

Instant Messenger Using JAVA RMI

By

Hina Fatima



Supervised
By
Mr. Ali Iqbal

A report is submitted to the department of Computer Science. Bahria University,
Islamabad

In partial fulfillment of the requirement for the degree of MCS

Department of Computer Sciences Bahria University, Islamabad

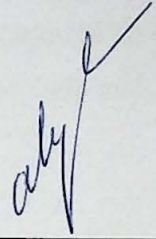
**Dedicated to my Parents and my bestest friends
Hureen Fatima & Gulmina Rextina**

Certificate

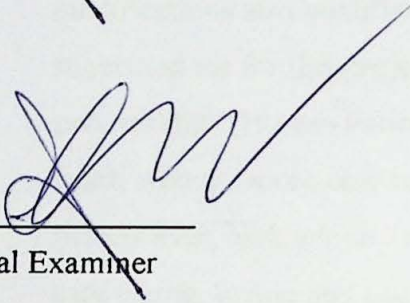
We accept the work contained in this report as confirming to the required standard for the partial fulfillment of the degree of MCS in the subject of _____



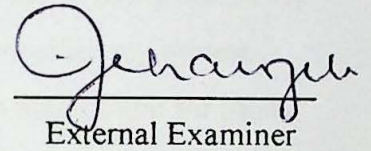
Head of Department



Supervisor



Internal Examiner



External Examiner

Acknowledgments

Above all I am grateful to Allah Almighty for enabling me to complete this project.

I am extremely grateful to my supervisor Mr. Ali Iqbal for guiding me throughout this project. I am also very grateful to Mr. Fazal-e-Wahab, Ms. Farzana for their help and support.

I will like to thank my beloved family, specially my father who besides his post Graduate qualifications also qualified computer science from UET, Lahore. My father well supported me for this project. After my dad & mom, I would like to thank two personalities (Hureen Fatima, Gulmina Rextina) who hold a very important place in my heart, who are more dear to me even than my own life, whom I consider my bestest friends ever, with whom I really feel comfortable. I would like to thank them for their care taking, loving and encouraging attitude towards me. I am thankful to them for their moral support and prayers. Their moral support and specially prayers gave me the feeling that I hold a very special place in their hearts. I don't have proper words even to thank them in a special way when I think of their love, support, prayers, and encouraging attitude towards me. While writing my acknowledgement I tried my best to express my feelings, gratitude for all those I mentioned above, but I think I am out of words.

Abstract

This project is designed in JAVA RMI, MS ACCESS, JMF. This project is designed to enable users to communicate easily and effectively not only through textual messages but also through live audio and video conferencing. Instant messenger allows one to maintain a list of people that you wish to interact with. You can send messages to any of the people in your list, often called a **buddy list** or **contact list**, as long as that person is online. Sending a message opens up a small window where you and your friend can type in messages that both can see. It also allows video and audio conferencing.

Table of Contents

1. Introduction	1
1.1. Purpose of Project.....	1
1.2 Project Overview	1
1.2.1 The Directory Server	2
1.2.2 The Client Application.....	2
1.2.3 Voice & Video Conferencing.....	2
2. Literature Survey	3
2.1. Messenger History	3
2.2. Relational Database	5
2.3 Distributed Computing.....	6
2.3.1 Difference b/w Socket Programming & Distributed Computing	7
2.4. Some Important Features of JAVA	8
2.4.1 Java is object-oriented	8
2.4.2 Java is platform Independent	8
2.4.3 Java is Distributed	9
2.4.4 Java is robust	9
2.4.5 Java is multithreaded	9
2.4.6 Java is Dynamic	9
2.4.7 Java is simple	9
2.5 Java RMI (Remote Method Invocation)	10
2.5.1 Remote Interfaces, Objects & Methods.....	11
2.5.2 RMI Architecture	12
3. Proposed System	14
3.1 Existing System with tools & techniques	14
3.2 Proposed System with tools & techniques.....	14
3.3 Advantages of Proposed System	15
4. Design.....	16
4.1. Design	16
4.1.1 Object Oriented Design	16
4.1.1.1 UML Diagram	16
4.1.1.2 User Interface Diagram	16
4.1.2 Prototypes	17
4.2 Class Description	23
4.2.1 Description.....	23
4.2.1.1 Sign-In	23
4.2.1.2 Messenger Screen	23
4.2.1.3 Client.....	23
4.2.1.4 Server	23
4.2.1.5 UserSerImpl	23
4.3.2 Description	26
4.3.2.1 Messenger	26
4.3.2.2 MsgWin.....	26
4.3.2.3 AVTransmit	26

4.3.2.4 AVRecieve	26
4.3.2.5 Video Transmit.....	26
4.3.3 Description.....	28
4.3.3.1 Sign-Up.....	28
4.3.3.2 ClientReg	28
4.3.3.3 UserRegImpl	28
4.3.3.4 UserRegSerImpl	28
5. Implementation	29
5.1. Hardware Requirements.....	29
5.2. Software Requirements	29
5.3. Implementation Tools	29
5.3.1 Java Media Framework	29
5.3.2 Jdk 1.3	29
5.3.3 Rational Rose 2000	30
5.3.4 Adobe Photoshop	30
5.4. Features of System.....	30
5.4.1 Registration.....	30
5.4.2 Sign-In	30
5.4.3 Change Password	31
5.4.4 Add Friends.....	31
5.4.5 Delete Friends	31
5.4.6 Audio Conversation.....	31
5.4.7 Video Conferencing	31
5.4.8 Additional Features	32
5.4.8.1 Font Color.....	32
5.4.8.2 Smiley	32
5.4.9 Help Menu	32
5.5 Screen	33
6. Testing and Evaluation	42
6.1. Testing	42
6.1.1 Verification	42
6.1.2 Validation	42
6.2 Unit Testing.....	42
6.3. Integration Testing.....	42
6.4 System Testing	43
6.5 Design Test Cases.....	43
6.5.1 Equivalence Class Analysis	43
6.5.2 Boundary Analysis	43
6.5.3 Cause Effect Graphing	43
6.5.4 Error Guessing	43
7. Conclusion	50
8. Future Development	51
Bibliography.....	52
Appendix A	53
Appendix B	58
Appendix C	64

Figures Table of Content

Fig 1: Yahoo Messenger	4
Fig 2: ER Diagram	5
Fig 3: RMI Distributed Application.....	11
Fig 4: RMI System.....	12
Fig 5: RMI System.....	13
Fig 6: RMI Architecture	13
Fig 7: Prototype (User Registration Screen)	17
Fig 8: Prototype (Message Box)	18
Fig 9: Prototype (Thankyou Screen).....	18
Fig 10: Prototype (Sign-In Screen).....	18
Fig 11: Prototype (Main Messenger Screen).....	19
Fig 12: Prototype (Messaging window).....	19
Fig 13: Prototype (Audio / Video Interface)	20
Fig 14: Use Case Diagram.....	21
Fig 15: Class Diagram for User Sign-In.....	22
Fig 16: Class Diagram for Client & Client Interaction.....	25
Fig 17: Class Diagram for New user registration	27
Fig 18: Splash Screen.....	33
Fig 19: Sign-In & Sign-Up Screen.....	33
Fig 20: Registration Window	34
Fig 21: Password mismatch error dialog box	34
Fig 22: Error message	35
Fig 23: Online Friends.....	36
Fig 24: Chat Window	37
Fig 25: Color Chooser	38
Fig 26: Smiley Window	39
Fig 27 Password change dialog box.....	40
Fig 28: Offline Dialog box	40
Fig 29: Offline Friends.....	41