* in the MSDOS prompt. *System.out.println()* method is used in Java to print out something needed. Bug detection is important during the *implementation phase* because it can save the time of *system testing phase.*

5.8 Java Debugger

Java debugger is a tool provide by some Java IDE, like *JBuilder*, to debug the system's coding. With java debugger tool, the debugging procedure is easy to do because it traces the coding line by line to check out the error. Developer can add the *breakpoint* into the coding and watch the *immediate result windows* too. It eases the debugging procedures.

Chapter 6 System Testing

Introduction

To produce quality software – software with fewer error and which work according to specification and performance requirements, system need to be tested. The purpose of testing is to detect the errors that have not been discovered yet .All tests should be traceable to customer requirements too. That's means the software must meet all the requirements of the customer. Test data are being input into the system for processing and the results examined. A number of users are given the opportunity to try the system so as to trace any uncover errors or misunderstandings before launch the system.

6.1 Testing Strategies

Testing strategy is a strategy of establishing the existence of errors. It is also a general approach to the testing process rather than a method of devising particular system or component tests.

The testing strategies consists of following:

- Top-down testing

  Testing starts with the most abstract component and works downwards until all the modules are tested.

- Bottom-up testing

  It is one of the popular approaches used to test large systems. Testing starts with the fundamental components and works upwards

- Back-to-back testing

  Used when versions of a system are available. The systems are tested together and their outputs are compared.

- Thread testing

  Used for system with multiple processes where the processing of transaction threads its way through these processes.

6.2 System Testing

System testing can be divided into unit testing, integration testing and system testing. Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case has a high probability of finding an undiscovered error from the system.

For AceChat whiteboard system, we use *white box* for *unit/ module testing* and *bottom – up testing* for *integration testing*.

6.2.1 Unit Testing

The primary goal of *unit / module testing* is to confirm that the unit is correctly coded and that it carries out the functions it is suppose to carry out. It is the initial testing stage for the completion of each component class. All the logical error that contain in the classes will be detected. The interaction testing between components are initially avoided and to be carried out later in the bottom-up integration testing.

The *White Box Testing* can ensure that lines of codes are examined one by one. White box testing focuses on the idea of coverage, as measures of how much of a module or system has been exercised or executed by a test case or a series of test cases. There are several criteria of coverage:

- Statement coverage
  - Every segment / statement should be covered at least once.
- Branch coverage
  - Every decision should be tested at east once.
- Path coverage
  - Every path should be tested at least once.

The *table 6.1* explains how the *white box* approach is applied to the *AceChat whiteboard* system.

Coverage	Explanations
Statement Coverage	Each line will be evaluated to both true or false. For example:

```
1   public void actionPerformed(ActionEvent E)
2   {
3       if (E.getSource() == bFreehand)
4       {
5           myCanvas.isEraser = false;
6           myCanvas.drawType =  myCanvas.FREEHAND;
7
8           bPrevious.setBackground(buttonColor);
9           bFreehand.setBackground(selectedBtnClr);
10
11          bPrevious = bFreehand;
12          return;
13      }
14
15      else if (E.getSource() == bLine)
16      {
17          myCanvas.isEraser = false;
18          myCanvas.drawType = myCanvas.LINE;
19
20          bPrevious.setBackground(buttonColor);
21          bLine.setBackground(selectedBtnClr);
22
23          bPrevious = bLine;
24          return;
25      }
26  }
```

** bFreehand = freehand button*

** bLine = draw line button*

Above is a part of coding in *AceChatWindow.java* file.

When *'freehaand'* button is pressed, line 3 will return a *'true'* value, otherwise, it will return a *'false'* value. Coding from line 5 to line 12 will execute if 'true' value is returned.

To execute line 17 to line 24 for statement coverage, another test case is prepared, in which to test whether the '*draw line*' button is pressed.

<table>
<tr>
<td>Branch Coverage</td>
<td>Branch testing covers everything that statement testing covers. In branch coverage, every decision should be tested at least once. For example:

<table>
<tr>
<td>1
2
3
4
5</td>
<td>if (!run)
{
this.disconnect();
}</td>
</tr>
</table>

Please notes that the "if" statement above is without "else" block. Branch testing is evaluated the condition a false while statement testing does not. This is the only different between statement testing and branch testing. Sometimes the statement testing can be considered as the sub-testing program of branch testing.</td>
</tr>
<tr>
<td>Path Coverage</td>
<td>Path coverage means every path should be tested or executed at least once. However, it is impossible to execute all possible way when there is a looping in a program. A function with loops may have an infinite number of distinct paths. To deal with some programs, which do not show any looping, we can do path coverage with bounds on loops. Rather than testing all possible paths through loop, one way we can put bounds on the loops is to test a path that does not execute the body of the loop and a path that executes the body of the loop one or more times.</td>
</tr>
</table>

Table 6.1: White Box Approach

Multiple conditions decisions should be tested for all possible simple conditions and outcomes. Every condition within each decision must be evaluated with every combination of *true* and *false* outcomes at some points during test execution. For example:

```
if (floatingText = = true && drawType == this.TEXT){
        drawText (true);
}
else {
        drawText (false);
}
```

Condition 1 = C1 = "floatingText == true"
Condition 2 = C2 = "drawType == this.TEXT"

C1	C2	Expected Outcome
True	True	drawText(true);
True	False	drawText(false);
False	True	drawText(false);
False	False	drawText(false);

6.2.2 Integration Testing

Integration testing is used to ensure that the system will work correctly when all the files combine together. For *AceChat whiteboard* system, *Bottom – Up Integration testing* is selected. . Each component at the lowest level of the system hierarchy is tested individually first. Then the next components to be tested are those call the previously tested ones. This approach will repeat until all the components in the system are tested.

The figure 6.1 shows the *Bottom – Up Integration* testing for *AceChat whiteboard* system. The *AceCanvas.java* file and *AceChatWindow.java* file must tested together first because they show the linkage each other. The lower level testing is base on the client side and the server side, which means the client side files will test together first before trace to the upper level testing.

A = AceCanvas.java	D = AceComboBoxListener.java	G = AceClient.java
B = AceChatWindow.java	E = AceCommand.java	H = AceServer.java
C = AceRatioButtonListener.java	F = AceColorWindow.java	I = AceClientSocket.java

Table 6.2: Reference table

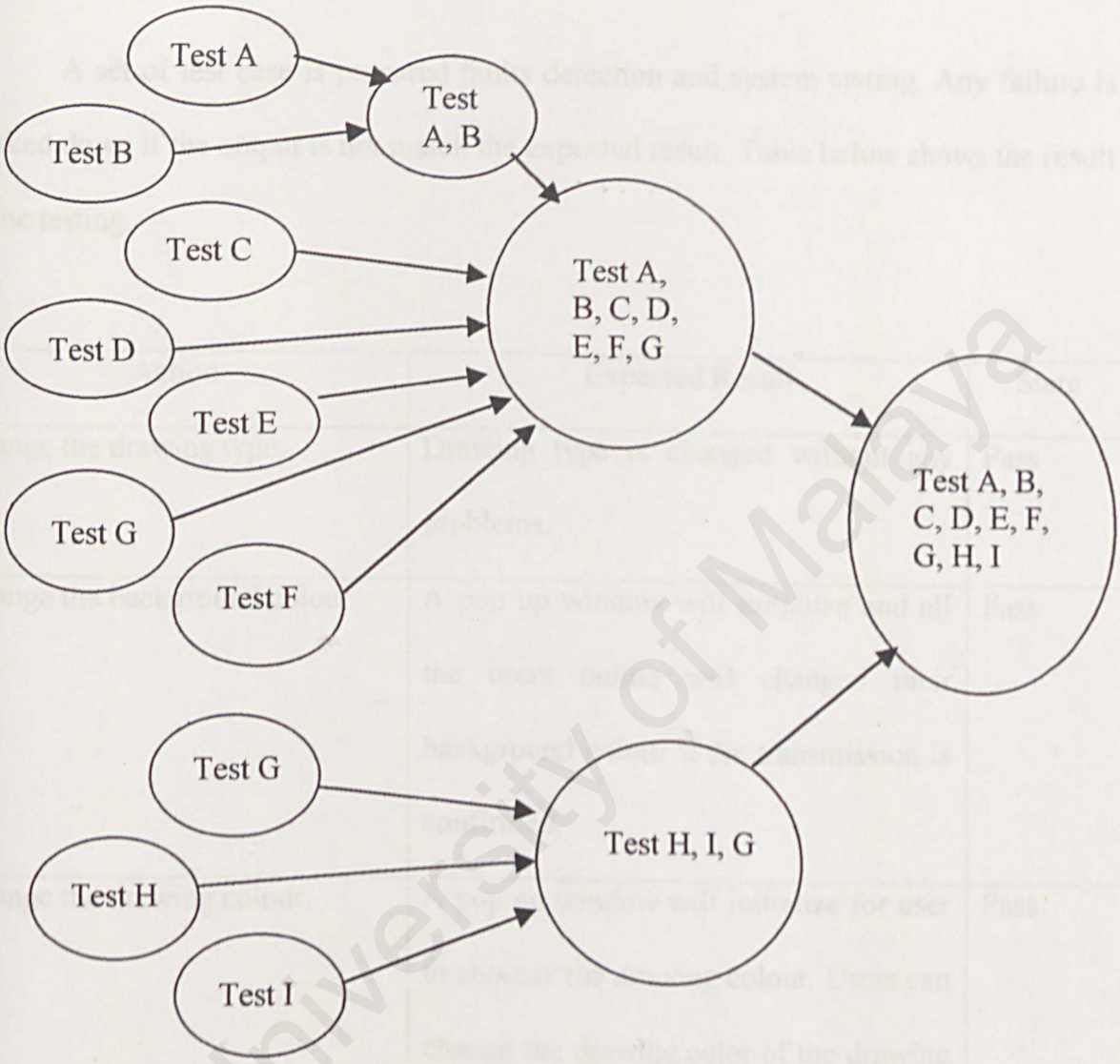


Figure 6.1: Bottom – Up Integration testing

6.3 Test Case

A set of test case is prepared faults detection and system testing. Any failure is marked down if the output is not match the expected result. Table below shows the result of the testing.

