

WEB-BASED INSTANT MESSENGER

By

Charles Atuchukwu

A thesis submitted in partial fulfillment of
the requirements for the degree of

BSc. Computer Science [Honours]

University of the Western Cape

2009

Date: June 4, 2009

University of the Western Cape

Abstract

WEB-BASED INSTANT
MESSENGER

by Charles Atuchukwu

Chairperson of the Supervisory Committee: Professor Bill W. Tucker
Department of Computer Science

Instant messaging (IM) application has witnessed a tremendous improvement and growth in popularity as a means of internet communication since inception because of its real-time and non real-time nature. Compare to other internet communication methods such as e-mail where you have to wait for the recipient to check his or her email and sent reply, IM is instant when the recipient is online. Traditional IM applications have to be downloaded installed and configured before it can be used; besides, they are platform dependent. These constitutes serious problems for users especially those using the application in public computers such as school libraries, computer labs, and internet cafés, where most of the time the application is not installed and the user will not have administrator's right to install the application if at all he or she knows how to do that. Web-based instant messenger is the solution to these problems since it does not require any download, installation or configuration and besides it is platform independent. This project will try to implement a web-based instant messaging application.

TABLE OF CONTENTS

TABLE OF CONTENTS	I
LIST OF FIGURES	III
ACKNOWLEDGMENTS	V
GLOSSARY	VI
INTRODUCTION	1
The project.....	1
Motivation	2
Project layout.....	2
Conclusion	2
the users requirement document (urd)	3
Introduction	3
User's view of the problem	4
Brief description of the problem domain	5
Complete description of the problem	5
What is expected from the system?	6
What is not expected from the system?	7
Summary.....	7
requirement ANALYSIS document (rAd)	8
Introduction	8
The real problem(s)	8
Application download, installation and configuration	8
Platform dependent	9
Firewall issue	9
The systems and software requirements for the system	9
Software requirement.....	9
Server requirement	10
Existing solution.....	11
The components of IM implementation	12
User interface specification.....	13
Introduction	13
Complete description of the user interface	13
What the interface looks like to the user	14

Login Interface	14
New Account interface.....	15
Main Window	16
Chat Window	17
How the user interface behaves	17
How the user interacts with the system.....	19
Summary.....	21
OBJECT ORIENTED ANALYSIS (OOA) / HIGH LEVEL DESIGN (HLD).....	22
Introduction	22
Data dictionary	22
Objects Attributes.....	22
Description.....	22
Class diagram	24
Summary.....	28
OBJECT ORIENTED DESIGN / LOW LEVEL DESIGN	31
Introduction	31
Details of class attributes and methods	31
Objects Attributes.....	31
Data type	Error! Bookmark not defined.
Description.....	31
State diagram	33
Pseudo-code	Error! Bookmark not defined.
INTRODUCTION	Error! Bookmark not defined.
APPENDICES.....	39
BIBLIOGRAPHY.....	40
INDEX.....	42

LIST OF FIGURES

<i>Number</i>	<i>Page</i>
FIGURE 1: BAR CHAT REPRESENTATION OF FEATURES USERS USUALLY USE IN IM APPLICATION	4
FIGURE 2: MYSPACE IM APPLICATION	5
FIGURE 3: THE PROCESS OF INSTANT MESSAGING COMMUNICATION.	12
FIGURE 4: LOGIN INTERFACE FOR LOGGING IN TO THE SYSTEM.....	15
FIGURE 5: NEW ACCOUNT INTERFACE FOR CREATING NEW ACCOUNT IN THE SYSTEM.....	15
FIGURE 6: MAIN WINDOW FOR PRESENCE INFORMATION AND CONTACT LIST OF THE USER.....	16
FIGURE 7: CHAT WINDOW FOR CHATTING WITH OTHER USERS.....	17
FIGURE 8: MAIN WINDOW SHOWING OPTIONS FOR PRESENCE AWARENESS AND ADD CONTACT FORM.	18
FIGURE 9: MAIN WINDOW SHOWING OPTIONS AVAILABLE WHEN A CONTACT IS RIGHT CLICKED ON.....	19
FIGURE 10: USE CASE DIAGRAM SHOWING HOW USER INTERACT WITH THE SYSTEM	20
FIGURE 11: CLASS DIAGRAM SHOWING THE NAME, ATTRIBUTES AND METHODS OF EACH CLASS	25
FIGURE 12: RELATIONSHIP BETWEEN THE CLASSES	26
FIGURE 13: CLASS DIAGRAM FOR THE APPLICATION DOMAIN	27
FIGURE 14: CLASS DIAGRAMS FOR THE INTERFACE DOMAIN	28
FIGURE 15: LOGIN STATE DIAGRAM DESCRIBING THE PROCESS OF LOGGING IN TO THE SYSTEM.....	34
FIGURE 16: NEW ACCOUNT STATE DIAGRAM DESCRIBING THE PROCESS OF CREATING ACCOUNT.....	34
FIGURE 17: SEQUENCE DIAGRAM SHOWING THE STEPS FOR USER INTERACTION WITH THE SYSTEM	35
FIGURE 18: ACTIVITY DIAGRAM SHOW THE PROCESS OF USER INTERACTION WITH THE SYSTEM.....	36

LIST OF FIGURES

TABLE 1: THE DATA DICTIONARY OF THE SYSTEM	24
TABLE 2: DESCRIPTION OF CLASS FOR THE APPLICATION DOMAIN	28
TABLE 3: DESCRIPTION OF TABLE FOR THE INTERFACE DOMAIN	29
TABLE 4: THE DETAILED DESCRIPTION OF THE ATTRIBUTES AND METHODS.....	33

ACKNOWLEDGMENTS

The author wishes to [Click and type acknowledgments]

GLOSSARY

Word. [Click and type definition here.]

Junk. This is junk, a term for nonsense.

Chapter 1

INTRODUCTION

Instant messaging (IM) is an online real-time communication between two or more people based on typed text messages, over the internet or Local Area Network (LAN) [1]. Web-based Instant Messaging is an IM that is integrated or embedded in a web browser; it requires no application, or update download or installation.

Web-based IM has witness a tremendous growth in popularity since inception because of the advantages over a traditional IM application. Web-based IM requires no application download, updates, installation or configuration. You only require a computer that is connected to the internet with a web browser to communicate with your colleagues, friends, and relations. Secondly, web-based IM is platform independent, which means that it can be used in any operating system provided that there is a web browser and internet connection or LAN. Lastly, with web-based IM there is no firewall problem. Since the application is embedded in a web browser it uses http port so there is no need for any configuration to bypass firewall.

The project

The main idea of this project is to develop and implement a web-based instant messaging system. If time permits, voice chat and use of web-camera will be added to the system to increase its functionality. The system will have support for text chat, contact list, presence awareness, chat history log, file transfer, voice chat, and use of web-camera.

Motivation

Traditional instant messaging applications presents a lot of problems and challenges to users especially those using them in public computers such as school libraries, computer labs, and internet cafés; due to the fact that the application have to be downloaded, installed and configured before it can be used and also the fact that it is platform dependent. These problems restrict user's access to cheap and convenient means of communicating with their colleagues, friends, and relations. Then an idea dropped off in my mind to develop and implement a web-based IM system that will completely eliminate these problems users are experiencing so that they will have unrestricted access to cheap and convenient means of communicating with their colleagues, friend, and relations.

Project layout

Chapter 1 of this project introduces traditional instant messaging IM applications and their drawbacks, and web-based IM as a solution to the problems of traditional IM applications. In chapter 2, takes a look at the User's Requirement Document (URD); the problems from the user's point of view, and features that are expected and those not expected from the system. Chapter 3 will deal on the Requirement Analysis Document and finally chapter 4 will concentrate on the project plan.