Action	Expected Result	State
Change the drawing type.	Drawing type is changed without any problems.	Pass
Change the background colour.	A pop up window will initialize and all the users online will changed their background colour if the transmission is confirmed.	Pass
Change the drawing colour.	A pop up window will initialize for user to chooser the drawing colour. Users can change the drawing color of the drawing tools.	Pass
Clear all the drawing on the drawing area.	All the drawing on the drawing area will be cleared.	Pass
Erase the drawing with eraser tool.	All the users online will see the changes.	Pass

Change the fill type of shape tool.	The shape button will change the icon base on the selection. Basic shape drawing follow the selected fill type.	Pass
Change the drawing thickness.	All the drawing tools change their drawing thickness.	Pass
Drawing the text to the drawing area.	A pop up window will initialize when user click on the drawing area. After confirm the transmission, typed text will follow the mouse pointer and confirmation to paste the text onto the drawing area by single left click.	Pass

Table 6.3 Test Case

Chapter 7: System Evaluation and Enhancement

7.1 Problems

There are several problems we face during implement the *AceChat whiteboard* system. Some of the problems can be overcome but some of them cannot. All the problems is listed down as below.

7.1.1 Socket Programming

AceChat whiteboard system uses the TCP/IP socket server concept. This socket connects the server and the client. One problem raise up is to decide what stream is suitable to use for sending and received data because socket programming provides a lot of stream objects like *DataStream* and *ObjectStream*. Finally *DataStream* is selected as the socket stream.

7.1.2 Graphical User Interface Design using Java Programming Language

Java provides a lot of layout and interface component to use. One and the only one problem caused by the Java programming language is it takes longer developing time if we compare with others programming language like *Virtual Basic* and *Active Pages Server*.

7.1.3 Lack of Knowledge in Java 2D Graphics

The biggest problem encountered in implement this whiteboard system is lack of knowledge in Java 2D Graphics. Although the Java 2D is a very powerful and flexible class, but because of lack of knowledge in this field, so the outcome system is not perfect enough.

7.2 System Strength

7.2.1 Cross Platform

Java is a programming language that provides "write once, run anywhere" features. *AceChat whiteboard* system can executed in any platform such as Window 98 Second Edition, Window 2000, Window ME, Window XP, Linux, Unix and so on. All the operating system can run this *AceChat whiteboard* system after download and install the *Java Virtual Machine*. They do not need to bother with any other specific hardware or software requirement imposed. As a result, there is always an open door for worldwide audience.

7.2.2 Real Time Drawing Activities

This system can consider fast enough if we compare with others existing whiteboard system. Java language is a very powerful language in graphical management. Because of this whiteboard system is designed for Local Area Network (LAN) system, the data transfer rate can reach 100 Mega Hertz, therefore, this system is become more real time closely.

7.2.3 Friendly User Interface Design

All the button graphics are different from others drawing editor system. The button's graphic is purposely design for the AceChat whiteboard system. All the buttons are easily understood by all level's user. Furthermore, almost all the drawing operation can be done by a single click on the button. It is hoped to ease the user.

7.2.4 Multi – Threading Connection Concept

Multi – threading concept is very useful in networking connection because it allows the server to process separate client connection in separate threads. The system becomes more stable because if one of the user's connected lost, it would not take effect on other users' connection. With multi – threading connection, the system will become more real – time closely.

7.2.5 Three – Dimensional (3D) Drawing

AceChat whiteboard provides 3D drawing tool for the users. Users can either draw a filled 3D rectangle or outlined 3D rectangle on the *Drawing Area.* Such function will make the system more attractive from others whiteboard system.

7.3 System Limitation

7.3.1 Lack of Functional Module

Currently, functionality that provided by *AceChat whiteboard* is limited due to time constraints. For an example, AceChat whiteboard system only provides a few basic shapes drawing for the users like rectangle, and circle. Although the concept of draw triangle, polygon is almost same with draw rectangle, but the system cannot implement it because of time considering. Moreover, this system also did not provide any drawing effect for the users.

7.3.2 Not Suitable For Web Base

Although *AceChat whiteboard* system can be upgraded to web-base application, but it is not suitable. The drawing will become not smooth enough if we use this system as a web – base system. It is because over web – base, the transfer rate will more depends on the system

7.3.3 Limited Font Selection

The *AceChat whiteboard* system has a limitation in font selection. User can only select around 20 types of font in this system. This is because every operation system has its own *available font types,* for an instance, font type "*Curlz*" may appear in Window 2000, but if pass this *"Curlz"* font to Window 98 Second Edition or Linux, it may cause a error. Therefore, *AceChat whiteboard* has a limitation in the font selection.

7.4 Future Enhancement

7.4.1 Functionality enhancement

The functionality of the *AceChat whiteboard* system can be enhanced easily because of its flexible coding. The system might provide more basic shapes for the users to select, for instance triangle, sector, polygon and so on. Other functions like *undo* and

redo must provided by the system to make it more user–friendly. The "*save picture* " function should be added into the system too.

7.4.2 Three – Dimensional (3D) Drawing Control

Because of the extremely growth in the computer world, all the multimedia components should be upgraded to Three – Dimensional (3D) if possible. We have try a 3D drawing tool – 3D rectangle drawing tool. Actually it is possible to make all the drawing tools become 3D since Java Programming Language is selected. Java is prepared to upgrade all its components or controls become 3D control or 3D component. Recently, a new Java package, Java3D is launched. All the 3D drawing tools will become easier to implement because of this Java package

7.4.3 Better connection

If the *AceChat whiteboard* system wants to upgrade as web – base application, a better connection is needed. As Java is used, developer can create a distributed system with *Remote Method Invocation, RMI*. *RMI* provide a better connection for both LAN base and web – base application.

7.4.4 Object Manipulation

This whiteboard system should prepare the object manipulation for the users. Object manipulation such as object rotation, translation, and scaling can ease the users and improve the drawing. Furthermore, with the object manipulation feature, users can copy or paste anything into the *drawing area* such as *.jpeg* or *.gif* file.

7.4.5 Drawing Area's History

For the current system, it is not provide the drawing area's history. In the other words, user login later cannot see the drawing drawn by other users who login previously. This can be solved if the AceChat whiteboard system provides the drawing area's history.

7.5 Conclusion

Developing AceChat whiteboard system is an excited experience. Although it spends almost half of a year to complete, but during the development, I gain a lot of knowledge. As a student who major in Multimedia, I am very happy because I have opportunity to develop this system. From the beginning of development until the moment submit the whole system, my interest in multimedia is increased.

As the *AceChat* system is a big system, the cooperation between group members is very important because we must help each other to success this project. Finally, I felt that all the experience gained from this project would be a useful guide for me in future work.

Appendix A

Ace Chat Installation Guide

Introduction

In order to run *AceChat* system, users need to install some runtime software and full fill some hardware requirements. Below is the Java Development Kit and Database setup guide for the users to refer.

1.0 Installation guide

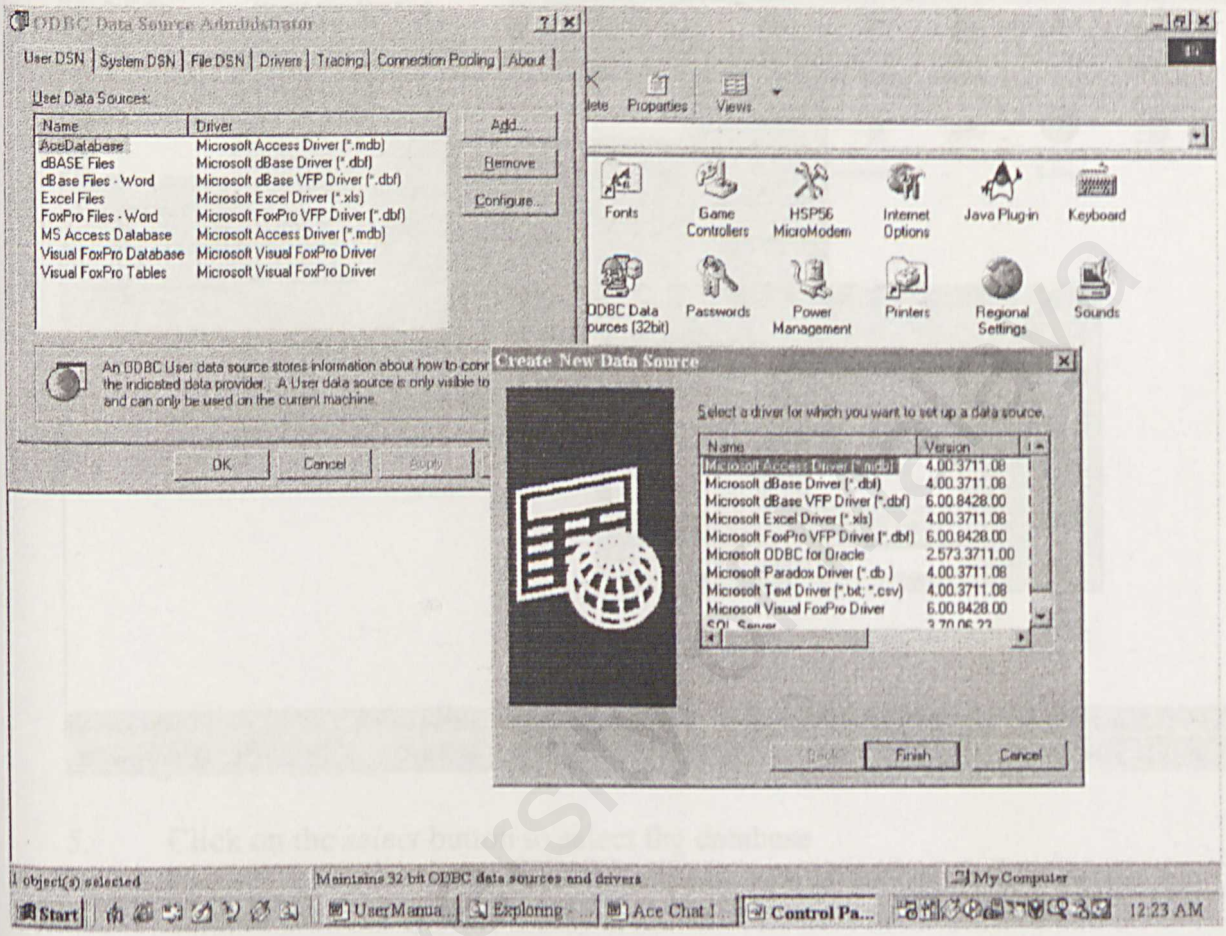
1.1 JDK Installation

1. Download JDK from the http://java.sun.com website.
2. Install JDK to local drive like *C:* or *D:*
3. Setting path for the JDK
    - modify the PATH variable to include the *c:\jdk xxxx\bin* directory (*jdk xxxx* is the installation path of the JDK)