Conclusion

The traditional instant messaging applications made a positive and tremendous impact in internet communication but they pose a lot of challenges to users which limits user's access to cheap and convenient means of communication. Web-based instant messaging provides an answer to most of the problems and challenges of traditional IM applications. The next chapter looks at user's requirement document.

Chapter 2

THE USERS REQUIREMENT DOCUMENT (URD)

Introduction

This chapter presents the Users Requirement Document (URD) which describes the problem from the user's point of view, a brief description of the problem domain, and complete description of the problem. Also what is expected from the system and what is not expected from the system will be highlighted.

This document is a reflection of survey carried out to gather information from users on the problems and challenges they are encountering from the current implementation of instant messaging applications, what they think about the system, how they see it and what they want from it. Questionnaire was used for the survey because of the following reasons; it is cheap to implement, do not require much effort like verbal and telephone survey, and often have standardized answers that make it easy to analyze.

The questionnaire consists of series of 10 questions- whether the user use IM applications, if yes why they use it, which application they use, why they prefer that particular application, the features they like most in that application, problem(s) they have encountered with that application if any, etc, (see appendix A, for a sample copy of the questionnaire); about ten (10) copies of the questionnaire were distributed to users and all the uses responded to the questions. See Fig 1 for a bar chart presentation of the survey.

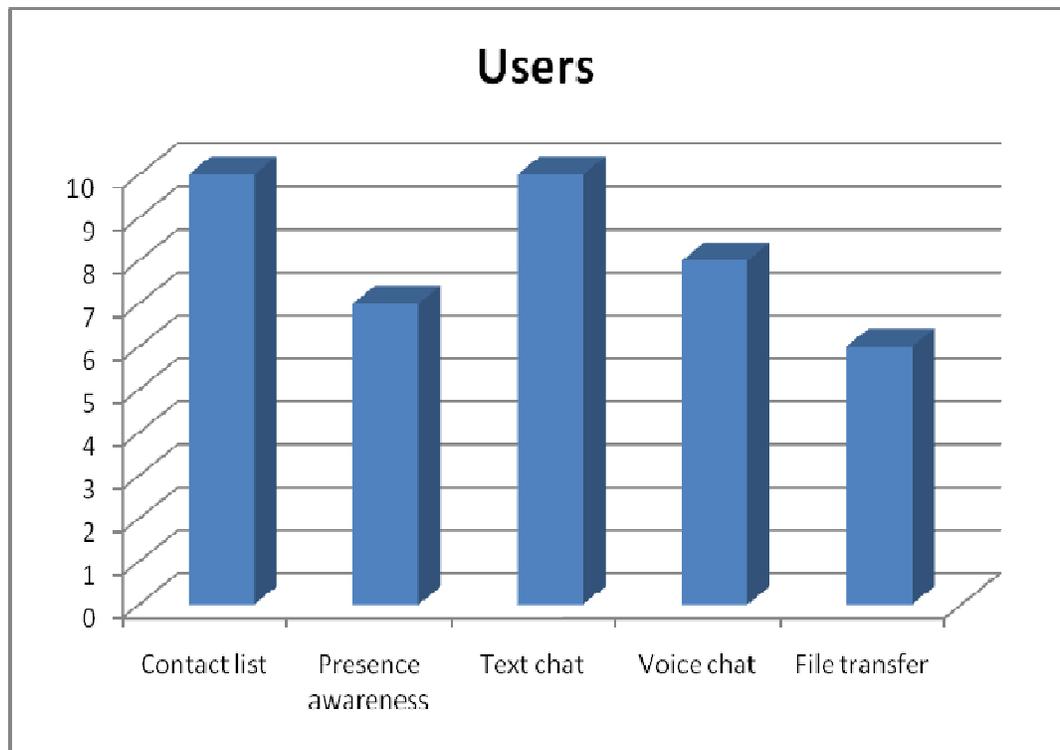


Figure 1: Bar chart representation of features users usually use in IM application

User's view of the problem

Based on the survey carried out, it was obvious that users require a system that will enable them to conveniently and effectively communicate with colleagues, friends and relations, in real-time or non real-time, based on typed text message or voice conversation. Allow them to maintain contact list- to eliminate the need for them to know and remember the user names or screen names of people they want to communicate with, and support for presence awareness- to allow them to set their online status or view the online status of people. Lastly, they want to be able to archive text conversation with other people (chat history log).

Brief description of the problem domain

Users want unrestricted access to a system, or platform that will allow them to conveniently and effectively communicate in real-time or non real-time with colleagues, friends and relations, by text message or voice. Using all the available tools and features of the system such as contact list, presence awareness, chat history log, web camera.

Complete description of the problem

For users to be able to communicate conveniently and effectively, the system must be able to support the following features: figure 1 show a snap shot of the features in action.



Figure 2: MySpace IM application

Text chat: ability to communicate easily and conveniently in real-time with family and friend based on type messages [4].

Contact list: a list of contacts (screen names, pictures) with whom a direct communication can be opened instantly when clicked on [4].

Presence awareness: a visual display indicator showing the status or availability of an instant messaging user [4].

Voice chat: support for voice (audio) communication [4].

Chat history log: ability of the system to store or maintain archive of text conversation with other people [4].

What is expected from the system?

The system is expected to have the following features.

Platform independent: - The system will not be platform dependent. It will work in any operating system provided that there is a browser and internet connection or LAN.

Text chat: - The system will support real-time communication between two or more people over the internet or LAN based of typed messages (text).

Contact list: - The system will support contact list- list of screen names of user's colleague, family and friends.

Presence awareness: -The system will support presence awareness feature.

Chat history log: - The system will support archiving of users instant messaging conversations.

Voice chat: - Voice chat will be implemented if time permits.

Video chat: - Video chat will be implemented if time permits.

What is not expected from the system?

The system will not support the following features.

Chat room: - The system will not have chat rooms feature.

File transfer: - The system will not support file transfer.

Secure / encrypted communication: - The system will not support secure / encrypted communication.

Summary

There has been a tremendous increase in the use of instant messaging applications for communication with two or more people. A survey (in form of questionnaire) was carried out to gather information from users on their experience with instant messaging applications. Some of the problems and challenges were described from the user's point of view. Also the features that are expected from the system and those not expected were described. The next chapter will interpret the user's requirements from the designer's point of view.

REQUIREMENT ANALYSIS DOCUMENT (RAD)

Introduction

This chapter takes the users requirement document (URD) as the starting point and looks at the problem from the from a designer's point of view. The analysis focus on the system and software requirements needed to implement the user requirements. Also the actual details of the problem that the users may not be aware of will be identified and discussed in this chapter.

The real problem(s)

Before the advent of web-based instant messaging, traditional Instant Messaging applications did not completely meet user's expectation because of some issues. To be able to use non web-based IM application, it has to be downloaded, installed and most time configured. Secondly, most non web-based instant messaging applications are platform dependent, which means different application for different operating system. Thirdly, in most cases especially in schools, office instant messaging is blocked by firewall so users will not be able to use the application.

Application download, installation and configuration

These are real hindrance to instant messaging communicating. Some of the instant messaging applications files are big (over 5MB), and bandwidth sometimes is expensive and download speed low. In these cases, downloading the application file and updates will not be cheap and also takes time which a times can be frustrating. Secondly, Most of the time users use public computer

(schools, cafés, etc) and installation is not allowed in most computers in schools, cafés, etc. In a situation, the IM application the user is using is not installed on the computer the user will not be able to use instant messaging.

Platform dependent

Non web-based instant messaging applications are platform dependent. That is, a single application file for an IM application will not be used in different operating system. The developer has to develop different application file for different operating system. This process is expensive and time wasting. And also users using different operating systems have to download more than one application file for one instant messaging program.

Firewall issue

Most schools, companies and organizations block instant messaging in the firewall. So users will not be able to communicate using instant messaging system in those places.

The systems and software requirements for the system

Software requirement

The following programming language will be used to develop and implement the system:

Flex: - an open source version of flash from Adobe Macromedia. Flex is used in developing and deploying cross-platform rich internet applications [8].

ActionScript: - a scripting language based on ECMAScript. It is used mainly for development of websites and software using Adobe Flash Player platform [9].

Server requirement

The system will be based on Extensible Messaging and Presence Protocol XMPP (both server and client) platform. XMPP is an open source Extensible Markup Language XML-based protocol for server-to-server near-real-time extensible instant messaging and presences [10]. Red5 will be used for audio and video streaming, and MySQL to implement database.

XMPP Server (Openfire): - An XMPP Server provides basic presence, messaging and routing features [11]. Openfire is an open source Real Time Collaboration (RTC) servers, that uses XMPP for instant messaging [12].

XMPP Clint (SparkWeb / Jabbear): - An XMPP Client is a software application that enables connection to an XMPP Server for instant messaging with other people [11]. SparkWeb is an open source, web-based instant messaging client using XMPP that is optimized for business and organization. It has built-in support for group chat and strong security [12]. Jabbear is an instant messaging client using XMPP.

Red5: - an open source flash Real Time Messaging Protocol (RTMP) server, written in java that supports streaming Audio/Video (FLV and MP3). Is an open source alternative to Adobe Flash Media Server [14].

MySQL: - an open source relational database management system (RDBMS) based on Structured Query Language (SQL) – one of the most popular languages for adding, accessing, and processing data in a database; and runs as a server providing multi-user access to a number of databases.

Existing solution

There are already existing open source implementations of this system. Some of them include:

Face book: - a social network site originally for college students, now catering for all age group. It is privately owned and operated by Facebook; Inc. Facebook developed a web-based instant messaging programming for its member to communicate instantly with each other.

Gmail: a free search-based webmail service that combines the best features of traditional e-mail with Google search technology. Gmail incorporates web-based instant messaging program in its e-mail service.

Meebo: - an Ajax-based web-based instant messaging program that supports multiple IM services, including MSN, Yahoo messenger, etc.

IMLove: - a web-based instant messenger service that support multiple IM programs such as Yahoo messenger, MSN messenger, AIM, GTalk, etc

The components of IM implementation

When the user logs in to the IM interface on the browser, the user login details and the computers IP address are sent to the XMPP Server through the XMPP Client. XMPP Server authenticates the user, and sends the contact list of the user, and the IP address of the computer of the contacts that are online. The user can then communicate with the user directly see Figure 3.

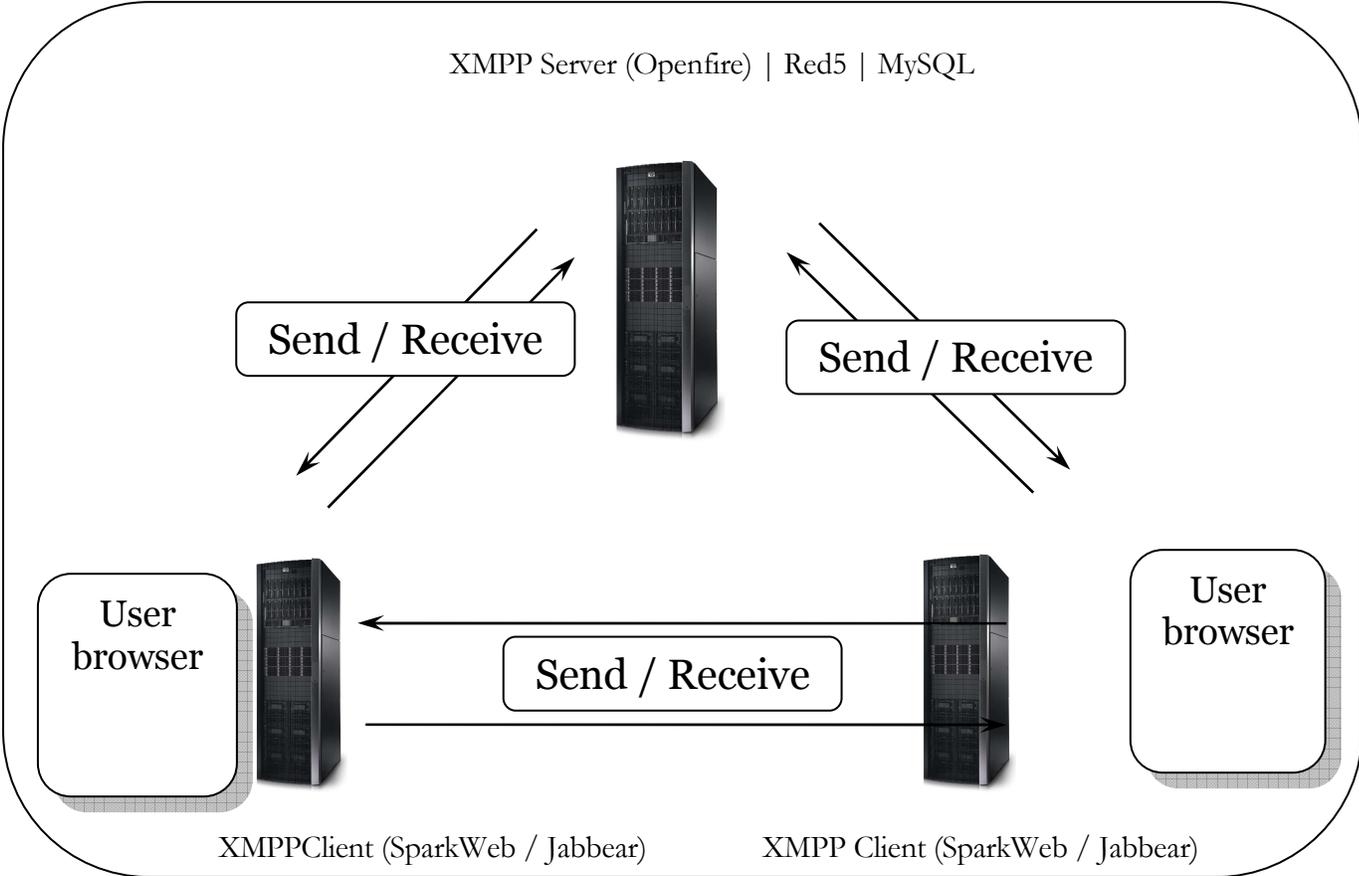


Figure 3: The process of instant messaging communication.

Chapter 4

USER INTERFACE SPECIFICATION

Introduction

The previous chapter discussed and analyzed the user's requirement from the designers point of view based on the system and software requirements to implement the requirement. In this chapter the user Interface Specification (UIS) will be discussed. That is a complete description of the user interface, what it looks like, what exactly it is going to do, and how the user interacts with the system.

Complete description of the user interface

The system comprises of "Login Interface" for logging in to the system, "New Account Interface" for creating new account in the system, "Main Window" for presence awareness and contact list display, and "Chat Window" for chatting with other users.

The user logs in to the system using the "Login Interface" by typing the username in the username TextInput, the server to log in to in the server ComboBox, and the password in the password TextInput, then click the "Login Button" to login. Before user can use the system he/she must login first.

To login to the system you are required to have created an account in the system. The "New Account Interface" is used to create an account in the system by typing the username in the username TextInput, the server in the server ComboBox, the password in the password TextInput, and the password

confirmation in the confirm password TextInput, then click the “Create Button” to create an account or the “Cancel Button” to cancel the process.