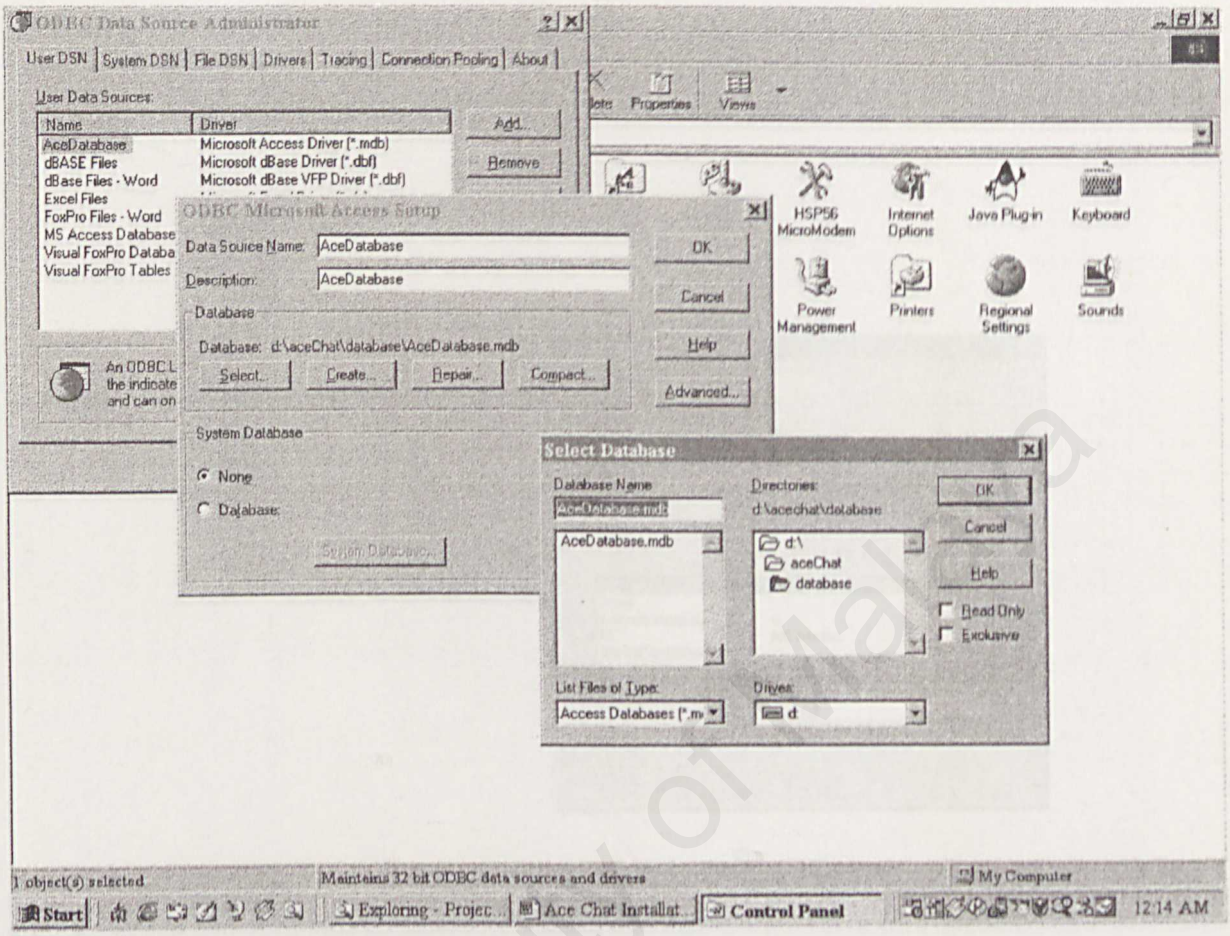
1.2 Text – to Speech Installation

1. Install ibmjs and the ViaVoice Runtime;
2. Setting path and classpath for the ibmjs:
    - modify the CLASSPATH variable to include *dir*\lib\ibmjs.jar, and,
    - modify the PATH variable to include the *dir*\lib directory
3. *dir* refer to the directory of ibmjs installed.
4. run the install.bat file from ibmjs installed directory.
5. Copy the Ace Chat folder into desired drive* from Ace Chat CD;

1.3 Setting Up AceDatabase in ODBC for Server



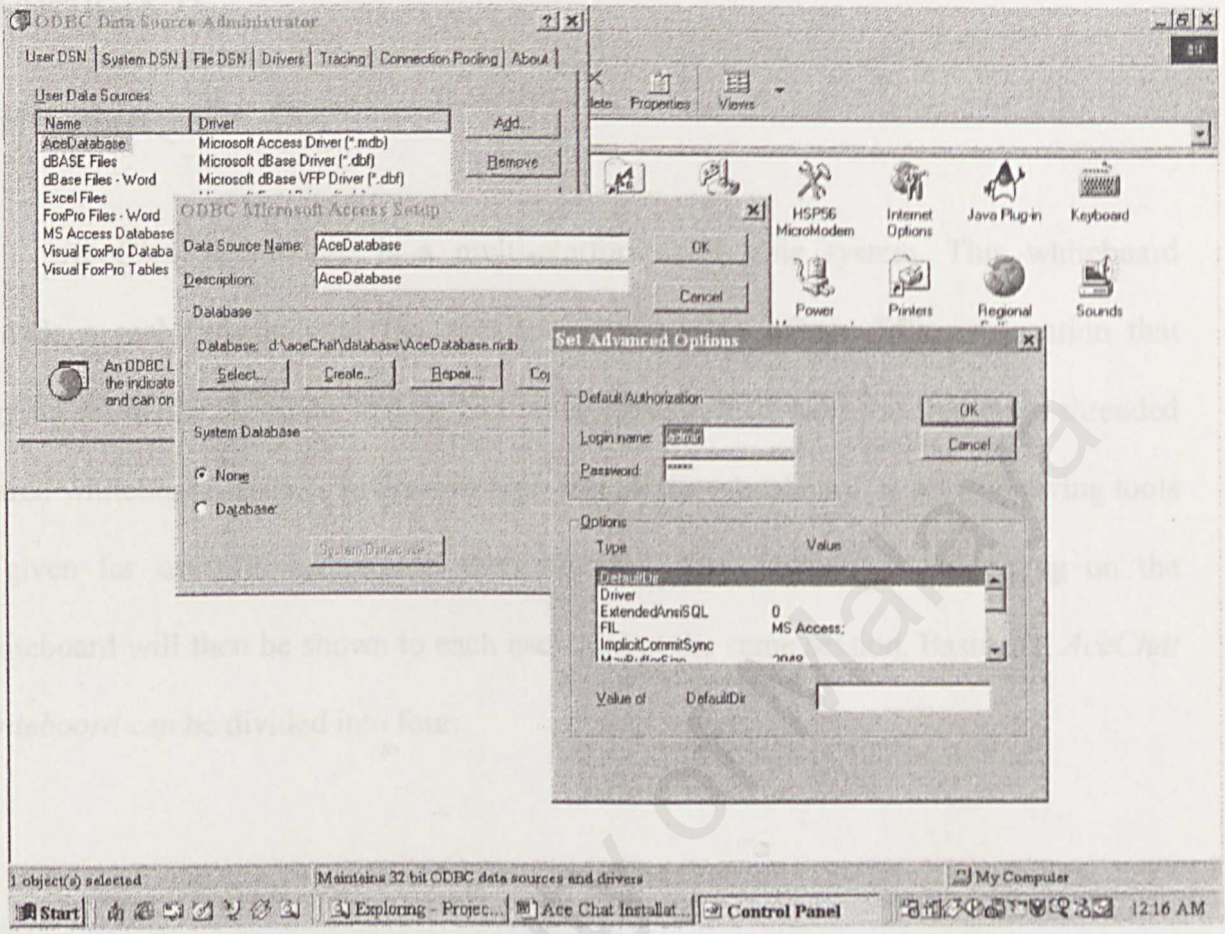
1. Open the ODBC 32bit Driver in Control Panel for Windows users.
2. In the Create New Data Source window, select the Microsoft Access Driver.
3. In the Data Source Name field, type in AceDatabase.
4. In the Description field, type in AceDatabase.



5. Click on the *select* button to select the database

6. Choose the database and click *ok*

7. Then click on the advanced button



8. For login name, enter 'admin'.

9. For password, enter 'admin'.

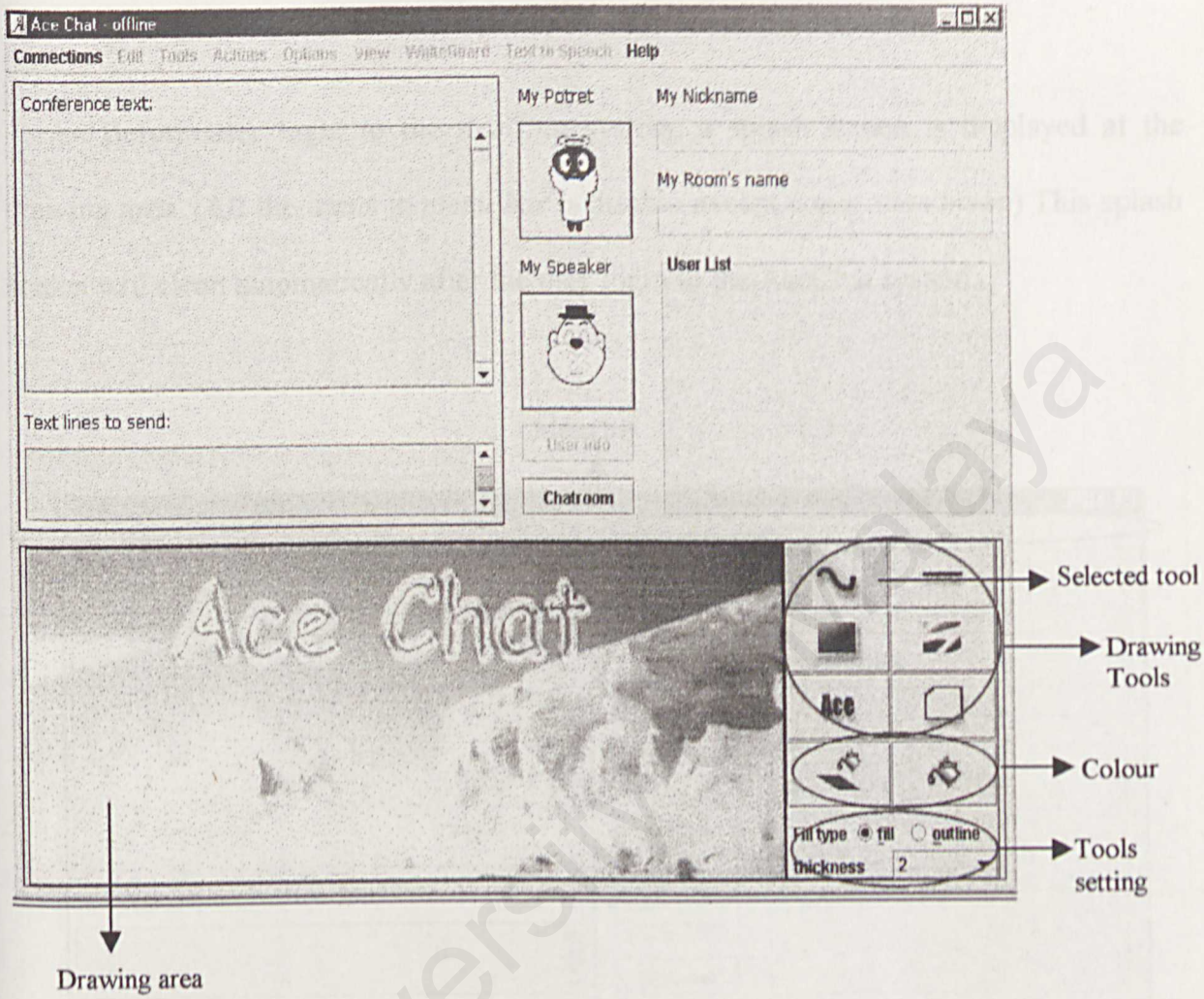
10. Click *OK* to exit the database setting

Appendix B

User Manual

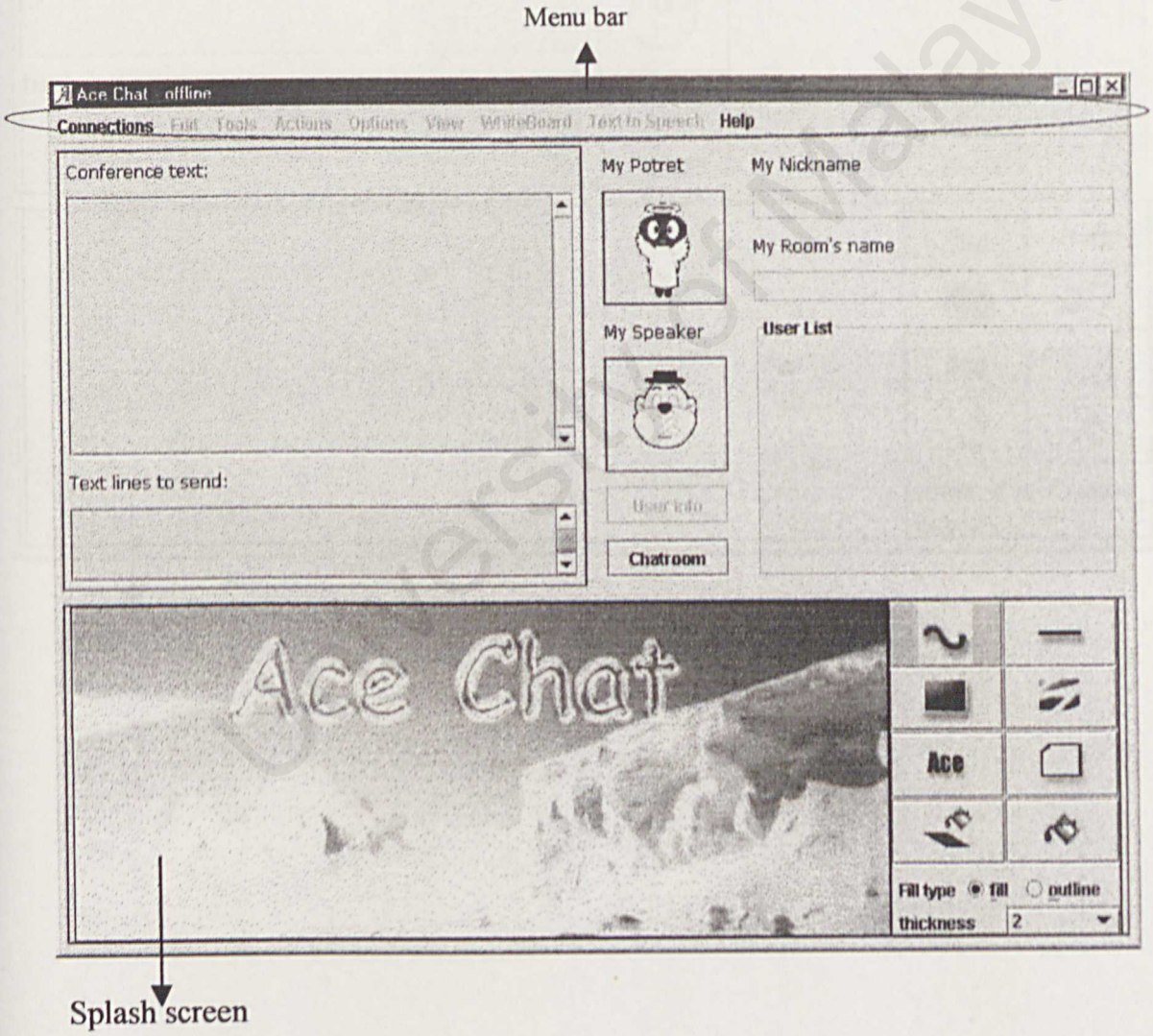
AceChat whiteboard is a multi-platform real time system. This whiteboard provide a real time activity for *AceChat* system. Users from different location that logging on to the AceChat system can communicate, and work on this multi-threaded shared whiteboard. Users can draw or type text on the whiteboard. A set of drawing tools is given for users to accomplish their drawing. The drawing or the typing on the whiteboard will then be shown to each user who at the same session. Basically, *AceChat whiteboard* can be divided into four:

- Drawing Area
- Drawing Tools
- Tool Settings
- Colour

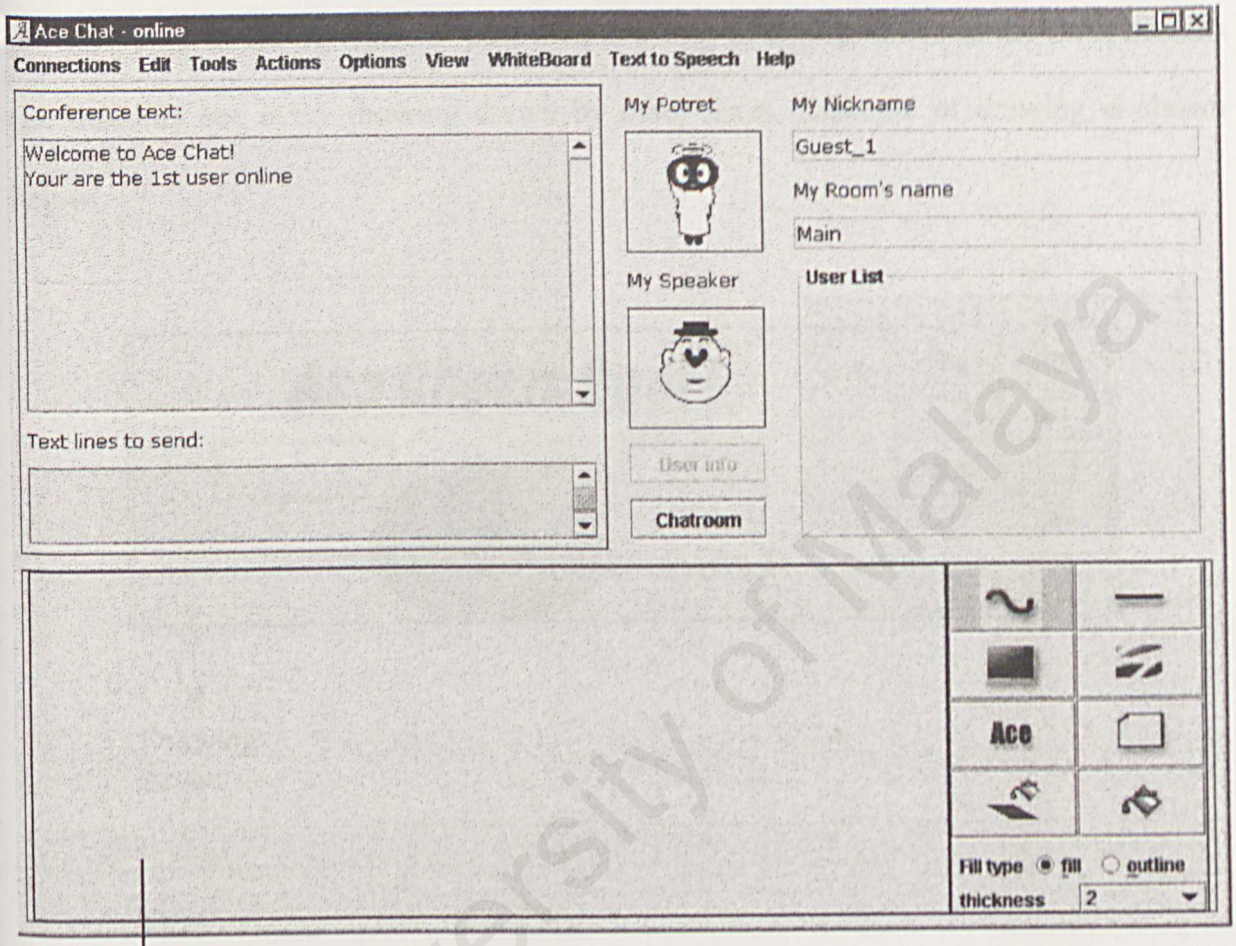


Drawing Area

Before user login to the *AceChat* system, a splash screen is displayed at the drawing area. (All the items in *menu bar* is disable except *connection menu*) This splash screen will clean automatically after the user login to the AceChat system.