When a user logged in to the system the “Main Window” will show. The “Main Window displays the user’s image if any, and the username. It also displays a ComboBox for the user’s presence awareness, where user set his/her presence information (available, away, do-not-disturb, etc), a search TextInput for searching contacts, and the user’s contact list (i.e. the list of all the contacts the user added). Also, it shows the contacts tab for contact list display, conferences tab for conference chat display, a button for searching users in the server, a button for adding contact, and exit button for logging out and closing the “Main” and “Chat” windows.

The “Chat Window” is used for chatting. User type the message in the message TextArea and the message is displayed in the Text filed when the message is sent. The TabBar displays different chat windows for different contacts the user is chatting with.

What the interface looks like to the user

Login Interface

This is the first interface the user will see; it is where the user logs in to the system, see Figure 4. The login interface comprises of “User Name” TextInput for the user’s user name, “Server” ComboBox for the server user want to login to, “Password” TextInput for the users password, “Login Button” to login after typing the user name, server and password, and the “Create Account” button for creating a new user account.



Figure 4: Login interface for logging in to the system

New Account interface

The new account interface is the interface the user uses to create a new account in the system, see Figure 5. It comprises of “User Name” TextInput for the user’s user name, “Server” ComboBox for the server user want to create an account in, “Password” TextInput for the users password, “Confirm Password” TextInput for the users password confirmation, “Cancel Button” to cancel the creation of new account and “Create Button” to create a new account after typing the user name, server, password and password confirmation.



Figure 5: New account interface for creating new account in the system.

Main Window

The main window is the primary window of the system, see Figure 6. It displays the user's image, account name, a ComboBox for the user's presence awareness, a search TextInput for searching contacts, and the user's contact list. It also displays contacts tab for contact list display, conferences tab for conference chat display, a button for searching users in the server, a button for adding contact, and exit button.

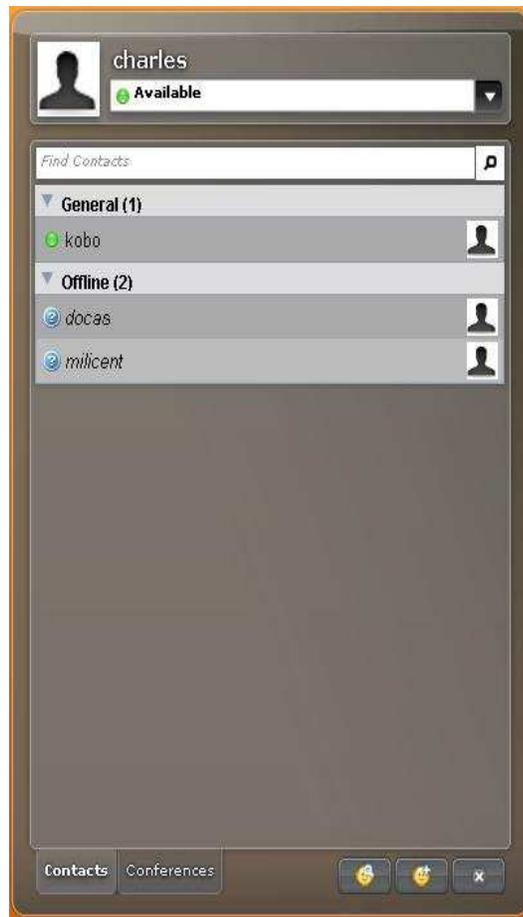


Figure 6: Main window for presence information and contact list of the user.

Chat Window

The chat window is the window for chatting see Figure 7. It comprise of TextArea for typing the message, Text- a non-editable multiline text field the displays the sent messages, and a TabBar that contains chat windows for different contacts the user is chatting with.

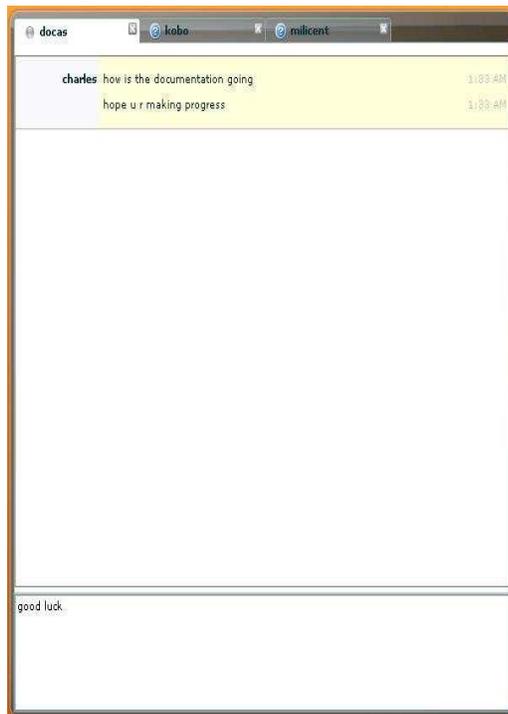


Figure 7: Chat window for chatting with other users.

How the user interface behaves

When user enters username and password in the login interface and click enter, the main window opens if the username and password exist in the system, if not, the interface displays an error message is displayed. The new account interface opens the login interface with a message that the new account was created

successfully if the information entered by user is accepted, otherwise, it will display an error message stating that the account was not created.

On the main window, clicking on the presence awareness ComboBox displays the awareness options (available, away, on phone, do-no-disturb, etc), see Figure 7. Clicking on the add contact button displays the add contact table as shown in Figure 8. When a contact on the contact list is right clicked on, it displayed options like start a chat, rename contact, remove contact, view profile, groups, etc, as shown in Figure 9.

The chat window displays message typed in the message TextArea in the Text field. When a contact TabBar is clicked on it displays the chat window for that contact.

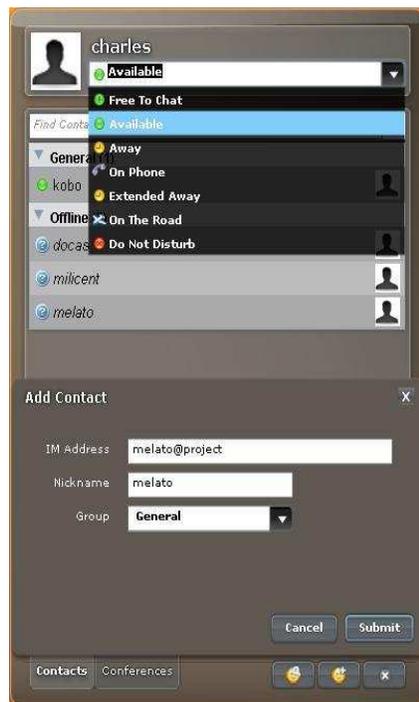


Figure 8: Main window showing options for presence awareness and add contact form.



Figure 9: Main window showing options available when a contact is right clicked on.

How the user interacts with the system

The overview of user interaction with the system is illustrated with a use case diagram as shown in figure 11 below.

The user uses the “New Account” window, Figure 5 to create an account in the system by typing the username in the username TextInput, the server in the server ComboBox, the password in the password TextInput, and the password confirmation in the confirm password TextInput, then click the “Create Button” to create an account.

The “Login” window, Figure 4 is used by the user to login to the system by typing the username in the username TextInput, the server to log in to in the server ComboBox, and the password in the password TextInput, then click the “Login Button” to login.

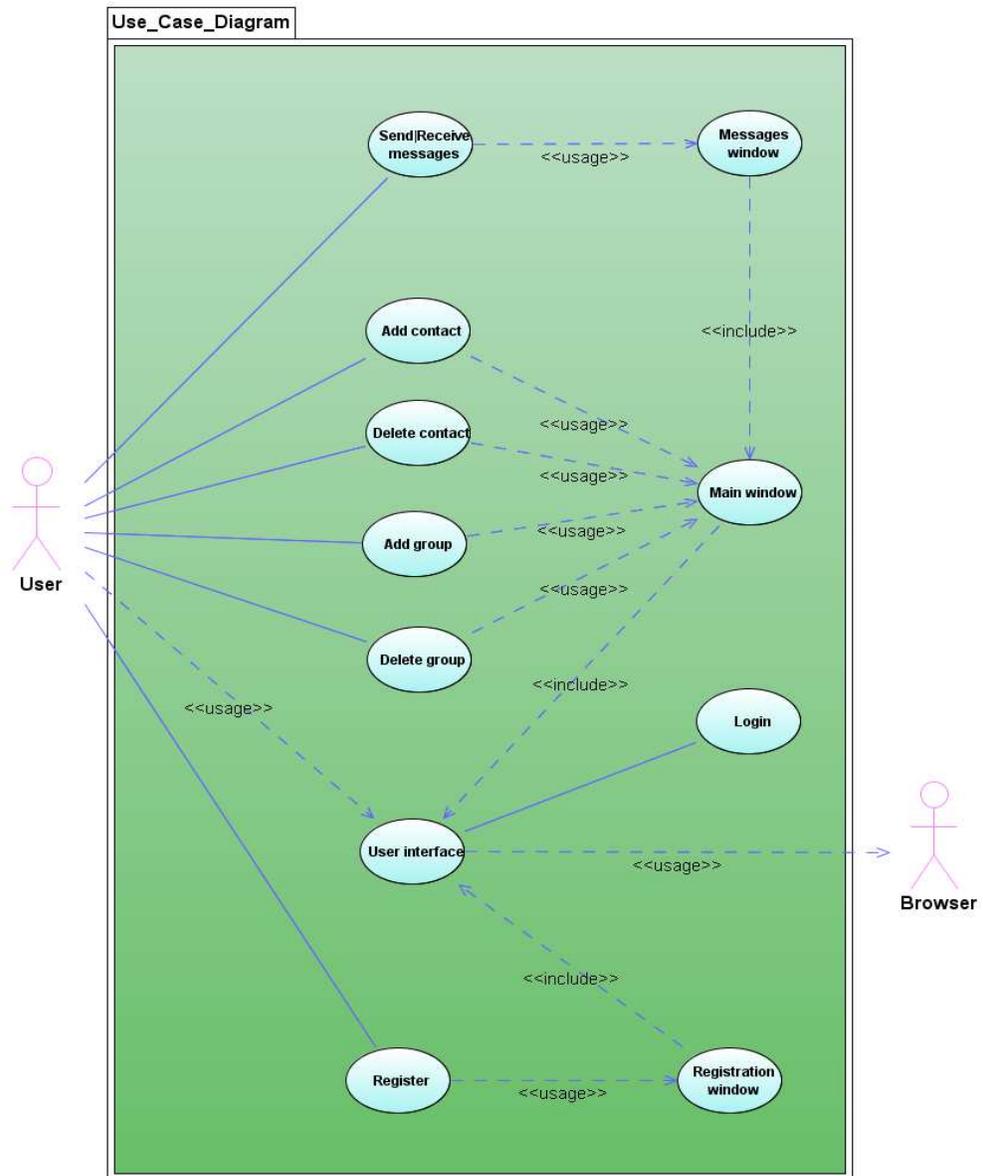


Figure 10: Use Case diagram showing how user interact with the system

The user uses the “Main Window” Figure 6, to set the presence awareness status by selecting the options from the ComboBox, search for contact by typing the contact’s username in the search TextInput and clicking search or by clicking on

the search contact button, typing the contact username and/or address and clicking search button. Also add contact to the contact list by clicking the add contact search button which pops up the add contact form, the user fills the form and then click submit. By right clicking on a contact the user can initiate search with the contact by selecting the start a chat option, rename and remove the contact by selecting the rename or remove option, view the contact profile by selecting the view profile option. The user exits the “Main” and “Chat” window by clicking on the exit button.

Summary

This chapter described the complete user interface, what the interface does, and how the user interacts with the system. The complete screen shot of the interfaces were presented and described. Also, all the interactive component of the interfaces such as buttons, ComboBox, etc and the actions they triggers were discussed.

Chapter 5

OBJECT ORIENTED ANALYSIS (OOA) / HIGH LEVEL DESIGN (HLD)

Introduction

This chapter will describe the object oriented analysis (OOA) known as the high level design, which is an object oriented view of the system. A detailed description of each object is presented and documented in the data dictionary. Class diagram showing the name, attributes, and methods of each class will be discussed, and the relationships between the classes established.

Data dictionary

The data dictionary as shown in Table 1, illustrates the detailed description and meaning of all the objects of the system.

Objects Attributes	Description
Interface	Interface is the parent class for both the login and new account interfaces. It has three attributes; username, server, and password.
- userName	Username created by the user in the system
- Server	Name of the server the user opened an account in

- Password	Login password chosen by the user
Login interface	Login interface is the interface user users to login to the system; it is the first interface user see when accessing the system. It contains the button for new account interface. Login interface inherits three attributes; username, server, and password from the interface class.
New account interface	New account interface is used by user to create an account in the system. It has four attributes; three of them (username, server, and password) it inherited from the interface class, the last confirm password is unique to it.
- Confirm password	Confirmation of the Login password
Main window	Main window is the primary window of the system. It is the window that displays the profile, status, and group and contact list of the user. The main window has three attributes; contactName, groupName, and status.
- contactName	Username of the contact the user want to add
- groupName	Group name the user added the contact to
- Status	Presence awareness of the user
Message window	Message window is used to send/receive message. It has three attributes; senderName, receiverName,

	and message.
- senderName	Username of the user that sends the message
- receiverName	Username of the user the message was sent to
- message	The message that was sent or received

Table 1: The data dictionary of the system

Class diagram

Class diagram is used to describe the structure of the system by showing the system's objects/classes, their attributes, methods within the classes, and the relations between the classes. The system has one abstract class interface and four main class login interface, new account interface, main window, and message window see Figure 11.