Now the *AceChat* system is online.



Drawing Area

The *Drawing Area* is the most important part of the *AceChat whiteboard.* It provides an area for the users to post their drawing or typing. Users in the same session can instantly see every drawing drawn by other users. Example of drawing is shown below.

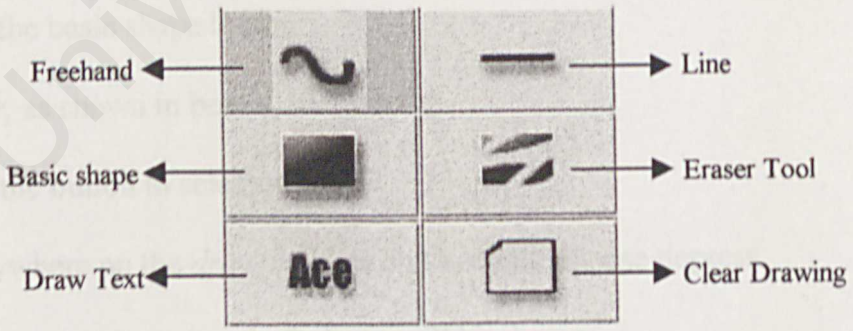
Drawing Area

Drawing Tools

Drawing tools for the *AceChat whiteboard* consists of following tools:

- Freehand
- Line
- Basic shape
- Eraser
- Text
- Clear Drawing

All functions that provide by *AceChat whiteboard* system will perform a real time result during the online discussion.



Freehand and Line

How to draw a line into the drawing area

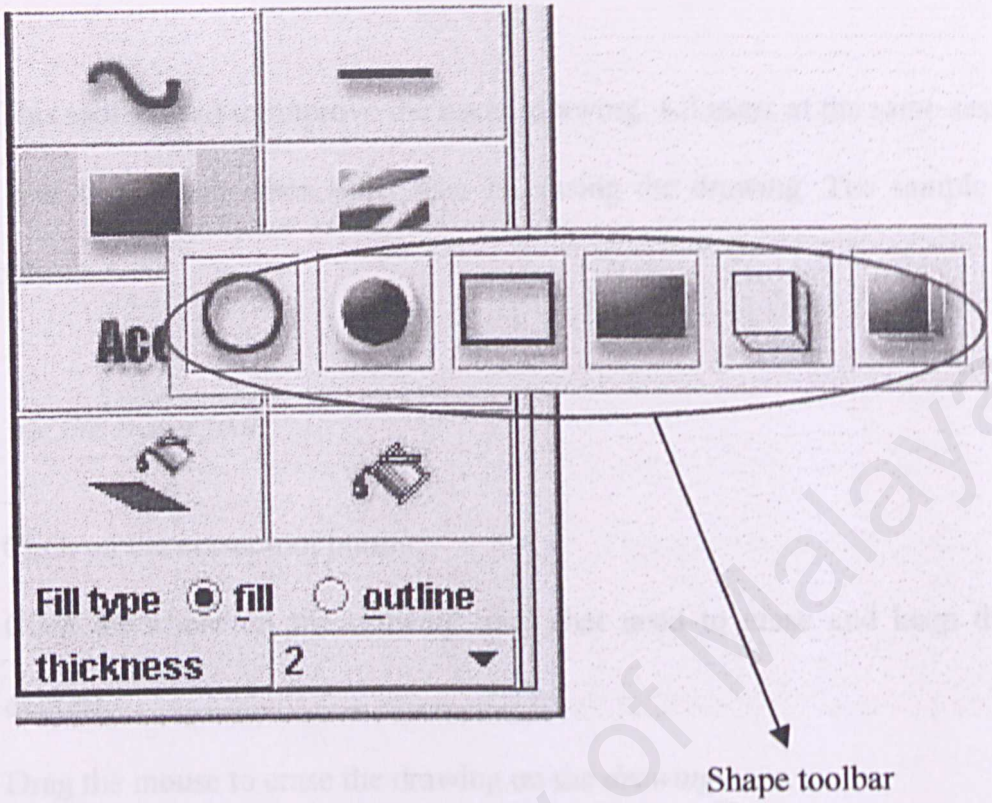
1. On the *Drawing tools,* click either the *Freehand Tool* or the *Line Tool* button.
2. Click anywhere on the *Drawing Area* and keep the mouse button depressed.
3. To draw the line, drag the mouse, a draft line will be shown.
4. When user releases the mouse button, a line is drawn

Basic Shape

AceChat whiteboard provides 2d rectangle, 3d rectangle and circle shape for the users to choice. These basic shapes function is hoped can ease user.

How to draw a shape into the drawing area

1. Click on the basic shape button.
2. A *toolbar,* as shown in below, will pop up.
3. Click on the button to select a shape.
4. Click anywhere on the *drawing area* and keep the mouse depress.
5. Drags the mouse button, a draft shape will be shown.
6. When user releases the mouse button, the shape is drawn

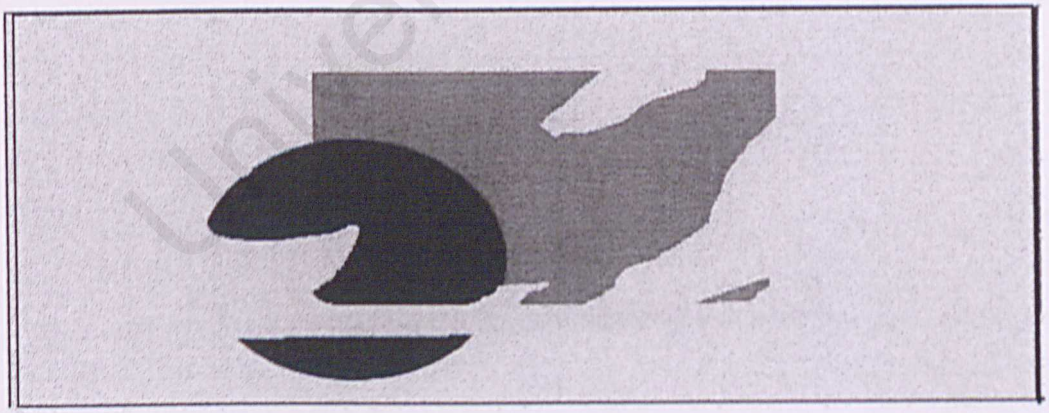


Eraser Tool

This tool is used to improve the users' drawing. All users at the same session may see the real time result when other user is erasing the drawing. The sample result is shown below.

How to use the eraser tool

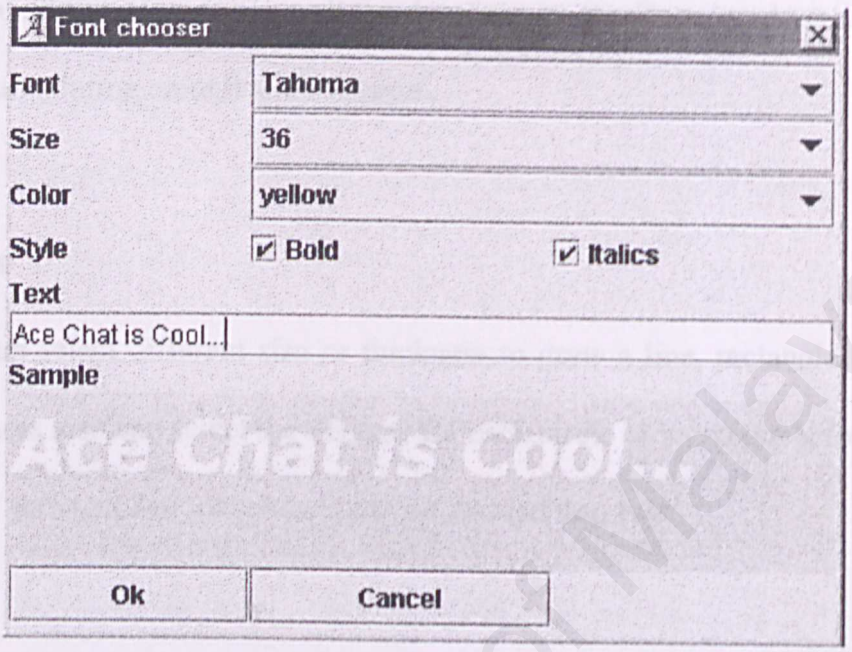
1. Click on the eraser tool button.
2. Click anywhere on the *drawing area* that need to erase and keep the mouse depress.
3. Drag the mouse to erase the drawing on the *drawing area*.
4. Release the mouse when finish.



Text Drawing

How to draw text into the drawing area

1. Click on the draw text button.
2. A Font Chooser window, as shown in below, will pop up when user click at the *drawing area.*
3. Type in the text at the *text area,* select the prefer font type, size, colour, and style. The sample text is shown. Click on the *OK button* to confirm or *Cancel button* to discard.
4. The *draft floating text* will follow the mouse position.
5. Left click the mouse once to draw the text into *Drawing Area.*



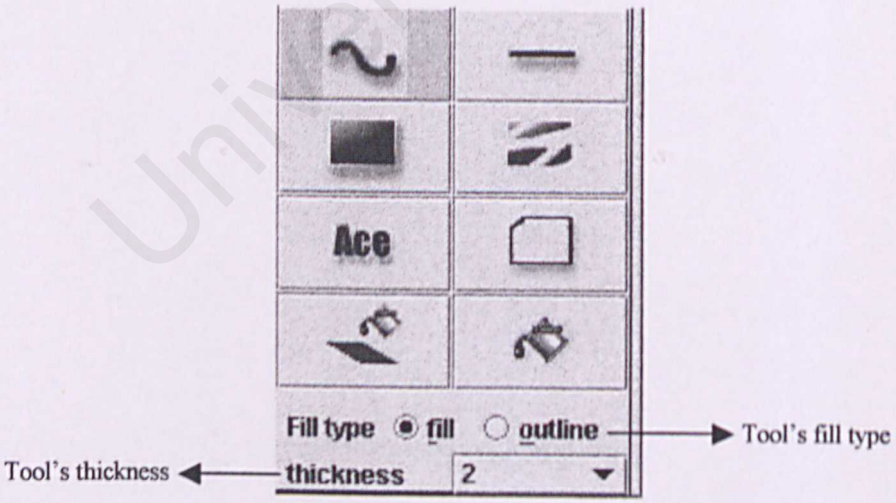
Clear Drawing Area

This tool allows users to clear all the drawings on the *Drawing Area* and view the results in real time during an online discussion.