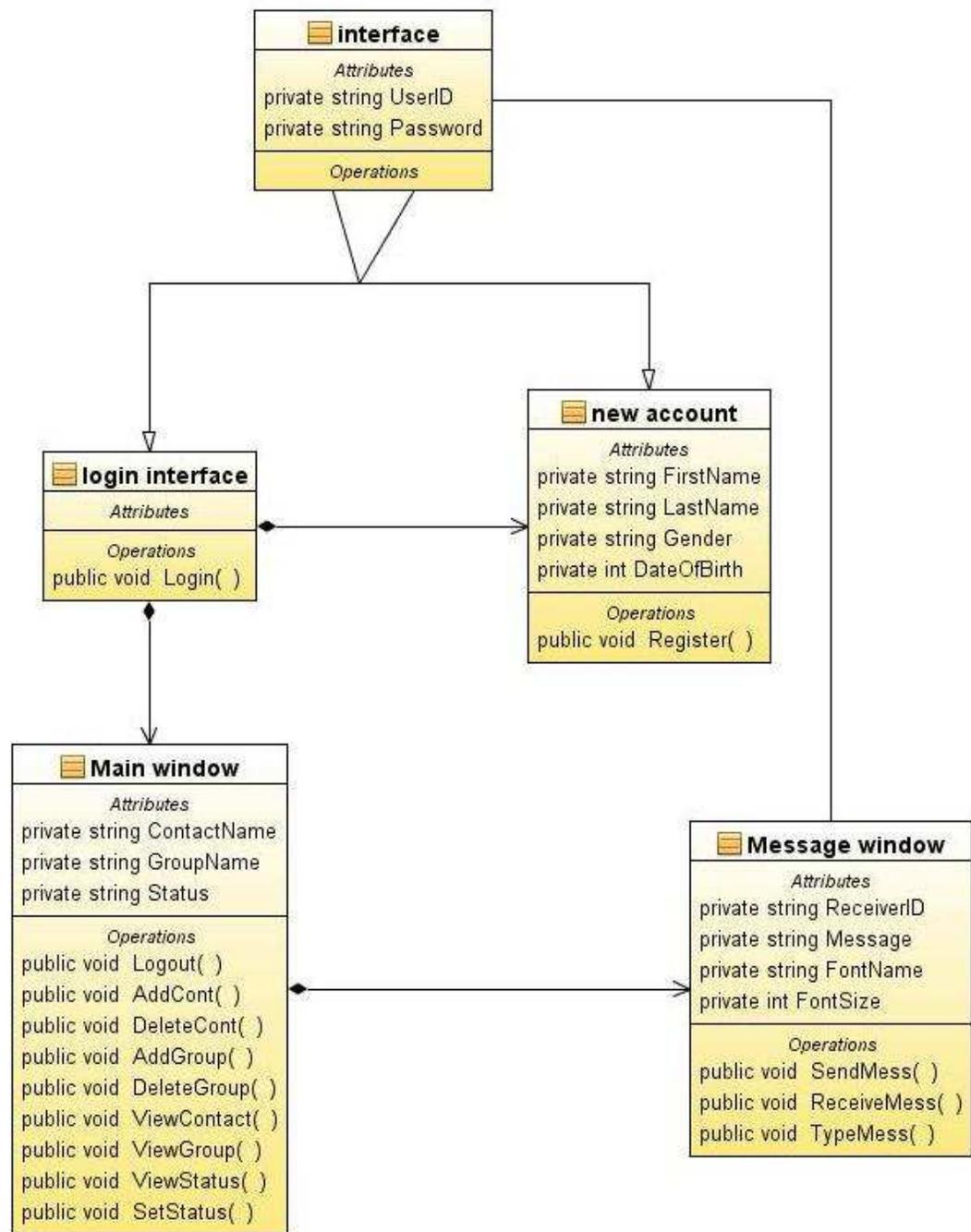


Figure 11: Class diagram showing the name, attributes and methods of each class

Relationships between the classes

Figure 12 illustrates the relationships between the classes of the system.

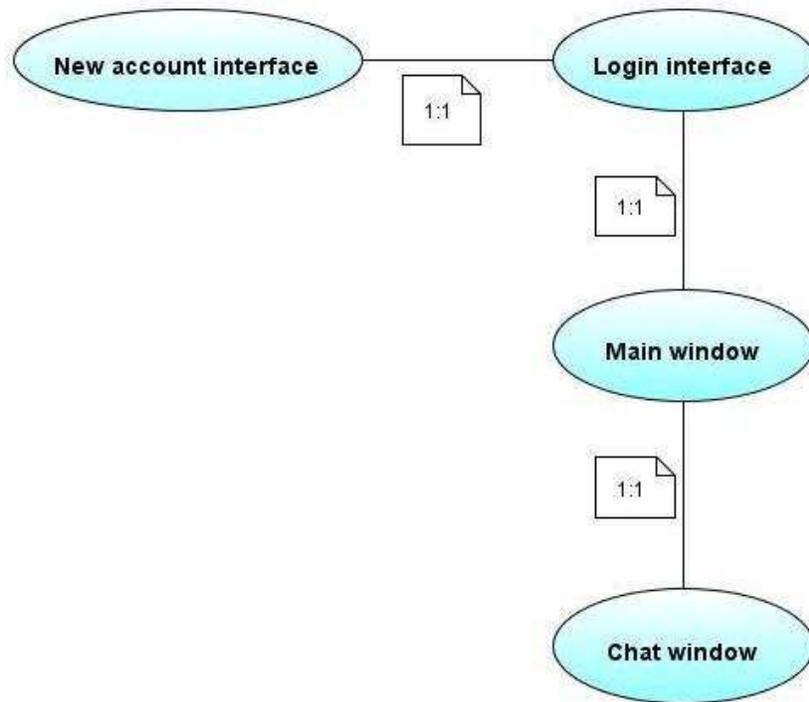


Figure 12: Relationship between the classes

Set of class diagrams and data dictionary for the application domain

For the application domain, we have the main and chat-window class with their attributes, see figure 13 and described in table 4.

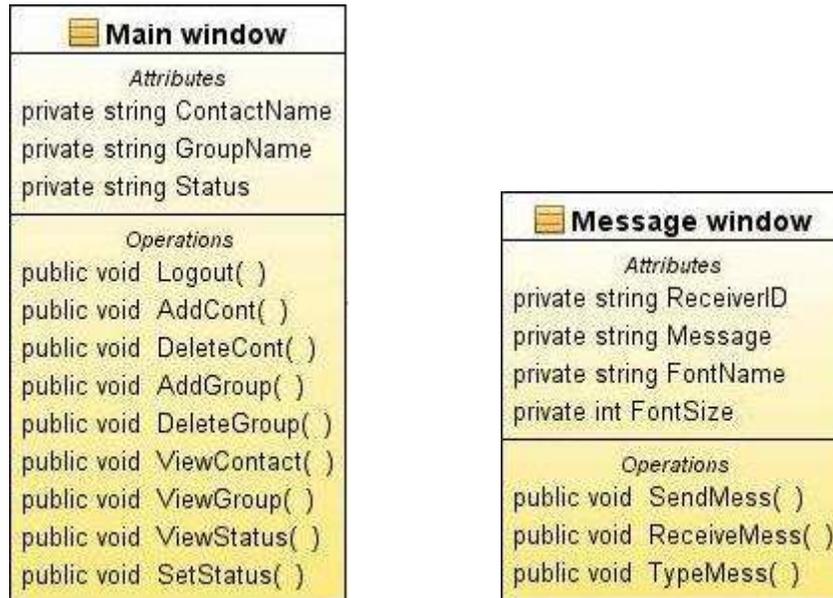


Figure 13: Class diagram for the application domain

Objects Attributes	Description
Main window	Main window is the primary window of the system. It is the window that displays the profile, status, and group and contact list of the user. The main window has three attributes; <code>contactName</code> , <code>groupName</code> , and <code>status</code> .
- contactName	Username of the contact the user want to add
- groupName	Group name the user added the contact to

- Status	Presence awareness of the user
Message window	Message window is used to send/receive message. It has three attributes; senderName, receiverName, and message.
- senderName	Username of the user that sends the message
- receiverName	Username of the user the message was sent to
- message	The message that was sent or received

Table 2: Description of class for the application domain

Set of class diagrams and data dictionary for the interface domain.

The interface domain comprises of the Login interface and the new account interface with their attributes see figure 14, and described in table 4.

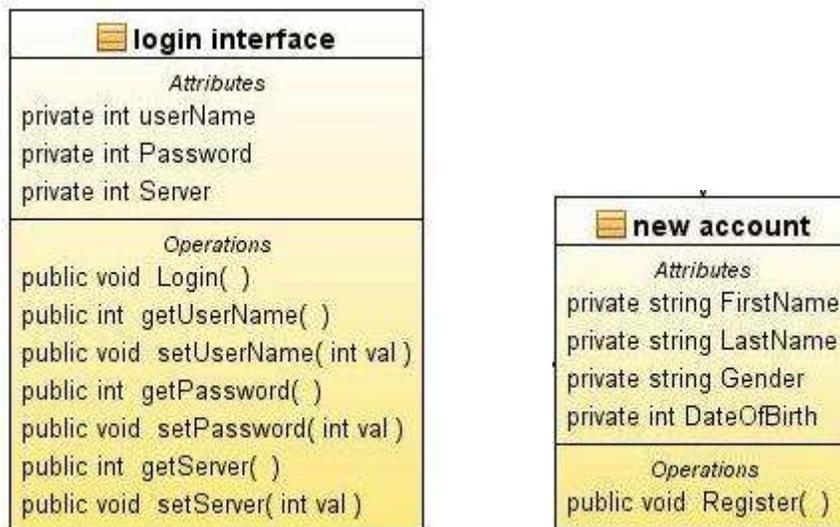


Figure 14: Class diagrams for the interface domain

Objects Attributes	Description
Login interface	Login interface is the interface user users to login to the system; it is the first interface user see when accessing the system. It contains the button for new account interface. Login interface has three attributes; username, server, and password.
- userName	Username created by the user in the system
- Server	Name of the server the user opened an account in
- Password	Login password chosen by the user
New account interface	New account interface is used by user to create an account in the system. It has four attributes; username, server, password, and confirm password.
- userName	Username created by the user in the system
- Server	Name of the server the user opened an account in
- Password	Login password chosen by the user
- Confirm password	Confirmation of the Login password

Table 3: Description of table for the interface domain

Summary

This chapter discussed an object oriented view of the system, and description of each object and their attributes. Also class diagram showing the name attributes, and methods of each class, and the relationships between the classes were presented and described.

Chapter 6

OBJECT ORIENTED DESIGN / LOW LEVEL DESIGN

Introduction

The previous chapter discussed the object oriented view of the system, and the class diagram with the name, attributes, and methods of each class, and the relationships between the classes. This chapter the Object Oriented Design takes the classes in Object Oriented Analysis a level deeper by providing algorithm description and pseudo-code for class | object methods.

Details of class attributes and methods

The Table 2 below illustrates the detailed description and meaning of all the class attributes and methods.

Objects Attributes	Description
Login interface	
userName	Username created by the user in the system
Server	Name of the server the user opened an account in
Password	Login password chosen by the user
Login()	Method that logs in user to the system. It gets the username, server, and password, authenticate the login

	information, and logs the user in
New account interface	
userName	Username created by the user in the system
Server	Name of the server the user opened an account in
Password	Login password chosen by the user
Confirm password	Confirmation of the login password
createAccount()	This method gets the user details (username, server, password, and confirm password), verify the details and create and account in the system for the user.
Main window	
contactName	Username of the contact the user want to add
groupName	Group name the user added the contact to
Status	Presence awareness of the user
Logout()	The logout method closes the user connection and exit from the main and chat windows.
addCont()	This method gets the username of the contact the user want to add and add it to the user contact list.
deleteCont()	This method gets the username of the contact the user

	wants to delete and remove it from the user contact list.
addGroup()	This method gets the username of the contact and the group name the user want to add the contact to and add the contact name to that group.
deleteGroup()	This method deletes the group name and all the contacts in it from the main window.
setStatus()	This method displays the user presence awareness status.
Message window	
senderName	Username of the user that sends the message
receiverName	Username of the user the message was sent to
message	The message that was sent or received
sentMess()	This method delivers the message from the sender to the receiver.
receiveMess()	This method receives the message from the sender.

Table 4: The detailed description of the attributes and methods

State diagram

Figure 15 shows the state diagrams for the login object/class; it describes the process the user goes through to login. While figure 16 shows the new account object/class that describes the process of creating account in the system.

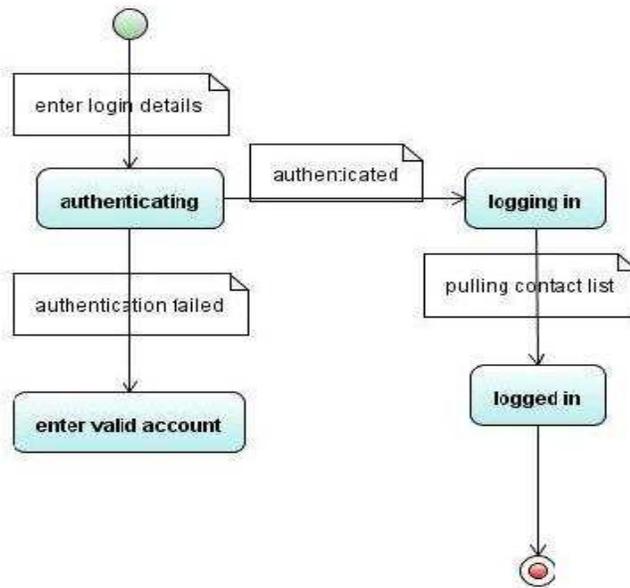


Figure 15: Login state diagram describing the process of logging in to the system

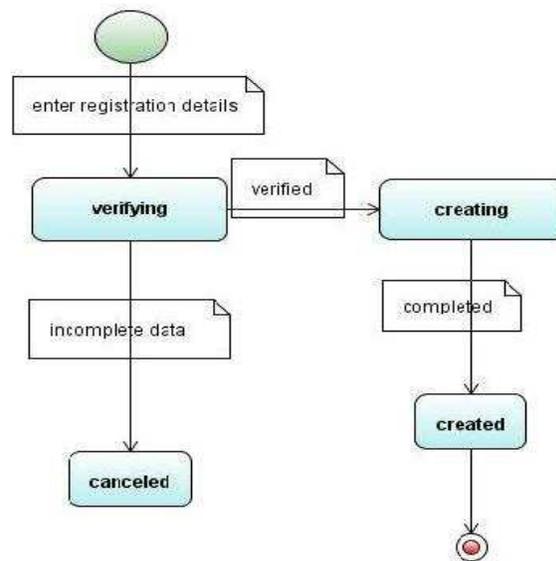


Figure 16: New account state diagram describing the process of creating account

Event diagram

Event diagram outlines the process/steps of using the system. See figure 17.

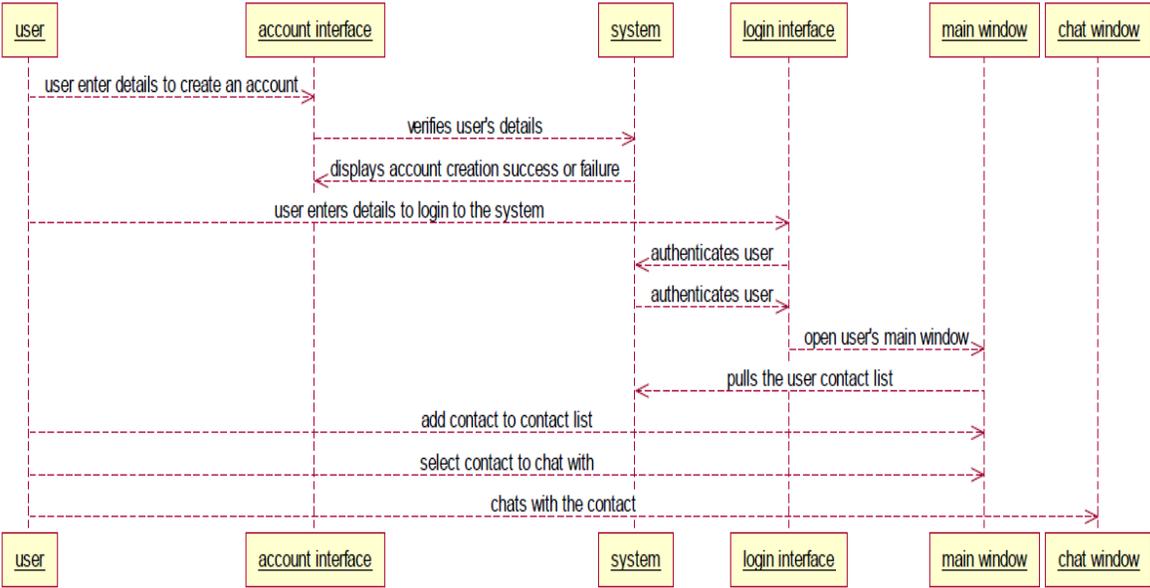


Figure 17: Sequence diagram showing the steps for user interaction with the system

Activity diagram

Fig 18 shows the activity diagram for the system; it describes the process of user interaction with the system.