Tool Settings

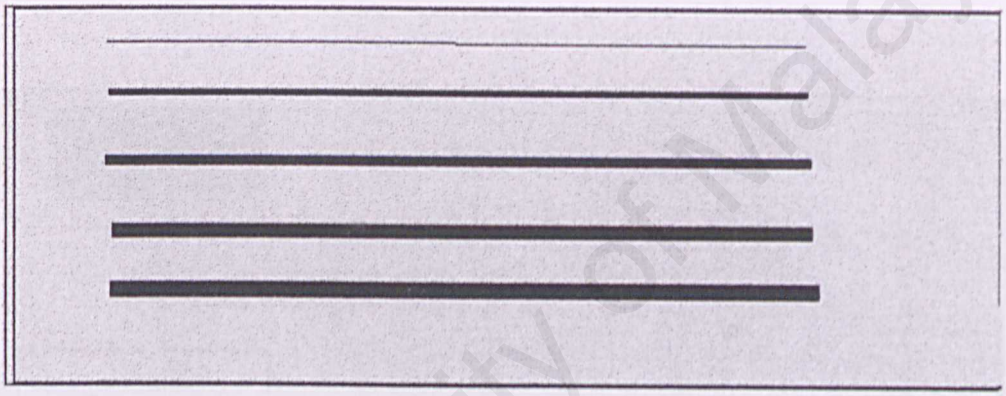
Users can select different size or thickness to draw a line, rectangle, and circle. Furthermore, users also can select the fill type for each circle or rectangle before draw it. *Tool Settings* of the *AceChat whiteboard* can be divided into two:

- Thickness
- Fill type



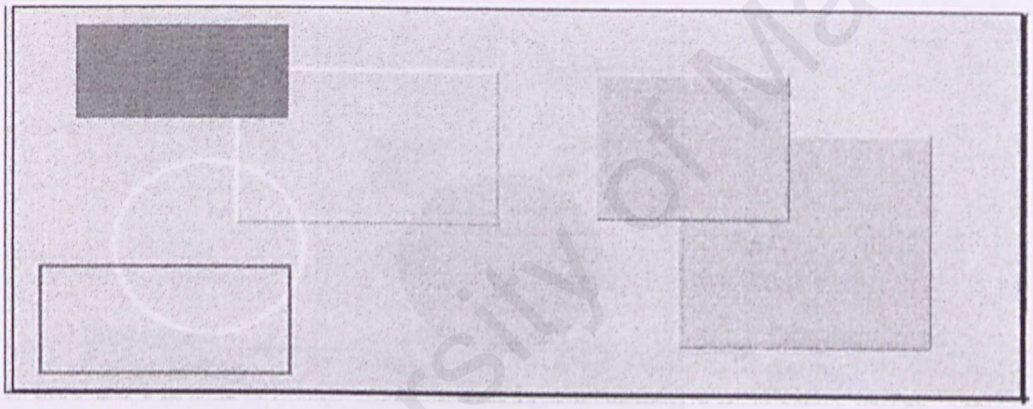
Thickness

When users want to draw a line or a shape into *Drawing Area*, they can choose different thickness for their drawing tools. It is done by simply click on the thickness drop down menu and select the preferred size. Below shows some lines drew by different thickness.



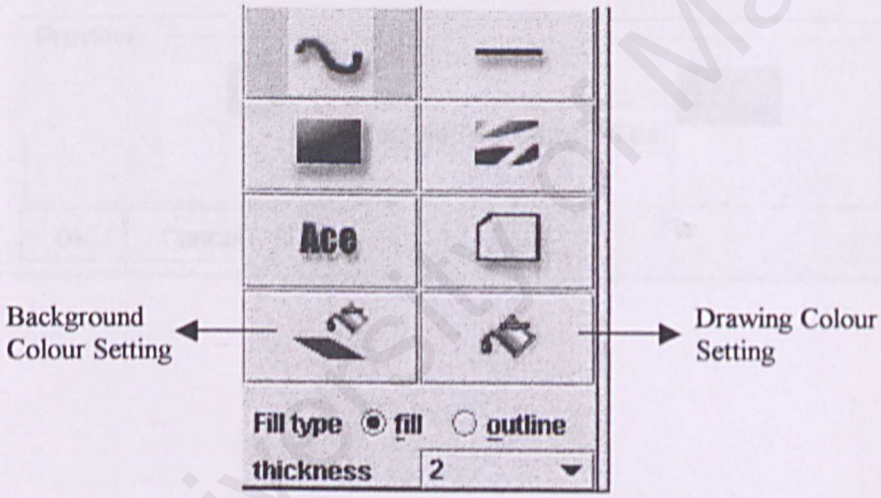
Fill Type

All the basic shapes that provide by *AceChat whiteboard* can be filled or outlined shape. For instance, users can draw a filled circle or an outlined circle. The *Fill Type* status can be change by click on the *fill* or *outline* check box. The sample results of the basic shape drawing with different *Fill Type* is shown in below.



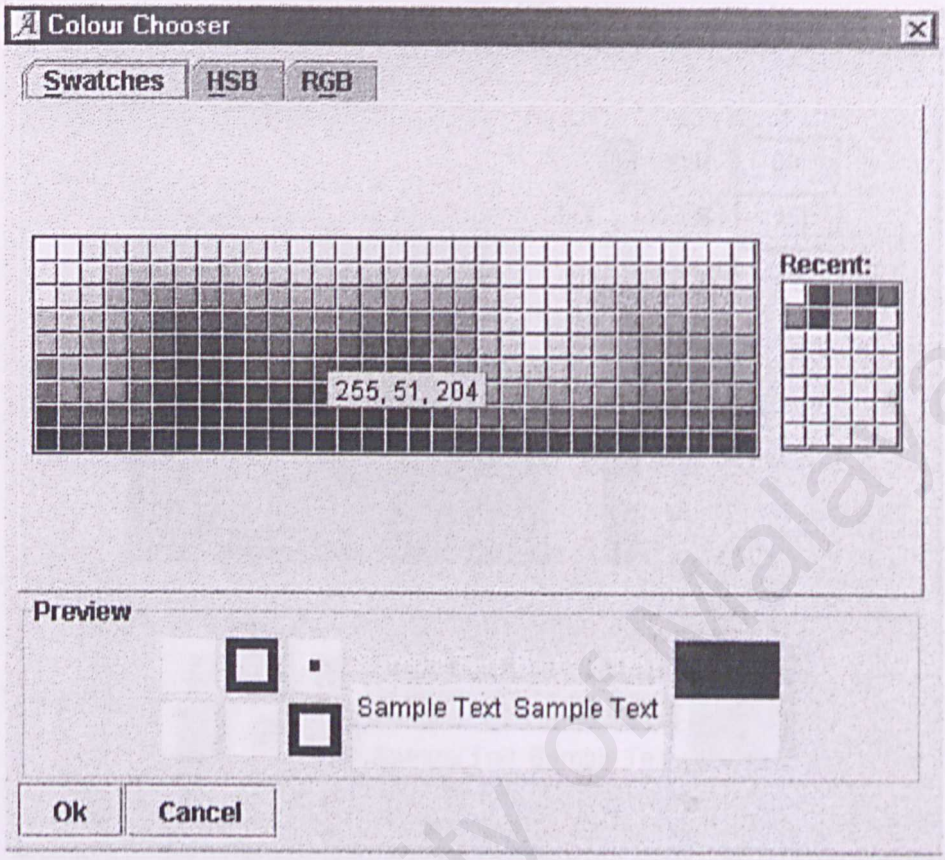
Colour

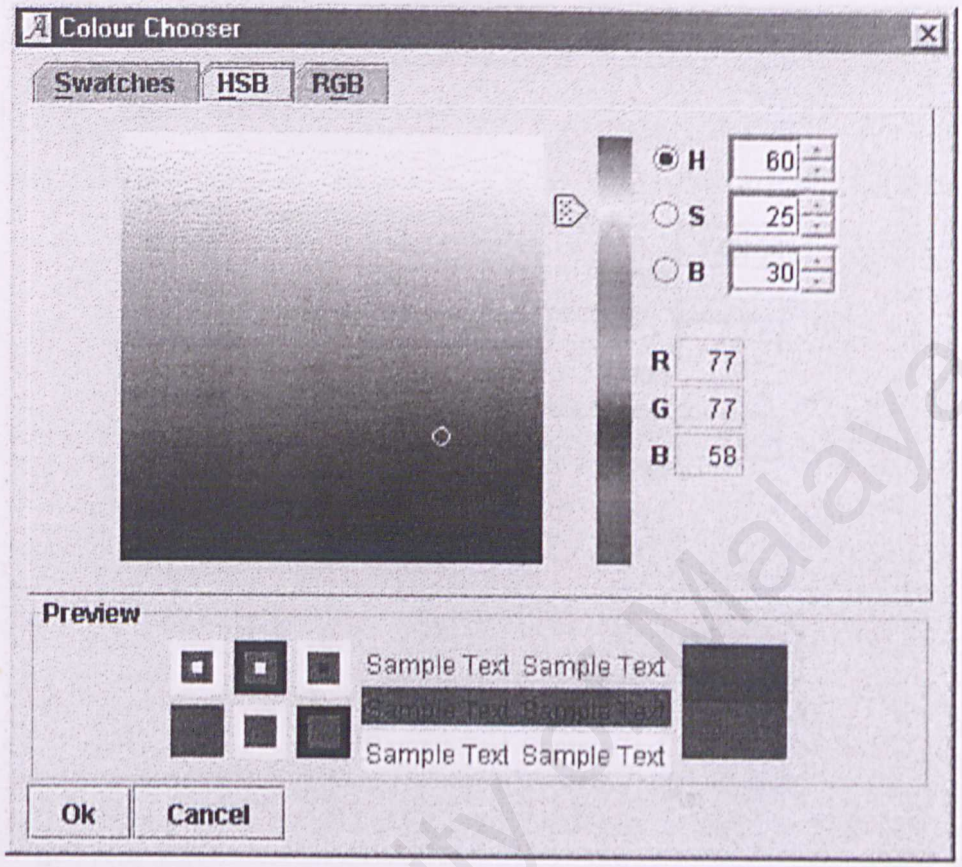
Colours enables user to set the colour of the selected drawing tool. It is hoped to provide a variety of colours that will ease users in their drawing. The colour module of the *AceChat whiteboard* consists of *Background Colour setting* and *Drawing Colour setting*. *Background Colour setting* will change the colour of the *Drawing Area* while *Drawing Colour setting* will change the *Drawing Tool*'s colour.

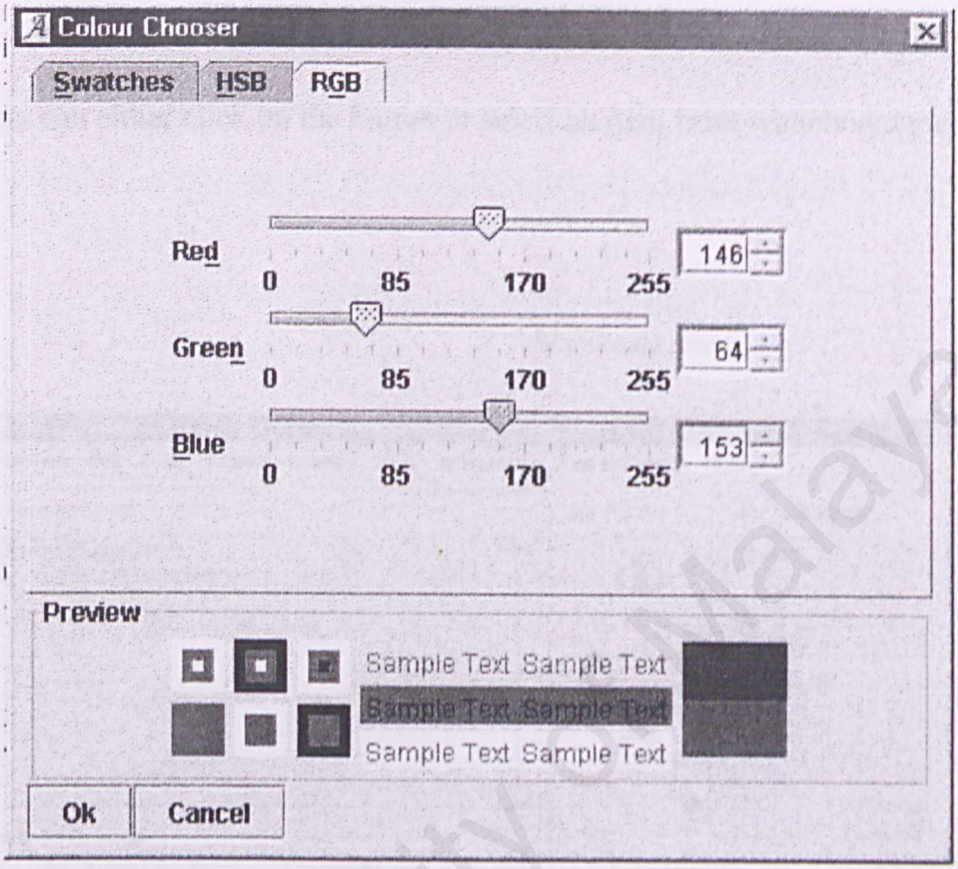


How to set the background colour or drawing colour

- Click on *Background Colour setting* button or *Drawing Colour setting* button.
- A *Colour Chooser* window, as shown in *figure 13*, will pop up.
- User can select the preferred colour by **RGB mode** or **HSB mode**.
- Click *OK button* to confirm the transform or *Cancel button* to discard the transform.