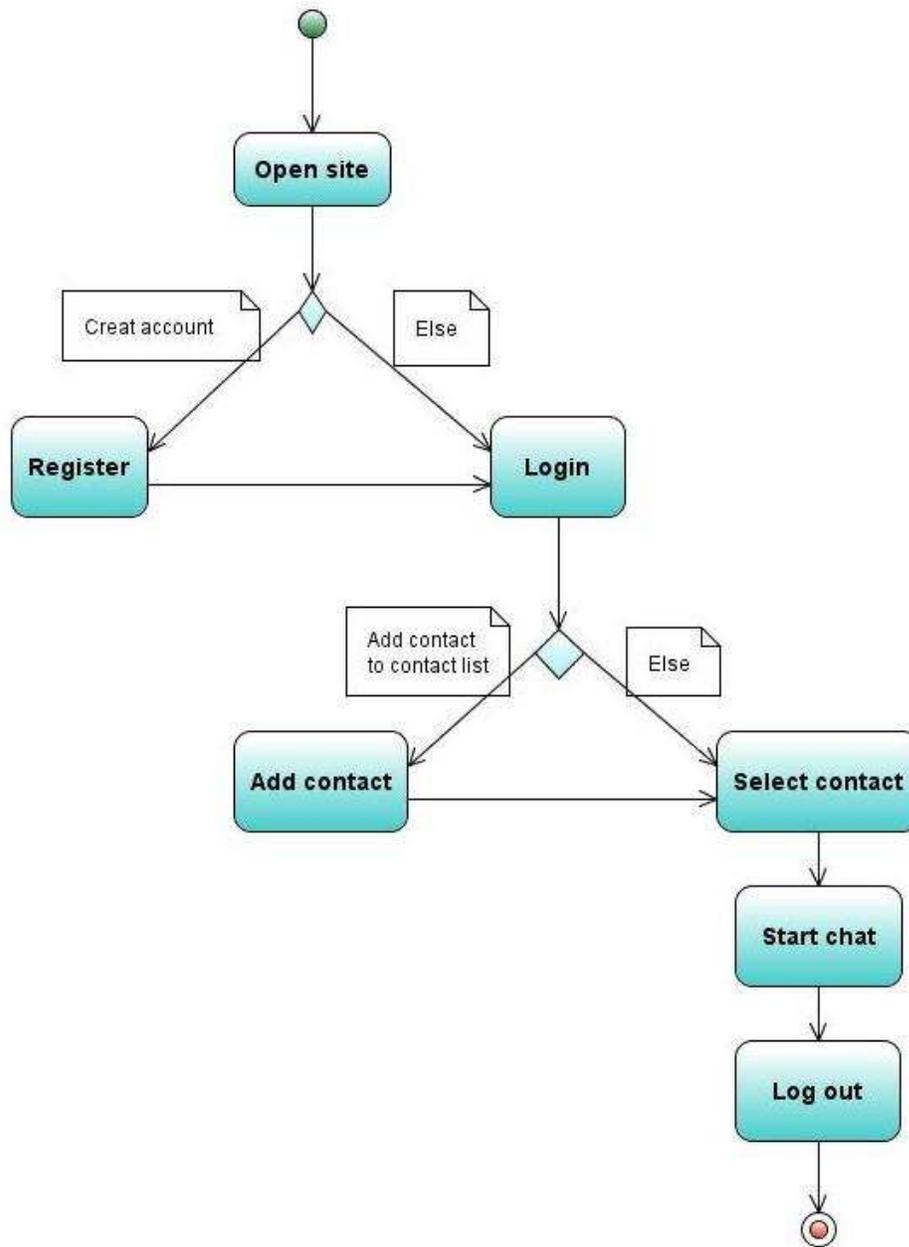


Figure 18: Activity diagram show the process of user interaction with the system

Algorithm

```
Class login Interface extends Interface {  
  UserName: enter user name;  
  Server: enter server name;  
  Password: enter password;  
}
```

```
Class new account {  
  UserName: enter user name;  
  Server: enter server name;  
  Password: enter password;  
  Confirm password: re-enter password;
```

```
Register () {  
  Enter username, server, password, and confirm password.  
  Click create to create the account  
}  
}
```

```
Class Main window {  
  ContactName: add username of contact;  
  GroupName: add group name;  
  Status: select presence awareness status;
```

```
Logout () {  
  Click exit to logout  
}
```

```
AddCont () {  
  Select and add contact to the contact list  
}
```

```
DeleteCont () {  
  Select and delete contact from the contact list  
}
```

```
AddGroup () {  
  Select and add group to main window  
}
```

```
Delete Group () {  
  Select and delete group from main window  
}
```

```
}  
  
Class Message window {  
SendMess () {  
Type and send message.  
}  
}
```

Summary

This chapter discussed the low level design; the inner details of the class attributes and methods, state diagram, activity diagram, event diagram. The chapter also features the algorithm for the system.

APPENDICES

BIBLIOGRAPHY

- [1] “Instant messaging”, http://en.wikipedia.org/wiki/Instant_messaging, November, 2007.
- [2] Jeff Tyson and Alison Cooper, “How Instant Messaging Works”, <http://communication.howstuffworks.com/instant-messaging.htm>, March 27, 2009.
- [3] Google, “Welcome to Google Talk”, <http://www.google.com/talk/about.html>, March 27, 2009
- [4] Robin Good, “Instant Messaging Tools and Technology: A Mini-Guide”, http://www.kolabora.com/news/2006/09/28/instant_messaging_tools_and_technology.htm, September 28, 2006.
- [5] Michael Arrington, “Dead Simple Flash Based WebChat For MySpace”, <http://www.techcrunch.com/2008/06/16/dead-simple-flash-based-webchat-for-myspace/>, June 16, 2008.
- [6] Robin Good, “*Instant Messaging 2.0: Secure Instant Messaging Set To Be The Center Of Your Communication And Collaboration Universes*”, http://www.masternewmedia.org/news/2006/06/23/instant_messaging_20_secure_instant.htm, June 23, 2006.
- [7] *The translatorscafe*, “Instant messaging — a new language”, <http://thetranslatorscafe.wordpress.com/2008/05/08/instant-messaging-a-new-language/>, May 8, 2008.

[8] Wikipedia, “Adobe Flex”, http://en.wikipedia.org/wiki/Adobe_Flex,
November 2008.

[9] Wikipedia, “ActionScript”,
<http://en.wikipedia.org/wiki/ActionScript>, (assessed 30th March, 2009).

[10] webopedia, “XMPP”,
<http://www.webopedia.com/TERM/X/XMPP.html>, September 20, 2004.

[11] xsf, “XMPP Software”, <http://certs.xmpp.org/software/clients.shtml>,
(assessed: March 30, 2009).

[12] Igniteraltime, Openfire / SparkWeb”,
<http://www.igniterealtime.org/projects/openfire/index.jsp>, (assessed:
March 30, 2009).

[13] Jabbear Messenger, “Jabear”,
<http://www.jabbear.com/en/index.php>, (assessed: March 30, 2009).

[14] Red5, “Open Source Flash”, <http://osflash.org/red5>, (assessed: May
4, 2009).

INDEX

A

After printing · 1

J

Junk · iv