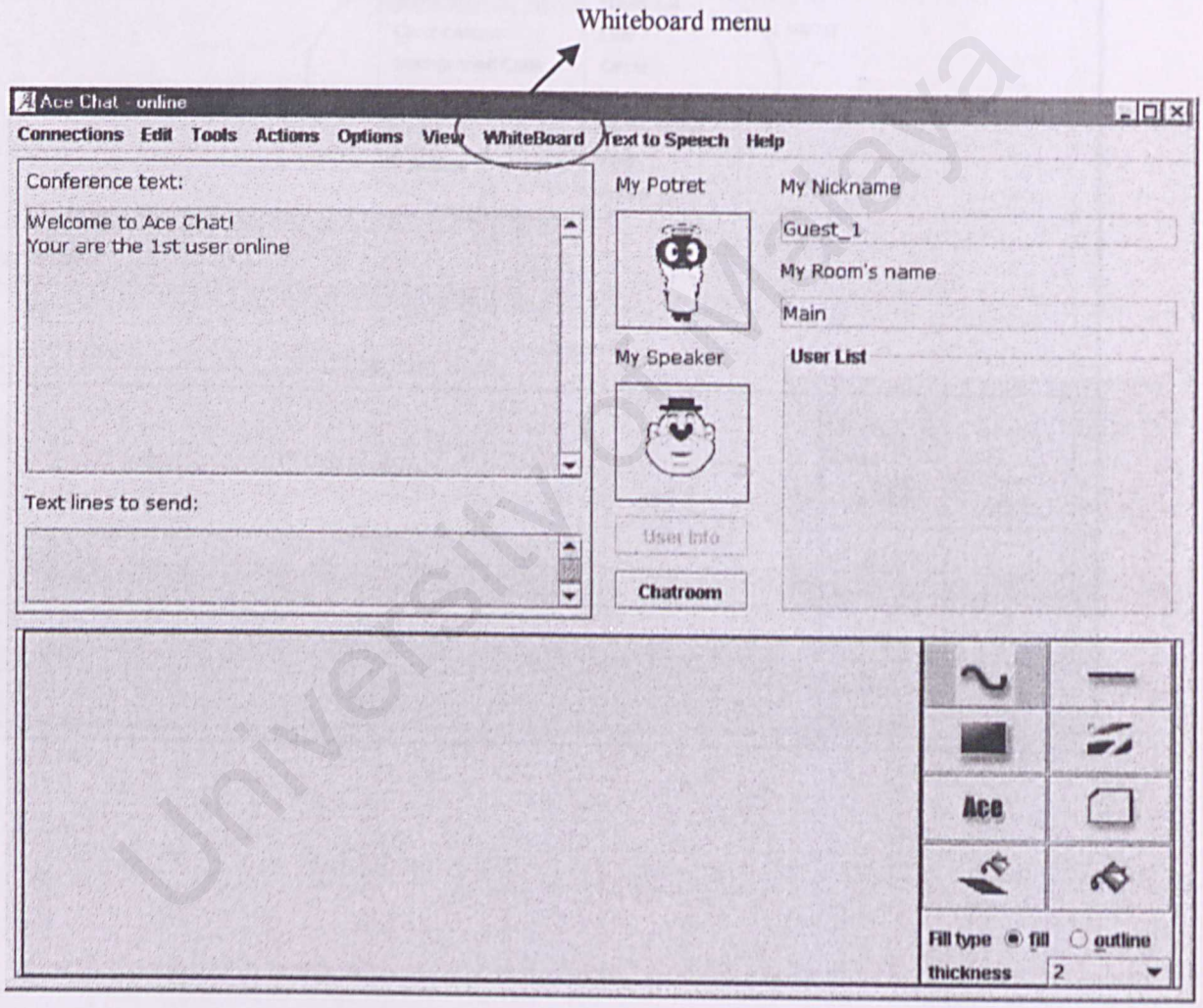




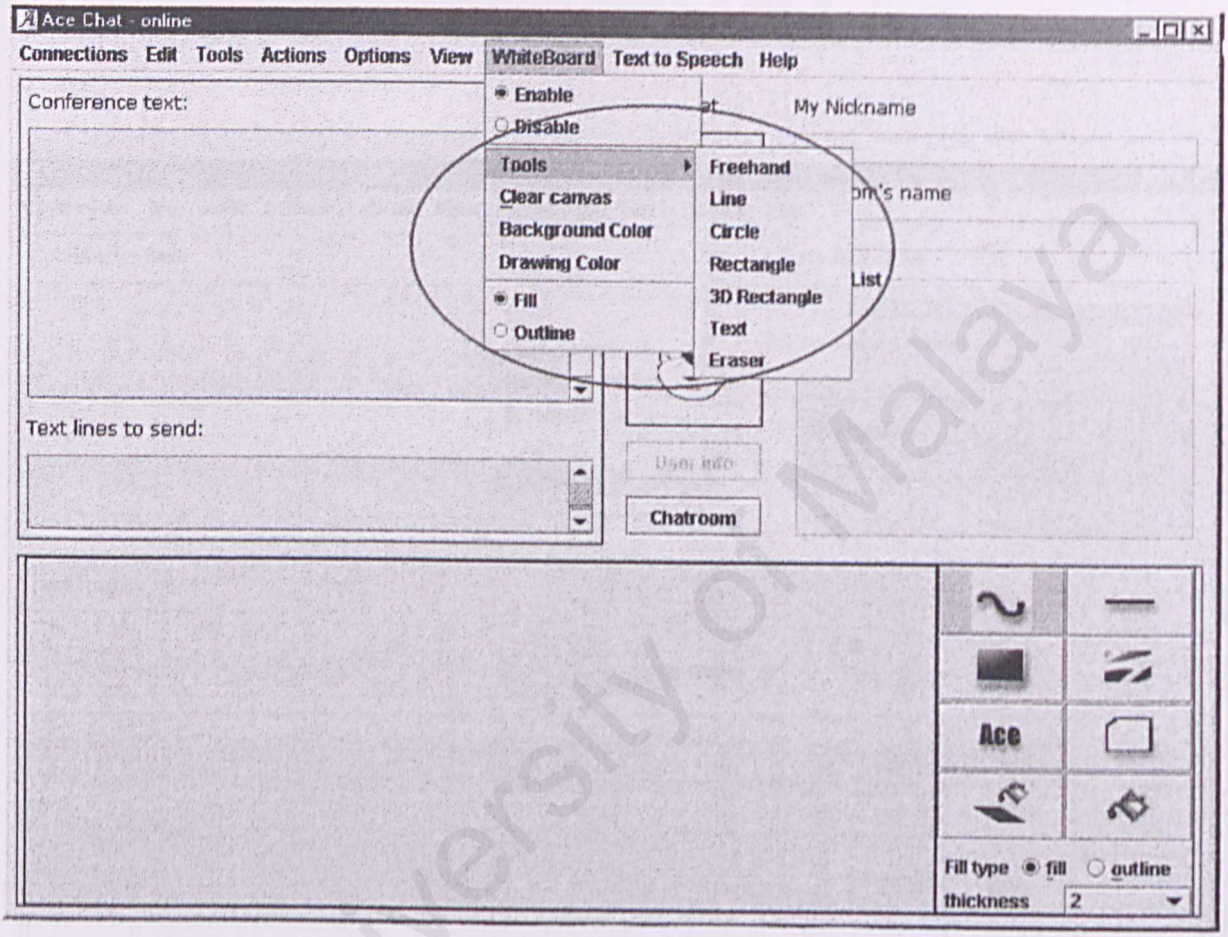


Using the Whiteboard Menu at the Menu Bar

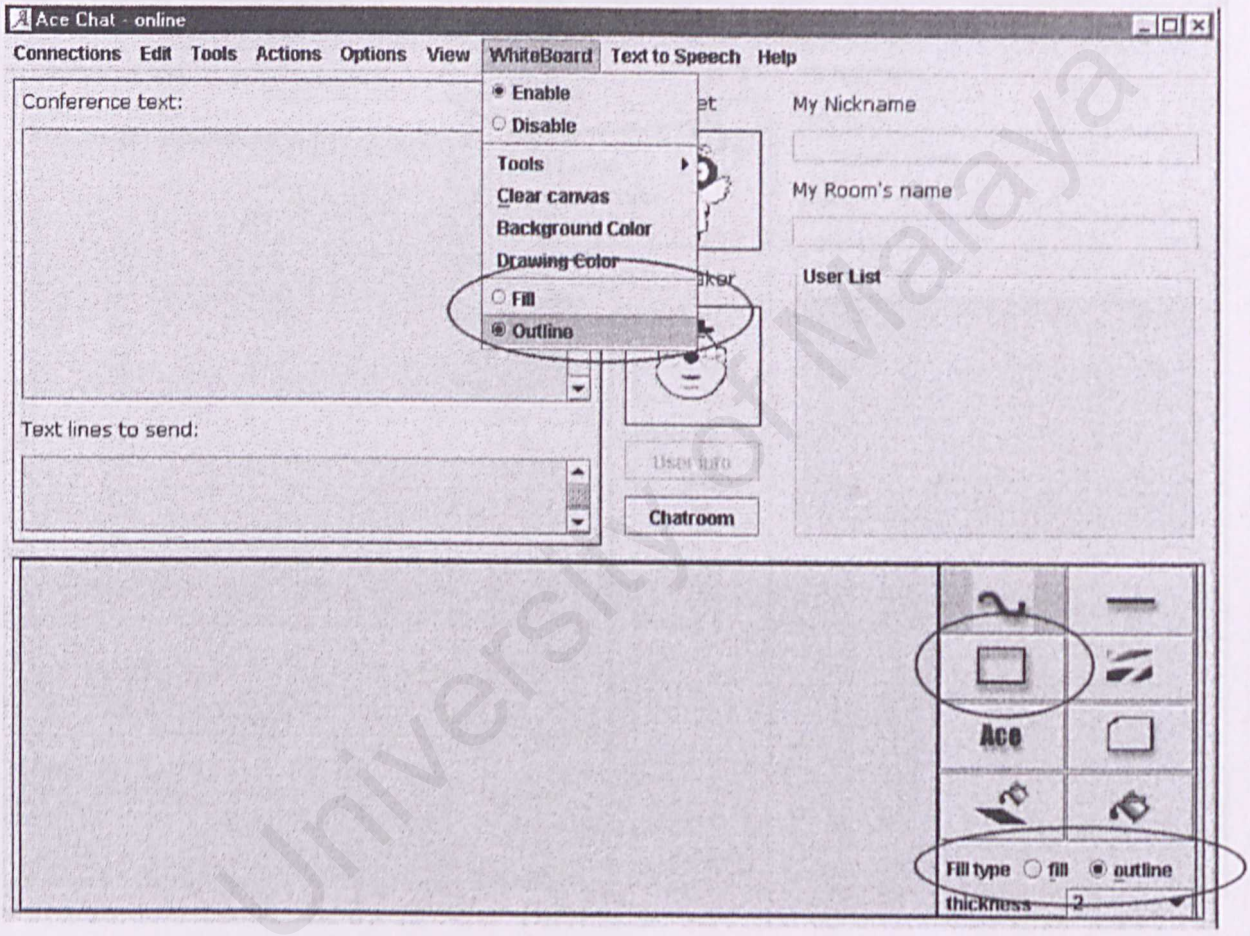
Users can either click on the button or select an item from whiteboard menu bar to draw.



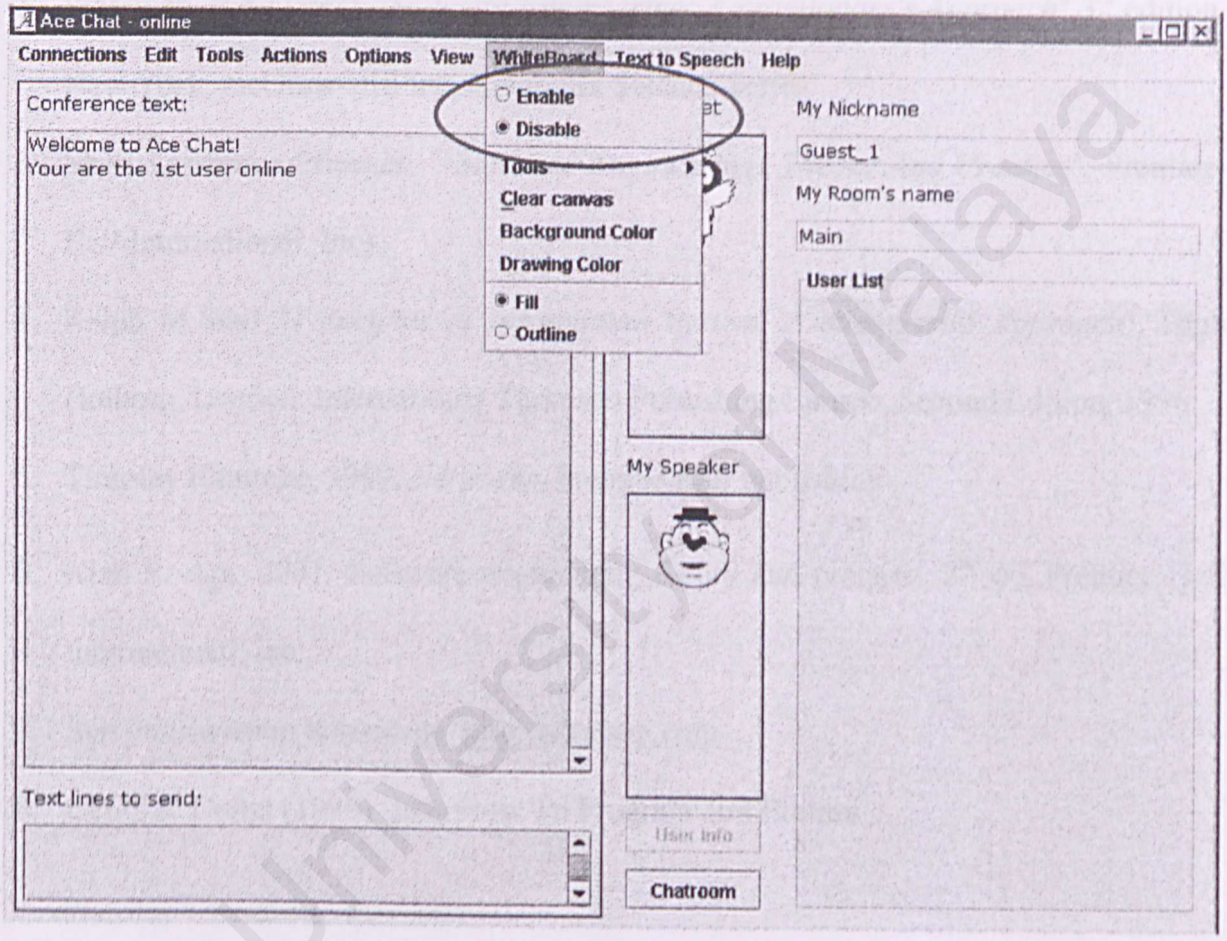
When users click on the whiteboard menu, an items list will drop down from the whiteboard menu. Users can select the drawing tools from the *tool* selection.



There is a linkage between *tool setting, basic shape button and tools preference* at the whiteboard. When the users click on the *fill* or *outline* check box at the *whiteboard menu*, the tool setting will change its filled type too. At the same time, the basic shape will change according to the changes make.



If the users want to concentrate on text chatting, they can hide the *drawing area* by select the *disable* item from the *whiteboard* menu. When the users want to use the whiteboard, they can select the *enable* item from the *whiteboard* menu to enable it.



References

1. Kendal & Kendall, "*System Analysis and Design*", Prentice Hall International Editions, Fourth Edition, 1998.
2. Pressman, R.S (1992), "*Software Engineering: A practionaer's Approach*" 3rd edition. New York, mcGraw-Hill Inc, Computer Science Series.
3. Shari Lawrence Pfleeger, " *Software Engineering: Theory And Practice*", Prentice-Hall International, Inc)
4. Ralph M.Stair "*Principles of Information System, A managerial Approach*", High Holborn, London: International Thomson Publishing Europe, Second Edition, 1996.
5. Timothy Ramteke, 1999, *Networks*, Prentice Hall publishing
6. Alan R. Apt, 2001. Software engineering theory and practice, 2nd ed, Prentice Hall International, inc.
7. Sun microsystem homepage. http://java.sun.com
8. Deitel & Deitel (1999). Java How To Program 3nd Edition.